February 2009

LM3: A High-Resolution Lake Michigan Mass Balance Water Quality Model

by

Wilson Melendez
Michael Settles
Computer Sciences Corporation

James J. Pauer
Z-Tech Corporation, an ICF Company

Kenneth R. Rygwelski
U.S. Environmental Protection Agency
Office of Research and Development
National Health and Environmental Effects Research Laboratory
Mid-Continent Ecology Division-Duluth
Large Lakes and Rivers Forecasting Research Branch
Large Lakes Research Station
9311 Groh Road
Grosse Ile, Michigan 48138

Recycled/Recyclable
Printed with vegetable-based ink on paper that contains a minimum of 50% post-consumer fiber content processed chlorine free.

Notice

The information in this document has been obtained primarily through funding by the United States Environmental Protection Agency (USEPA) under the auspices of the Office of Research and Development (ORD) and by the Great Lakes National Program Office (GLNPO). The report has been subjected to the Agency's peer and administrative review and it has been approved for publication as an USEPA document. Mention of trade names or commercial products does not constitute endorsement or recommendation for use.

Foreword

The Lake Michigan Mass Balance Project (LMMBP) was developed by the United States Environmental Protection Agency (USEPA) to study the transport, fate, and bioaccumulation of toxic chemicals, such as polychlorinated biphenyls (PCBs), in Lake Michigan. The project consisted to two main phases: (1) data collection and sample analysis, and (2) water quality model development and application. The data collection and sample analysis were carried out by various contractors under the guidance of the Great Lakes National Program Office (GLNPO). The data collection took place during the time period of 1994 and 1995. The data analysis was conducted over a period of several years including the 1994-1995 study period and beyond. The data were then used within the framework of various water quality models for calibration and validation purposes. There were several water quality models developed that simulated the movement of PCBs within the lake and their bioaccumulation in fish. The models had different spatial resolutions for the water column and sediment column.

The present report is a user's manual that describes one of the water quality models used for the LMMBP. LM3 is a mass balance water quality model that simulates eutrophication, atrazine, chloride, and PCBs. It is a high-resolution model with a horizontal scale of 5 km x 5 km on each cell of the grid. The LM3 development started in the mid-1990s. The eutrophication model in LM3 was used primarily to simulate eutrophication in the lake and to provide biotic carbon loads to the toxic chemical model known as LM2-Toxic. LM3 was also applied to study the transport and volatilization of atrazine. It has been configured to simulate PCB congeners.

This report contains the mathematical equations used to simulate the kinetics of the eutrophication, atrazine, chloride, and PCB congeners. It contains a detailed description of the transport algorithm used in the model. There is a description of the sediment transport model used for PCBs. This sediment transport model is considered a stopgap solution before a more complete sediment transport algorithm can be implemented. All the calibration parameters used in LM3 are described. The program code organization and management are also described. Examples of input files used for various simulations are provided.

Abstract

This report is a user's manual that describes the computer application known as LM3. LM3 is a high-resolution mass balance water quality model for Lake Michigan. It was developed to simulate eutrophication, atrazine, chloride, and polychlorinated biphenyl (PCB) congeners in Lake Michigan in support of the Lake Michigan Mass Balance Project (LMMBP). The LMMBP was a study conducted by the United States Environmental Protection Agency (USEPA), Office of Research and Development in cooperation with the USEPA Great Lakes National Program Office.

This manual includes a discussion of the theory behind the parameters being modeled, a description of the organization of the source code, and the user interface associated with the LM3 application. The manual describes the kinetic equations used to model eutrophication, atrazine, chloride, and PCB congeners in Lake Michigan. There is a detailed mathematical description of the ULTIMATE QUICKEST (UQ) transport algorithm that is used in both the water and sediment columns. LM3 does not model hydrodynamics; it uses the hydrodynamics output generated by the application known as the Princeton Ocean Model (POM) as part of its input.

The eutrophication model consists of six carbon species (including two phytoplankton species, one zooplankton species, dissolved organic carbon, refractory organic carbon and labile organic carbon), four phosphorus species, five nitrogen species, and two silica species. The application can model up to 36 PCB state variables. Because some of the PCB state variables represent coeluting congeners, LM3 can account for up to 54 congeners. LM3 can also model atrazine and chloride. The user has the option to model all of the state variables available in the application at the same time or just a subset of them. The sediment bed component of LM3 consists of two carbon state variables, dissolved organic carbon and refractory organic carbon, which serve as the carbon states for PCB congener partitioning. These two carbon states do not undergo any kinetics transformations in the sediment bed. The resuspension velocities are calculated *via* resuspension fluxes output by the SEDZL application, and fed into LM3 as input parameters.

LM3 uses three external libraries for various tasks such as reading input values and writing model output. The three libraries are described. Input file examples for the simulations of atrazine, chloride, eutrophication, and PCBs are included.

This abstract does not necessarily reflect the USEPA policy.

Table of Contents

Notice ii
Foreword iii
Abstract iv
Table of Contents v
List of Figures ix
List of Tables xi
List of Abbreviations xiii
Acknowledgments xv
Executive Summary xvi

Chapter 1 Introduction 1

1.1 Model Background 1
1.2 Model Features 2
1.3 Model Limitations 3
1.4 Model Further Developments 4

Chapter 2 Major Revision History 5

2.1 IPX Revision History 6
2.2 LM3 Revision History 9

Chapter 3 Model Parameters 12

3.1 Parameter Specification 13
3.2 Non-Scalar Input Program Parameters 13
3.3 General Program Parameters 14
3.3.1 General Program Parameter *SCALE 15
3.4 Basic Program Control Option Parameters 15
3.5 Eutrophication and Chemical-Specific Parameters 21
3.6 General Transport Parameters 21

Chapter 4 LM3 New Features 24

4.1 New Features 24

Chapter 5 Water Column Transport 29

5.1 Model Grid 30
5.2 The ULTIMATE QUICKEST Transport Scheme 31
5.3 Vertical Implicit Transport 33

Chapter 6 Eutrophication Kinetics 35

6.1 Development of Eutrophication Equations 35
6.1.1 Phytoplankton Growth 36
6.1.1.1 Nutrient Growth Dependency 37
6.1.1.2 Temperature Growth Dependency 41
6.1.1.3 Light Growth Dependency 41
6.1.2 Zooplankton Kinetics 44
6.1.3 Carbon Interactions 45
6.1.3.1 Phytoplankton Mortality and Decay 45
6.1.3.2 Zooplankton Mortality and Decay 47
6.1.3.3 Hydrolysis and Mineralization 47
6.1.4 Phosphorus 49
6.1.4.1 Phosphorus Uptake by Phytoplankton 49
6.1.4.2 Phosphorus Release Due to Phytoplankton Metabolism/Mortality 49
6.1.4.3 Phosphorus Release Due to Zooplankton Predation 50
6.1.4.4 Phosphorus Release Due to Zooplankton Mortality 51
6.1.4.5 Hydrolysis and Mineralization of Phosphorus 51
6.1.5 Nitrogen 52
6.1.5.1 Nitrogen Uptake by Phytoplankton 53
6.1.5.2 Nitrogen Release Due to Phytoplankton Metabolism/Mortality 53
6.1.5.3 Nitrogen Release Due to Zooplankton Predation 54
6.1.5.4 Nitrogen Release Due to Zooplankton Mortality 54
6.1.5.5 Hydrolysis and Mineralization of Nitrogen 55
6.1.5.6 Nitrification 56
6.1.6 Silica 57

Chapter 7 Sediment Transport 59

7.1 Outline of the Bottom Shear Stress Calculation 60
7.1.1 Introduction 60
7.1.2 Total Bed Shear Stress 60
7.1.3 Friction Factors 64
7.1.3.1 Current Friction Factor 64
7.1.3.2 Wave Friction Factor 66
7.1.4 Calculation Scheme 66
7.1.5 Figures 67
7.2 PCB Diffusion Process Between Sediment and Water 81
7.3 LM3 Sediment Bed 82
7.3.1 Initial Conditions 83
7.3.1.1 ROC Sediment Concentrations 83

7.3.1.2 Clay Sediment Concentrations 84
7.3.1.3 PCB Congener Sediment Concentrations 84
7.3.1.4 DOC Sediment Concentrations 84
7.3.2 Sediment Parameters 84
7.3.2.1 Particle Densities 84
7.3.2.2 PCB Congener Partition Coefficients 84
7.3.2.3 Porosities 84
7.3.3 Boundary Conditions 85
7.3.3.1 ROC Sediment Boundary Condition 85
7.3.3.2 PCB Congener Sediment Boundary Condition 85
7.3.3.3 Clay Sediment Boundary Condition 85
7.3.4 Sediment Transport Data 85
7.3.4.1 Resuspension Velocities 87
7.3.4.2 Bulk Mixing Coefficients 89

Chapter 8 Polychlorinated Biphenyl Kinetics 91

8.1 PCB Partitioning 91
8.2 PCB Volatilization 100

Chapter 9 Atrazine Kinetics 108

9.1 Atrazine Lumped First-Order Decay 109
9.2 Atrazine Volatilization 111
9.2.1 Henry's Law Constant 111
9.2.2 O'Connor Formulation 112
9.2.2.1 Kinematic Viscosity and Density of Air 112
9.2.2.2 Diffusivity of Atrazine in Air and in Water 113
9.2.2.3 Shear Velocity Iteration Scheme 113
9.2.2.4 Contributing Coefficients $Z_0(U^*)$, $\Gamma(U^*)$, K_δ, and K_τ to the Liquid-Phase Transfer Coefficient K_L 114
9.2.2.5 Liquid-Phase Transfer Coefficient 116
9.2.2.6 Gas-Phase Transfer Coefficient 116
9.2.2.7 Overall Mass Transfer Coefficient 117
9.2.2.8 Volatilization Rate 118
9.2.2.9 Steps of the O'Connor Algorithm 118
9.2.3 Wanninkhof and Schwarzenbach Formulation 119

Chapter 10 Program Code, Organization, and Management 121

10.1 The LM3 Algorithm 121
10.1.1 LM3 Individual Program Units 126
10.1.1.1 Main Program 126
10.1.1.2 Initialization and Input Subroutines 127
10.1.1.3 Simulation Execution and Output Subroutines 130
10.1.1.4 Finalizing Subroutines 134
10.1.1.5 Modules 134
10.1.1.6 Generic Subroutines 137
10.1.1.7 INCLUDE Files 137

10.2 LM3 Source and Application Management 138
10.3 LM3 Support Libraries 139

Chapter 11 Data Structures 145

11.1 Input Data Structures 145
11.1.1 Initial Conditions Data Structure 148
11.1.2 Hydrodynamics Data Structure 148
11.1.3 Loads Data Structure 153
11.1.3.1 Load Location Structure 153
11.1.3.2 Load Mass Values Structure 153
11.2 Output Data Structures 155
11.2.1 Timestamps 159
11.2.2 Hydrodynamics 159
11.2.3 Shear Calculations 160
11.2.4 Eutrophication 160
11.2.5 PCB Congeners 161
11.2.6 Clay 168
11.2.7 Atrazine 168
11.2.8 Chloride 175
11.2.9 Sediment 175

Chapter 12 Support Utilities 180

12.1 Editing and Filling CDF Structures 180
12.2 Examining Program Output 182

Chapter 13 Examples 185

13.1 Submitting LM3 Simulations 185
13.2 LM3 Example Atrazine Calculation 186
13.3 LM3 Example Chloride Calculation 188
13.4 LM3 Example Eutrophication Calculation 190
13.5 LM3 Example PCB Calculation 197
13.6 Example of a2cdf Calculation 205
13.7 Example of x2cdf Calculation 208

Chapter 14 References 210

Appendix A Development of a Hydrodynamic/Water Quality (POM/IPX) Linkage for the Lake Michigan Mass Balance Project 218
Appendix B The FIREAD Primer: Format Independent Reading of Program Input Parameters 269
Appendix C UT Version 1.3.1 288
Appendix D An RCS Cheat Sheet 295
Appendix E Derivation of the ULTIMATE QUICKEST Method in One Dimension 302

List of Figures

2.1 IPX revision history 7
2.2 LM3 revision history 10
3.1 Logical associations among the LM3 program flags 20
5.1 The LMMBP high-resolution grid 30
7.1 Locations for the three selected areas where shear stress was calculated 68
7.2 Distribution of shear stress for location A, 1982, k_N = 0.2 cm 69
7.3 Distribution of shear stress for location A, 1982, k_N = 1.0 cm 69
7.4 Distribution of shear stress for location A, 1982, k_N = 2.0 cm 70
7.5 Distribution of shear stress for location A, 1982, k_N = 5.0 cm 70
7.6 Distribution of shear stress for location B, 1982, k_N = 0.2 cm 71
7.7 Distribution of shear stress for location B, 1982, k_N = 1.0 cm 71
7.8 Distribution of shear stress for location B, 1982, k_N = 2.0 cm 72
7.9 Distribution of shear stress for location B, 1982, k_N = 5.0 cm 72
7.10 Distribution of shear stress for location C, 1982, k_N = 0.2 cm 73
7.11 Distribution of shear stress for location C, 1982, k_N = 1.0 cm 73
7.12 Distribution of shear stress for location C, 1982, k_N = 2.0 cm 74
7.13 Distribution of shear stress for location C, 1982, k_N = 5.0 cm 74
7.14 Distribution of shear stress for location A, 1994, k_N = 0.2 cm 75
7.15 Distribution of shear stress for location A, 1994, k_N = 1.0 cm 75
7.16 Distribution of shear stress for location A, 1994, k_N = 2.0 cm 76
7.17 Distribution of shear stress for location A, 1994, k_N = 5.0 cm 76
7.18 Distribution of shear stress for location B, 1994, k_N = 0.2 cm 77
7.19 Distribution of shear stress for location B, 1994, k_N = 1.0 cm 77
7.20 Distribution of shear stress for location B, 1994, k_N = 2.0 cm 78
7.21 Distribution of shear stress for location B, 1994, k_N = 5.0 cm 78
7.22 Distribution of shear stress for location C, 1994, k_N = 0.2 cm 79
7.23 Distribution of shear stress for location C, 1994, k_N = 1.0 cm 79
7.24 Distribution of shear stress for location C, 1994, k_N = 2.0 cm 80
7.25 Distribution of shear stress for location C, 1994, k_N = 5.0 cm 80
7.26 Bottom boundary condition algorithm 86
7.27 Resuspension velocity calculation algorithm 90
9.1 Structural formula for atrazine 108
10.1 Topmost flow of control for the LM3 program 122
10.2 Flow of control for LM3 initialization 122
10.3 Flow of control for the LM3 main or batch level calculation 123
10.4 Flow of control for the LM3 eutrophication calculation 123
10.5 Flow of control for the LM3 finishing level calculation 124
11.1 The CDF rVariable data representation scheme 146

11.2 The CDF zVariable data representation scheme 147
12.1 The cdfedit tool main window 183
12.2 The cdfedit tool zEntries window 183
12.3 The cdfexport tool window 184
E.1 Cell geometry and notation used in this derivation 310
E.2 Interpolation of concentration used for the case u > 0 317
E.3 Interpolation of concentration used for the case u < 0 320

List of Tables

3.1 *FILE Key File Specifications 14
3.2 General Program Parameters 16
3.3 States for General Program Parameter *SCALE 17
3.4 Load Types for Program Parameter *SCALE 18
3.5 *DO Key General Program Parameters 19
3.6 Water Transport Parameters 22
3.7 Sediment Transport Parameters 22
6.1 Eutrophication Parameters 38
8.1 PCB Congener Groups Partition Coefficients 94
8.2 PCB Congener Groups Enthalpy Values 102
8.3 PCB Congener Groups Entropy Values 103
8.4 PCB Congener Groups Molecular Weight 106
9.1 Atrazine Parameters 110
11.1 Eutrophication Water Column Initial Conditions rVariables 149
11.2 PCB Water Column Initial Conditions rVariables 149
11.3 Atrazine, Chloride, and Clay Water Column Initial Conditions rVariables 150
11.4 Eutrophication Sediment Initial Conditions rVariables 150
11.5 PCB Sediment Column Initial Conditions rVariables 151
11.6 Input Hydrodynamics zVariables 152
11.7 Load Location gAttributes 154
11.8 Load File Update Time zVariables 156
11.9 Load File Mass Load zVariables 157
11.10 FIREAD Eutrophication Load Names and Recommended Input Files 158
11.11 FIREAD PCB Load Names and Recommended Input Files 158
11.12 FIREAD Atrazine, Chloride, and Clay Load Names and Recommended Input Files 159
11.13 Output Simulation Time zVariables 159
11.14 Output Hydrodynamics zVariables 160
11.15 Shear Calculation zVariables 160
11.16 Eutrophication State Concentration zVariables 161
11.17 Lake-wide Eutrophication State Mass and Load zVariables 162
11.18 Eutrophication Settling Losses and Rates zVariables 163
11.19 Phytoplankton Growth Limitation Factor zVariables 163
11.20 PCB Congener Total Water Concentration zVariables 164
11.21 PCB Congener Dissolved Water Concentration zVariables 165
11.22 PCB Congener Particulate Water Concentration zVariables 166
11.23 PCB Congener Water Column Mass zVariables 167
11.24 PCB Congener Load Mass zVariables 169
11.25 PCB Congener Settling Loss zVariables 170
11.26 PCB Congener Settling Loss Rate zVariables 171

11.27 Volatilization Absorbed Mass Output zVariables 172
11.28 Volatilization Loss Mass Output zVariables 173
11.29 Net Volatilization Mass Output zVariables 174
11.30 Clay zVariables 174
11.31 Atrazine zVariables 175
11.32 Chloride zVariables 176
11.33 Sediment Eutrophication Output zVariables 176
11.34 Sediment PCB Output zVariables 177
11.35 Sediment PCB Dissolved Output zVariables 178
11.36 Sediment PCB Particulate Output zVariables 179
11.37 Sediment Output zVariables 179
12.1 x2cdf FIREAD Input Key Definitions 181
A1 POMGL.OUTPUT Standard Output File 234
A2 IPXMT Control File 235
A3 MAP.DIA Listing 237
C1 FORTRAN Callable Array Manipulation Subroutines 293
C2 FORTRAN Callable Time Conversion Subroutines 293
C3 FORTRAN Callable Array Manipulation Functions 293
C4 C Callable Utility Functions 294
C5 REPORT Subroutines 294
C6 FIREAD Subroutines 294
E.1 FORTRAN Variables Used in u > 0 Calculation of Advection and Diffusion Multipliers 307
E.2 FORTRAN Variables Used in u < 0 Calculation of Advection and Diffusion Multipliers 308
E.3 FORTRAN Variables Used in Both u > 0 and u < 0 Transport Calculations 309

List of Abbreviations

CDF	Common Data Format
CE-QUAL-ICM	Corps of Engineers Water Quality Integrated Compartment Model
CGS	Centimeter-gram-second
CO_2	Carbon dioxide
cpp	C Preprocessor
CPU	Central processing unit
CSC	Computer Sciences Corporation
DOC	Dissolved organic carbon
DON	Dissolved organic nitrogen
DOP	Dissolved organic phosphorus
GBMBP	Green Bay Mass Balance Project
GLERL	Great Lakes Environmental Research Laboratory
GLNPO	Great Lakes National Program Office
HM	Hydrodynamic model
ICM	Chesapeake Bay model
IPX	In-place Pollutant eXport
JDN	Julian day numbers
LDC	Labile detrital carbon
LLRS	Large Lakes Research Station
LMMBP	Lake Michigan Mass Balance Project
LOC	Labile organic carbon
LON	Labile organic nitrogen
LOP	Labile organic phosphorus
LPOC	Labile particulate organic carbon
LPON	Labile particulate organic nitrogen
LPOP	Labile particulate organic phosphorus
m	Meters
MKS	Meter-kilogram-second
NASA	National Aeronautics and Space Administration
NH_4	Ammonia-nitrogen
NO_3	Nitrate-nitrogen
NOAA	National Oceanic and Atmospheric Administration
NSSDC	National Space Science Data Center
ODC	Organic detrital carbon
ORD	Office of Research and Development
PCBs	Polychlorinated biphenyls
PDC	Particulate detrital carbon
POC	Particulate organic carbon

POM	Princeton Ocean Model
PON	Particulate organic nitrogen
POP	Particulate organic phosphorus
RCS	Revision Control System
RDC	Refractory detrital carbon
ROC	Refractory organic carbon
RON	Refractory organic nitrogen
ROP	Refractory organic phosphorus
RPOC	Refractory particulate organic carbon
RPON	Refractory particulate organic nitrogen
RPOP	Refractory particulate organic phosphorus
SA	Dissolved silica
Si	Silica
SRP	Soluble reactive phosphorus
SU	Biogenic silica
TP	Total phosphorus
UQ	ULTIMATE QUICKEST
USACE	United States Army Corps of Engineers
USEPA	United States Environmental Protection Agency
USGS	United States Geological Survey
WASP	Water Quality Analysis Simulation Program
WDNR	Wisconsin Department of Natural Resources
WES	Waterways Experiment Station
WQCMB	Water Quality and Contaminant Modeling Branch
WQM	Water quality model

Acknowledgments

The United States Environmental Protection Agency, Great Lakes National Program Office provided leadership, support, and collaboration on the Lake Michigan Mass Balance Project. Carl Cerco, U.S. Army Corps of Engineers, contributed to the eutrophication section. Douglas Endicott, formerly with the U.S. EPA, Large Lakes Research Station, contributed to the implementation of the polychlorinated biphenyl kinetics, the design of the sediment bed model, the calculation of resuspension fluxes, and the overall guidance and moral support. Ronald Rossmann, U.S. EPA, Large Lakes Research Station, provided an editorial review and comments.

Executive Summary

This report is a user's manual for the application known as LM3. LM3 is a water quality mass balance model that has the capability to simulate eutrophication, atrazine, chloride, and polychlorinated biphenyl (PCB) congeners. The development of the current model started in the mid-1990s. LM3 has continued to be developed by adding an ecosystem model. However, as of this writing, the ecosystem model is still being tested. This report only describes the stage of the application that simulates the aforementioned systems.

LM3 has a high-resolution grid with cells that have a horizontal area of 5 km x 5 km. There are 2,318 cells across the lake and 19 sigma layers in the water column. Each sigma layer contains the same number of cells across, 2,318, which gives a total of 44,042 grid cells in the water column for the entire lake. Each sigma layer varies in thickness depending on the water column depth. In nearshore areas, where water depth can be as low as 2.5 meters, some of the sigma layers can have thicknesses under 0.1 meters. In offshore areas, where water depth can exceed 250 meters, some of the sigma layers can have thicknesses exceeding 30 meters.

The first component of the model to be implemented was eutrophication. The eutrophication kinetics are based on the Corps of Engineers Water Quality Integrated Compartment Model (CE-QUAL-ICM) formulations (Cerco and Cole, 1995), which were previously applied to Chesapeake Bay (Cerco and Cole, 1993). However, several changes were made, including improvements to the solar radiation equations, changes to the phosphorus transformation reactions, and the addition of zooplankton equations. The eutrophication model simulates 17 state variables: six carbon states which include two phytoplankton and one zooplankton groups, four phosphorus species, five nitrogen species, and two silica species. The phytoplankton and zooplankton groups are modeled as carbon states with fixed carbon-to-nutrient ratios to account for their content of phosphorus, nitrogen, and silica. Loadings and initial concentrations are specified as input parameters for each of the eutrophication states. There is a user-defined sediment flux including dissolved organic carbon, soluble reactive phosphorus, ammonia, and silica to account for sediment feedback in the model. LM3 does not have a sediment bed model for eutrophication; that is, no diagenesis model has been incorporated as of this writing.

LM3 uses the ULTIMATE QUICKEST (UQ) method as the transport scheme in the model. A derivation of this method is provided for the case of one spatial dimension. The transport scheme was adapted by the U.S. Army Corps of Engineers (Chapman *et al.*, 1997) for use in the LM3 application. The UQ method uses the flow and diffusion values that are generated using the Princeton Ocean Model (POM) (Schwab and Beletsky, 1998). LM3 does not simulate hydrodynamics; instead, it uses the output from POM as input.

Chloride is one of the chemicals that can be modeled in LM3. It is the simplest case to simulate because there are no kinetics and no sediment bed to track. Atrazine is another modeled chemical in the application, but unlike chloride, it can decay and volatilize. Atrazine's decay is a lumped first-order one because the biological and chemical processes that play a role in the chemical's breakdown in the water column are still not fully understood. The user has the option of choosing one of two volatilization schemes available in LM3 for atrazine: the O'Connor formulation (O'Connor, 1983a; 1983b) and the Wanninkhof and Schwarzenbach formulation (Schwarzenbach *et al*., 1993; Wanninkhof *et al*., 1991; Wanninkhof, 1992). Both atrazine and chloride can be simulated individually without having to turn on the eutrophication and PCB algorithms.

LM3 has the capability to model individual PCB congeners. The user has the ability to specify up to 36 PCB state variables. Each state variable represents either a single congener or coeluting congeners. Because some of the state variables represent coeluting congeners, LM3 can account for up to 54 congeners. Each PCB congener (or coeluting ones) is modeled as seven different phases in the water column and three different phases in the sediment column. The seven phases in the water column represent the six carbon states the PCB congener is binding to, and the purely dissolved state of the congener. The three phases in the sediment column represent the two carbon states the congener is binding to, and the purely dissolved state of the congener. The user has to specify two partition coefficients for each PCB congener: one partition coefficient corresponds to the partitioning of the congener to dissolved organic carbon (DOC) and the other corresponds to the partitioning to the five particulate organic carbon (POC) states. The LM3 model assumes that the partition coefficients in the water and sediment columns are the same.

PCB volatilization was implemented in LM3. The Wanninkhof and Schwarzenbach volatilization formulation (Schwarzenbach *et al*., 1993; Wanninkhof *et al*., 1991; Wanninkhof, 1992) was implemented for PCBs. The user has the ability to turn on or off the volatilization. Each PCB congener has its own set of values for entropy, enthalpy, and molecular weight that the user specifies in the input file. A PCB diffusion process between sediment and water was also implemented.

In LM3 a sediment bed construct was implemented, which is primarily designed to look at PCB transport between the water and sediment column. The sediment bed has a geometry that lines up with the water column, with 2,318 grid cells across and layers going down the sediment column that are user-specified. Each sediment grid cell has the same horizontal cross-sectional area as the ones in the water column, 5 km x 5 km. The sediment bed construct is relatively simple. It has only two carbon states, no resuspension velocities, no diffusion between sediment layers, constant cell volumes, constant solid concentrations, and only vertical transport. The vertical transport is described by a simplified UQ. The resuspension velocities are calculated from resuspension fluxes that are input into LM3. The resuspension fluxes were calculated outside LM3 using a separate application known as SEDZL. LM3 does have an algorithm that calculates shear stress that can potentially be used to calculate resuspension velocities internally, provided the current sediment bed construct is improved upon. Clay was added as a state variable in support of a future improvement upon the sediment bed construct but has not been used as of this writing. The user has the option to turn off the sediment bed calculation if wishing to do water column calculations only.

LM3 uses three external libraries for various tasks such as input and output. The UT library handles primarily the processing of the input file by parsing over format-independent parameters and filenames. It also handles the printing of diagnostic messages to standard output. The SNIPPETS library provides some basic scientific calculations such as sunrise and sunset times, and day length for a given time of the year that are used within the eutrophication model. The CDF library handles the input and output of large binary datasets. The binary files are created using a format that is independent of the computer platform being used. This allows the user to utilize the same CDF files on many operating systems without having to create duplicate files for each operating system.

CHAPTER 1

INTRODUCTION

This report is a user's manual of the LM3 water quality model. The LM3 water quality model was developed to primarily study eutrophication, polychlorinated biphenyls (PCBs), atrazine, and chloride processes in Lake Michigan as part of the Lake Michigan Mass Balance Project (LMMBP) (Rossmann, 2006). The LMMBP was established, organized, and set in operation by the United States Environmental Protection Agency's (USEPA) Great Lakes National Program Office (GLNPO). The LMMBP main objectives were to measure and simulate concentrations of contaminants prevalent in Lake Michigan (Rossmann, 2006). The contaminants considered were PCBs, *trans*-nonachlor, atrazine, chloride, and mercury. Of the aforementioned contaminants, only PCBs, atrazine, and chloride are described in this report because those are the ones that have been incorporated into the LM3 water quality model as of this writing. This report also contains a description of the LM3 eutrophication model.

1.1 Model Background

The LM3 water quality application evolved from earlier projects that dealt with studies of the Lower Fox River (Velleux *et al*., 2001), Green Bay (DePinto *et al*., 1993), and Chesapeake Bay (Cerco and Cole, 1995). The original application of the Lower Fox River was based on WASP4 (Ambrose *et al*., 1988). During the course of the study of the Lower Fox River, WASP4 was modified to enhance sediment transport and to allow the simultaneous simulation of an unlimited number of contaminants. These WASP4 modifications led its evolution into the In-place Pollutant eXport (IPX) model. IPX was designed to simulate the transport and fate of sediments and associated hydrophobic contaminants in tributaries. It is based on the key principle of conservation of mass and uses the Euler's method to solve the mass balance differential equations.

IPX was later used as a framework to develop an application that could be used to study PCBs and other contaminants in Lake Michigan. New source code modifications were implemented by adding organic carbon dynamics that have been previously used in the Green Bay Mass Balance Project (GBMBP) (DePinto *et al*., 1993). At this stage, the model was still known as IPX but under a different version number to distinguish it from the original IPX used for tributaries. The number of solids in the model was only three. Further source code modifications were carried out to expand the number of solids used in the model and to modify the segmentation grid used for Lake Michigan in support of the LMMBP. At this point, an eutrophication model and a new transport scheme using ULTIMATE QUICKEST (UQ) were

implemented within this version of IPX. This Lake Michigan version of IPX was then renamed LM3.

Algorithms for PCB congeners, atrazine, and chloride were later implemented into LM3. LM3 has undergone further development that now includes an ecosystem model. However, a description of the ecosystem model is not included in this report.

1.2 Model Features

The LM3 water quality model uses a high-resolution grid. This grid consists of 2,318 horizontal cells and 19 vertical sigma layers. Each sigma layer contains the same number of horizontal cells which results in a total of 44,042 cells for the entire water column. Each grid cell has a horizontal area of 5 km x 5 km. Each sigma layer has a thickness that varies depending on the water column depth. In nearshore areas, where water depth can be as low as 2.5 meters, some of the sigma layers can have thicknesses under 0.1 meters. In offshore areas, where water depth can exceed 250 meters, some of the sigma layers can have thicknesses exceeding 30 meters.

An eutrophication model (LM3-Eutro) has been implemented by Pauer (Pauer *et al*., 2006) that dynamically simulates three plankton species, three detrital carbons, and 11 nutrient states. It should be mentioned that the core of the transport algorithm and eutrophication kinetic calculations were borrowed from the Chesapeake Bay simulation model (ICM) developed and applied by the United States Army Corps of Engineers (USACE) at the Waterways Experiment Station (WES) in Vicksburg, Mississippi (Cerco and Cole, 1993; 1994; 1995).

LM3 has the ability to output biotic carbon loads that can used by applications that do not simulate eutrophication such as the LM2-toxic chemical model.

The user has two options when dealing with the time step of the model simulation. One option is to use the hydrodynamic update interval as a constant time step. The user specifies this option by setting a flag in the input declaration file. The second option does not require the user to specify a time step; instead it is internally calculated using an algorithm based on horizontal transport. This internally calculated time step is variable in time because it is derived from the Courant stability constraint on horizontal transport.

LM3 has a built-in transport algorithm based on UQ. Particle settling velocities are part of the transport calculation and not the eutrophication kinetics.

PCB kinetics were implemented by Endicott (Endicott *et al*., 2006) and Wilson Melendez (Computer Sciences Corporation [CSC]) that dynamically calculate volatilization and partitioning. The PCBs run on top of the eutrophication kinetics because they partition to the carbon states. Each PCB congener is modeled in the water column as seven phases: one of the phases corresponds to the purely dissolved state, another to the partition of the PCB congener with respect to dissolved organic carbon (DOC), and the remaining five phases correspond to the partitioning of the PCB congener with respect to the five particulate carbon states of LM3-Eutro.

A sediment transport algorithm was developed by Ray Chapman and Terry Gerald of the USACE at WES for the LM3 application in support of a PCB model. The sediment transport

algorithm was designed according to specifications submitted by Douglas Endicott (Great Lakes Environmental Center). These specifications include fixed sediment volume cells and a fixed carbon-solid concentration gradient among others. The sediment transport source code was incorporated into LM3 by Wilson Melendez (CSC). Each PCB congener is modeled as three phases in the sediment column: one of the phases corresponds to the purely dissolved state, one corresponds to the partitioning with respect to DOC, and the third corresponds to the partitioning with respect to refractory organic carbon (ROC). The present sediment algorithm accounts for only two carbon states: DOC and ROC.

Atrazine and chloride models were implemented into LM3. The atrazine model includes calculations for volatilization and lumped-sum decay. Chloride is a conservative chemical with only advective transport affecting its concentration. Both chemicals can be simulated independently of one another and of PCBs and eutrophication.

The application can output concentrations, growth rate limitation factors, and mass balance information on two grid resolutions: the high-resolution grid (which is also the computational grid) and a course grid. The course grid, or low-resolution grid, consists of 10 surficial segments with surface areas ranging from $3.5 \cdot 10^8$ to $1.3 \cdot 10^{10}$ m^2 and of up to five vertical layers with thicknesses ranging from two to 84 meters. Because model output is exclusively derived from calculations performed on the high-resolution grid, the data that is generated on the low-resolution grid is actually data that is aggregated from the high-resolution onto the low-resolution grid.

LM3 can run simulations for as many years as desired. The model is currently using the hydrodynamics output generated for the LMMBP study period of 1994-1995. From the 1994-1995 study period, the hydrodynamic output covers the period from January 1, 1994 to December 21, 1995. If simulations extend beyond December 21, 1995, the program uses the hydrodynamic data of the last 10 days of 1994 for the missing last 10 days of 1995 in order to complete the calendar year of 1995. Once the program reaches the end of 1995, it goes back to the beginning of the hydrodynamic output, January 1, 1994. The program re-reads the output, from that point on, going all the way through December 21, 1995. To complete the calendar date of 1995, the data corresponding to the last 10 days of 1994 are used as surrogate values for the last 10 missing days of 1995 once again. This entire process is repeated as many times as needed to cover the desired number of simulation years.

1.3 Model Limitations

LM3 does not compute hydrodynamics and instead uses the flows, diffusion coefficients, volumes, and water temperature output generated by the high-resolution hydrodynamics model known as the Princeton Ocean Model (POM) (Schwab and Beletsky, 1998) as input for the transport algorithm section of the program.

No dissolved oxygen model has been implemented in the application as of this writing.

The LM3 sediment algorithm does not include a calculation of resuspension fluxes; the fluxes are, instead, calculated and output using the application known as SEDZL (Ziegler and Lick, 1986; Lick *et al*., 1987; Gailani *et al*., 1991) and then, subsequently input into LM3 where they are used by the sediment algorithm.

LM3 does not contain a diagenesis model in the sediment bed as of this writing.

1.4 Model Further Developments

The present LM3 manual does not include a detailed description of the ecosystem model development presently done on the LM3 branches of 3.2.16, 3.2.17, and 3.2.20. There is a brief description of some of the work that has been done to date for the ecosystem model in Chapters 2 and 4. The kinetics equations of the ecosystem model are going to be included in another release of the LM3 manual in the future.

CHAPTER 2

MAJOR REVISION HISTORY

LM3 has a long revision history, tracing its origins to an older application known as IPX (In-place Pollutant eXport). The original IPX application was designed to simulate the transport and fate of sediments and associated hydrophobic contaminants in tributaries. One of the primary goals of this application was to study polychlorinated biphenyls (PCBs) in the lower Fox River. The Fox River has long been known to be a source of pollutants to Lake Michigan, in particular PCBs.

Because of the presence of PCBs in Lake Michigan and its impact on the water quality of the lake, it became imperative to develop a model to study the fate and transport of these chemicals within the lake. Hence, it was decided to use IPX as a starting point for a model that would simulate PCBs and other pollutants in Lake Michigan. IPX underwent a series of source code modifications that eventually led its evolution into the present day LM3.

A revision history of IPX and LM3 is presented here that fully illustrates the source code development of these applications. The source code development has been archived using the source control tool known as RCS (Revision Control System). RCS uses a particular version numbering system to keep track of source code development. To understand the version numbers shown in the revision history, a brief explanation of RCS revision numbers is needed. RCS assigns a revision number to each version of a file that is archived. This number has the format L.U.B.S., for level, update, branch, and sequence IDs, respectively. The number L represents the level of the original archive file; this level is the main branch, also known as the trunk, of the application. The second number U represents a specific revision of the main branch (trunk). The third number B represents a branch of the archive file L.U. The last number S represents a specific revision of the branch L.U.B. A one- or three-part number (L or L.U.B.) is a branch specifier, which names a branch without naming a revision to it. A two- or four-part number (L.U. or L.U.B.S.) is a revision specifier, which names exactly one revision. The branching processes can actually be continued to any number of levels forming a tree-like logical structure. However, in practice, more than two levels of branching are rarely needed.

Throughout the discussion that follows, only two- and three-part numbers of the revision history are shown. Four-part numbers correspond to revisions made to the source code dealing with bug fixes and minor changes that are not relevant to the description of the evolution of IPX and LM3. The IPX revision history is shown first followed by the history of LM3. In this way it will be clear how the evolution of IPX led to the development of LM3.

2.1 IPX Revision History

A detailed description of the revision history of the application IPX is given below. A diagram depicting the relationship between revisions and branches is shown in Figure 2.1.

- Version 1.1: Major revision released by Large Lakes Research Station (LLRS) in August, 1995: original VAX version of the code. This revision is not found in RCS.

- Version 2.1: Major revision released by LLRS in November, 1995: variable declarations. This revision is not found in RCS.

- Version 2.5: Major revision released by LLRS in November, 1995: reformat of code including internal documentation. First revision archived in RCS.

- Version 2.6: Major revision released by LLRS in November, 1995: major bug fix to buildup of system concentrations on boundary segments.

- Version 2.7: Major revision released by LLRS in November, 1995: RCS keywords included in all FORTRAN source files. This revision has four branches: 2.7.1, 2.7.2, 2.7.3, and 2.7.4 which are detailed below.

- Version 2.7.1: Minor revision branch. Several files with extension *.cmn were renamed.

  Version 2.7.2: Major revision branch. Several INCLUDE files were added and others were renamed. Several subroutines were renamed, several were deleted, and new ones were added.

- Version 2.7.3: Minor revision branch. It contains modifications to support parallel v-2.7.3 development branch with Wisconsin Department of Natural Resources (WDNR) Lagrangian sediment transport. Source files contain *.I2.* to distinguish them from later revisions with the same name.

- Version 2.7.4: Minor revision branch. It contains hardpan sediment WDNR modifications.

- Version 2.8: Major revision released by LLRS in February, 1996: it includes message and error reporting and several bug fixes. This version has two branches: 2.8.1 and 2.8.2.

- Version 2.8.1: Minor revision branch. New subroutines were added. INCLUDE files were split into smaller files.

- Version 2.8.2: Minor revision branch. It contains a HistoScope implementation test.

- Version 3.0: Major revision released by LLRS in June, 1996: it includes cpp INCLUDE statements, improved UNIX make capabilities, HistoScope output, and bug fixes. Source files have been renamed with the file extension *.F. This revision has the branches 3.0.1, 3.0.2, 3.0.3, and 3.0.4 which are detailed below.

- Version 3.0.1: Minor revision branch including modification of the IPX input format and use of the UT library FIREAD input format.

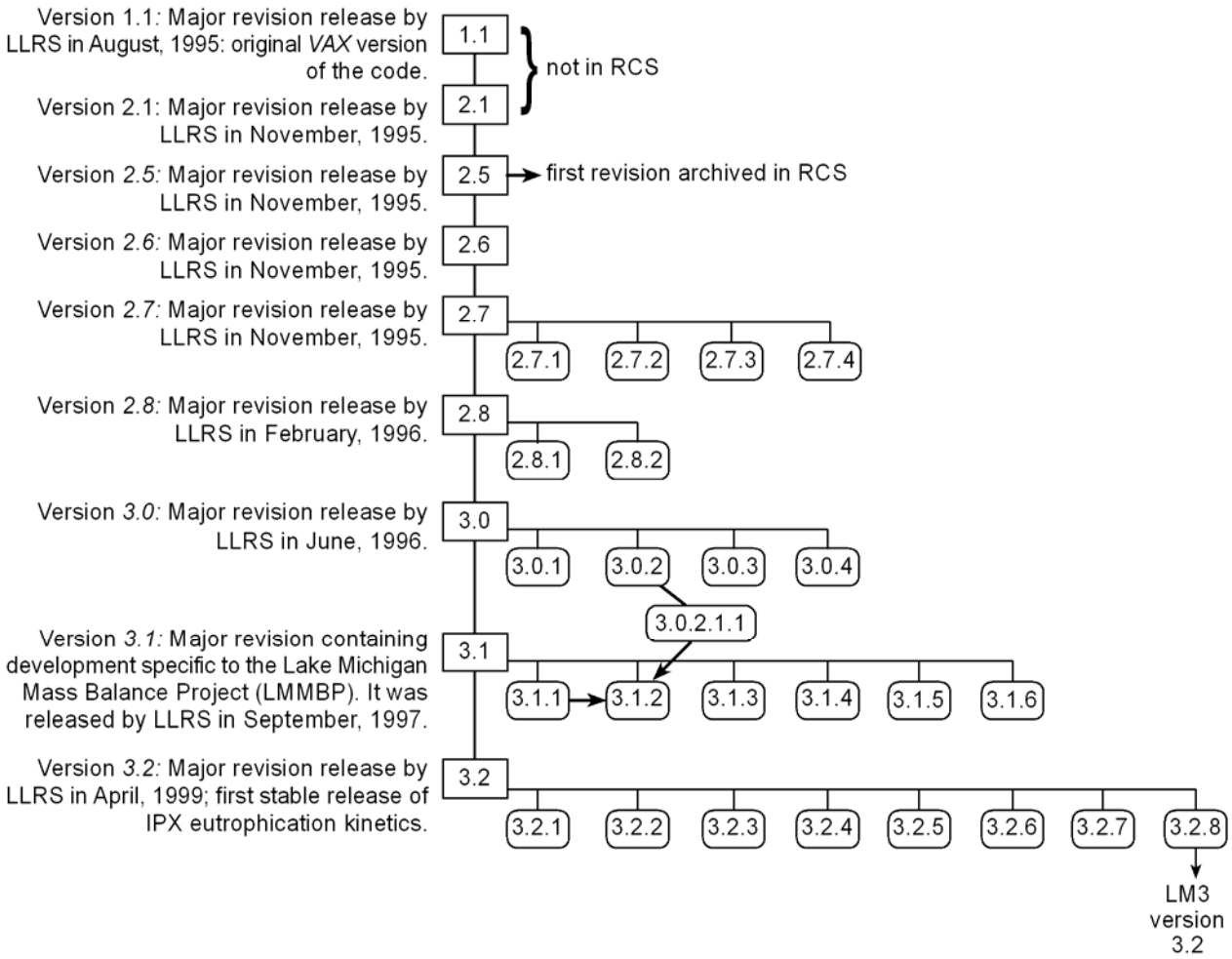


Figure 2.1. IPX revision history. The square boxes represent the IPX major revisions in RCS. The rounded boxes represent branches of the revisions. Note that revisions 1.1 and 2.1 are not stored in RCS.

- Version 3.0.2: Minor revision branch dedicated to the inclusion of organic carbon sorbent dynamics as were used in the Green Bay study. This branch has the sub-branch 3.0.2.1.1, which is detailed below.

- Version 3.0.2.1.1: Modifications to the IPX advective transport algorithm making use of ULTIMATE QUICKEST (UQ) algorithm.

- Version 3.0.3: Minor revision branch with code modifications made by Xia Xiangsheng (Computer Sciences Corporation [CSC]): inlining of several of the IPX kinetic calculation subroutines.

- Version 3.0.4: Minor revision branch containing modifications to the IPX kinetics calculations: optimizing memory and central processing unit (CPU) performance efficiency.

- Version 3.1: Major revision containing developments specific to the Lake Michigan Mass Balance Project (LMMBP). It was released by LLRS in September, 1997. This revision has six branches: 3.1.1, 3.1.2, 3.1.3, 3.1.4, 3.1.5, and 3.1.6 which are detailed below.

- Version 3.1.1: Minor revision branch containing modifications to the IPX state variable and transport scheme. Removed the WASP F() array approach and substituted a scheme in which the different sorption states are separate state variables; necessary to accommodate the upcoming eutrophication algorithm.

- Version 3.1.2: Minor revision branch containing merger of IPX versions 3.0.2.1.1 (implementation of Waterways Experiment Station [WES] transport) and version 3.1.1 (new state variables and eutrophication kinetics).

- Version 3.1.3: Minor revision branch containing updates to IPX FIREAD I/O: addition of *FILE cards for specification of multiple input data files.

- Version 3.1.4: Minor revision branch containing modifications to IPX state variables; reverted to use of fixed eutrophication state variable indices. Two distinct detrital carbon states (labile and refractory) were implemented.

- Version 3.1.5: Minor revision branch. Separate dM/dt updates from transport calculation; remove ICM parameter I/O; replace w/ FIREAD keys; bug fixes.

- Version 3.1.6: Minor revision branch. Code/algorithm re-organization; added zooplankton kinetics; completed debug of IPX eutrophication kinetics; the Common Data Format (CDF) output; bug fixes.

- Version 3.2: Major revision released by LLRS in April, 1999; first stable release of IPX eutrophication kinetics. This revision has eight branches: 3.2.1, 3.2.2, 3.2.3, 3.2.4, 3.2.5, 3.2.6, 3.2.7, and 3.2.8 which are detailed below.

- Version 3.2.1: Minor revision branch. Space- and time-dependent illumination; wave height, period and direction; write phytoplankton growth limitation factors; CDF formatted initial conditions; bug fixes.

- Version 3.2.2: Minor revision branch. Added bottom shear calculation; bug fixes.

- Version 3.2.3: Minor revision branch. Mass balance calculations and output of mass balance information; added SAND state variable; added ATRAZINE state variable; implemented high-resolution load calculation; bug fixes.

- Version 3.2.4: Minor revision branch. Output of hydrodynamic/shear information.

- Version 3.2.5: Minor revision branch. Modifications were made to zooplankton kinetics; bug fixes.

- Version 3.2.6: Minor revision branch. Lagrangian sediment transport algorithm.

- Version 3.2.7: Minor revision branch. Euler sediment transport algorithm.

- Version 3.2.8: Minor revision branch. New eutrophication default parameters were added for hydrolysis and mineralization. Removed unused zooplankton kinetics code. This is the last revision branch for IPX. The application name was changed to LM3 to distinguish the level 2 and level 3 water quality calculations.

2.2 LM3 Revision History

The first release of LM3 was version 3.2 and was released in April of 1999. The reason for using 3.2 as the first version number for the application LM3 was because it was created out of the IPX revision branch 3.2.8. Note that the first revision branch of LM3 is 3.2.8 and not 3.2.1. The subsequent revision history is detailed below. A diagram depicting the relationship between revisions and branches is shown in Figure 2.2.

- Version 3.2: First LM3 revision released by LLRS in December, 1999: it has combined hydrodynamic transport and eutrophication kinetics. This revision has 14 branches: 3.2.8, 3.2.9, 3.2.10, 3.2.11, 3.2.12, 3.2.13, 3.2.14, 3.2.15, 3.2.16, 3.2.17, 3.2.18, 3.2.19, 3.2.20, and 3.2.21 which are detailed below.

- Version 3.2.8: Implemented eutrophication loads.

- Version 3.2.9: New hydrodynamics input format.

- Version 3.2.10: WES modifications to hydrodynamics to accommodate tributary flows; improved vertical transport and new settling algorithm.

- Version 3.2.11: Restored the Version 3.2.9 water column transport algorithm; enabled processing of tributary flows; and added chloride state variable.

- Version 3.2.12: New modifications to support multi-year simulations. Added shoreline erosion loads. The light limitation factor was modified to account for shorter update intervals.

- Version 3.2.13: New Ray Chapman modifications to the UQ transport routine. This is the same transport routine that Version 3.2.10 has. Settling algorithm is part of the transport calculation. Source code logical flags were added to support a number of total phosphorus (TP) long-term scenarios.

- Version 3.2.14: New modifications to support a 20-year hindcast calibration for the eutrophication.

- Version 3.2.15: New WES sediment transport routine. Atrazine and PCB kinetics were added. Removed the sand state variable and in its place added the clay state variable.

- Version 3.2.16: New enhanced eutrophication. The original single zooplankton species has been broken down into four zooplankton species; these are *Mysis relicta, Bythotrephes longimanus*, carnivorous zooplankton, and herbivorous zooplankton. A stratified user-defined sediment flux was added to mimic sediment transport. This branch does not contain any atrazine or PCB kinetics.

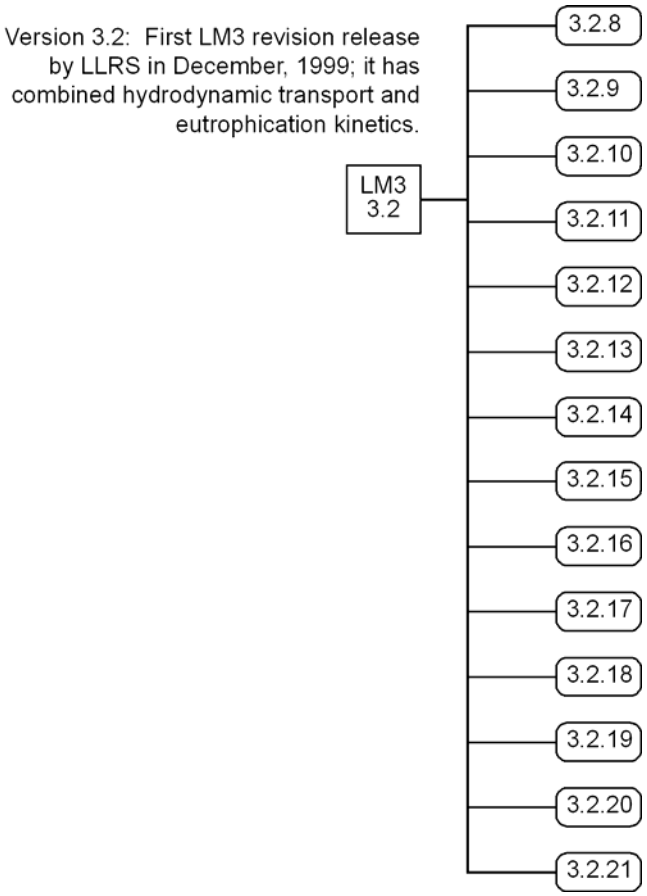


Figure 2.2. LM3 revision history. The square box represents the LM3 major revision in RCS. The rounded boxes represent the branches of the revision.

- Version 3.2.17: Expanded the eutrophication model by breaking down greens into four separate groups. Zebra mussels and *Diporeia* were added as part of the development of an ecosystem model. Sediment compartment is also a stratified user-defined sediment flux. This branch does not contain any atrazine or PCB kinetics. This branch is still being tested as of this writing.

- Version 3.2.18: This branch is an enhancement of 3.2.15. An algorithm was added that minimizes the occurrence of negative concentrations. This algorithm consists of threshold concentrations for dissolved silica, total inorganic phosphorus, nitrate, and ammonia that serve as lower bound values for those states. Whenever the concentrations of the aforementioned states drop below their corresponding user-defined threshold concentrations the program sets the concentrations to zero.

- Version 3.2.19: This branch, an enhanced version of 3.2.15, has some added modifications to support a eutrophication hindcast. There are two hindcast periods considered: 1960-2011 and 1976-2011.

- Version 3.2.20: Enhanced version of 3.2.16 that supports an ecosystem Phase I forecasts. This branch is still being tested as of this writing.

- Version 3.2.21: This revision branch is an enhancement of 3.2.17. It supports forecasts for the ecosystem model that includes zebra mussels and *Diporeia*. This branch is still being tested as of this writing.

CHAPTER 3

MODEL PARAMETERS

Model parameters represent a myriad of empirical data, calibration data, source code flags, and input filenames needed by LM3 to execute properly. This information is specified as input values that are read into LM3 during the execution of the program.

The empirical and calibration data are numerical values needed by LM3 because they are part of the mathematical equations of the model. Empirical data are parameters usually obtained from laboratory experiments or, in some cases, from *in situ* tests. These parameters do not need to be adjusted during the calibration processes of the model. On the other hand, calibration data are parameters that are not well known because of lack of experimental work or difficulty in measuring them. Consequently, modelers have the option of adjusting calibration data, preferably within a specified range, to fit the model output to the observed data.

Source code flags are logical variables that turn sections or subroutines within the program on or off depending on what the user wants to do. They control part of the flow of the model calculation by determining which algorithm should be used for a specific kinetic process, whether a particular process should be executed or not, and so on.

Input filenames are the names of files that contain data that LM3 needs for its execution. The data contained in these files are usually too big to be entered manually so they have been pre-processed and made ready-to-use for the user. Examples of such data are hydrodynamics flows, water temperature, initial conditions, and loads among others.

In the discussion that follows, we present a detailed description of the kind of parameters needed by the LM3 model and the proper syntax to use when specifying them in the input deck file. Section 3.1 discusses the most common syntax used to specify input parameters. Section 3.2 discusses what syntax to use when entering non-scalar parameters. Non-scalar parameters refer to large datasets that are input into LM3 by reading the data from separate files. Section 3.3 deals with general program parameters. Section 3.4 discusses the case of parameters that function as flags that control the activation of certain processes within the model. Section 3.5 briefly addresses the issue of eutrophication and chemical-specific parameters while directing the user to the relevant chapters. Section 3.6 discusses parameters associated with the transport algorithm.

3.1 Parameter Specification

LM3 parameters are specified using a format-independent parameter processing tool known as FIREAD (Settles, 1996). Parameters are read on standard input (stdin), and in most cases, use the syntax:

KEY VALUE

where KEY specifies the name of the parameter. Each key and value pair must appear on its own line. Other than that, no particular format is required. Parameters not specified in the input deck are assigned default values; all parameter values are logged in the output log file and parameters whose default values have been altered are flagged. Key and parameter values can appear in any order in the input deck. FORTRAN-style comment syntax can be used to help document the input deck. An example is provided below.

```
LIST   ! Echo the contents of this file.

OVERWRITE  1          ! Overwrite previous output dumps
CDFCOARSE 1           ! Write coarse (41-segment) concentrations
CDFFINE       0       ! Do not output high-resolution (5 km) concentrations
DD2WRT        2       ! How often to write output

C
C Begin eutrophication parameters
C
CGZ      2.3E-06          ! Zooplankton grazing coefficient
KTGD1  2.5E-03            ! diatom temp coefficient < 20C
KTGG1  2.5E-03            ! greens temp coefficient < 20C

C
C Input files
C
*FILE MAP       /home/llrs_model/dsets/LM3/R0105/map.dat            ! Map
*FILE GEO       /home/llrs_model/dsets/LM3/R0105/geo.dat            ! Geometry
*FILE AGG       /home/llrs_model/dsets/LM3/R0105/map1312.dat        ! Segment aggregation
*FILE HYDRO     /home/llrs_model/dsets/LM3/R0105/hydro94.cdf        ! Hydrodynamics
*FILE ICI       /home/llrs_model/dsets/LM3/R0105/ici.cdf            ! Initial conditions

STOP
```

3.2 Non-Scalar Input Program Parameters

Input program parameters that are not scalar in nature, that is, parameters that require a large set of values, are input using the key *FILE. Typical cases that require the input of large sets of values are initial conditions, hydrodynamics, and loads. See Section 11.1 for a more detailed discussion of these cases.

The *FILE key uses the following FIREAD syntax:

*FILE <value> <filename>

The above line consists of three items. The first item is the key *FILE itself. The second item, <value>, can be any of the keynames listed in Table 3.1. The last item, <filename>, is the name of the file that contains the set of values. The name of the file should be strung together with its pathname so the program will know where to locate it within the Linux system. The format of files can be either ASCII or CDF (Common Data Format).

Table 3.1. *FILE Key File Specifications

Key Name	Description
AGG	Input file for mapping between level 3 and level 2 geometries
AIR_CONC	Input file for chemical vapor concentrations
GEO	Input file for transport scheme cell definitions
HYDRO	Input file for hydrodynamics
ICI	Input file for water column initial conditions
ICI_SED	Input file for sediment bed initial conditions
LDLOC	Loads location file
LOAD	Loads file
L2PROD	Output file name for level 2 biotic carbon load
MAP	Input file for transport scheme flow face definitions
RESUSP_VELOCITIES	Input file for resuspension velocities
VGEO	Input file for sediment transport scheme cell definitions
VHYD	Input file for sediment bed hydrodynamics
VMAP	Input file for sediment transport scheme flow face definitions
WIND	Input file for wind speed values

The key *FILE can be specified more than once in the input deck. The number of times that it appears in the input deck will depend on what the user is trying to simulate. See Chapter 13 for examples of different simulations that show the use of the *FILE key.

3.3 General Program Parameters

There are several parameters that function as general program parameters in the sense that they trigger a major task during the execution of the program. Some of these tasks are associated with output options, eutrophication and/or polychlorinated biphenyl (PCB) kinetics, scaling factors, etc. For example, the key name below will cause the program to print the high-resolution output for each simulated state.

CDFFINE 1

Another example, given below, will cause the program to scale the illumination factor by 10.

ILMUL 10.0

This affects the light limitation factor that is used in the eutrophication kinetics throughout the entire simulation period.

The complete list of general program parameters, with their corresponding choice of values, is given in Table 3.2. All the general program parameters have default values so it is up to the user to decide whether a non-default value should be used for a particular simulation run or not. Note that some of the parameters have associated units and others do not. The parameters CDFCOARSE, CDFFINE, and OVERWRITE can assume only the integer values zero and one; the parameter DD2WRT can assume only integer values that are equal to or higher than one; the parameter SILIM can assume only the integer values of one, two, and three; the parameter *DO assumed only logical flags as defined in Table 3.5; all other parameters assume non-negative floating-point values. Most of the general program parameters should be specified no more than once in the input file. The only exceptions are the parameters *DO and *SCALE, that can be specified multiple times in the input file.

*3.3.1 General Program Parameter *SCALE*

The program parameter *SCALE has a rather complex syntax that requires a brief explanation. This parameter is used when the user wants to scale either initial conditions, loads, or both.

When scaling initial conditions, that is, initial concentration values, the user has to use the syntax shown below in the input file.

*SCALE <value> ICI <state>

There are four items in the line above. The first item is the FIREAD key *SCALE; the second item, <value>, is a non-negative floating number the user specifies as the scaling factor for a given state; the third item is the key name ICI which indicates to the program that the user is scaling water column initial conditions; the last item, <state>, can be any of the states listed in Table 3.3. The example shown below will cause the program to scale the diatom initial water concentrations by 1.5.

*SCALE 1.5 ICI DIA

When scaling loads, the user must use the syntax shown below in the input file.

*SCALE <value> <load_type> <state>

There are four items in the line above. The first item is the FIREAD key *SCALE itself. The second item, <value>, is any non-negative floating value the user wishes to specify as a scaling factor for a particular load type. The third item, <load_type>, can be any of the load types listed in Table 3.4. The last item, <state>, can be any of the states listed in Table 3.3. The example shown below will cause the program to scale the soluble reactive phosphorus (SRP) monitored loads by 100.

*SCALE 100.0 LOAD_MON SRP

3.4 Basic Program Control Option Parameters

Several basic program options are controlled by FIREAD keys using the syntax *DO key name. The *DO key names and their interpretations are given in Table 3.5. Some of the control flags are logically associated with one another and this association is demonstrated in Figure 3.1.

Table 3.2. General Program Parameters

Key Name	Description	Default Value	Units
CDFCOARSE	Selection of coarse segmentation output values; enable with 1, disable with 0	0	None
CDFFINE	Selection of fine segmentation output values; enable with 1, disable with 0	0	None
DD2WRT	Number of days between writes	1	Days
DLT	Default time step when hydrodynamic transport is not specified	4.32×10^4	*s
*DO	Basic program options specifying transport and kinetics calculations to be performed. See Section 3.4 for review and discussion	All flags are false	None
ILMUL	Scale factor for illumination	1.0	None
ILUMO	Value of illumination in absence of shortwave radiation	0.0	*W/m^2
OVERWRITE	Overwrite previous output; enable with 1, disable with 0	0	None
*SCALE	Scale initial conditions or loads. Format for initial condition is: *SCALE <value> ICI <state> Format for loads is: *SCALE <value><load_type> <state> where <load_type> can be any of the parameters listed in Table 3.4 and <state> can be any of the ones listed in Table 3.3	1.0 for all states	None
SILIM	Handling of silica and nitrogen limitation in diatom growth and handling of nitrogen limitation in greens growth. 1 enables $f = f_{silica} \times \min(f_{nitrogen} \times f_{phosph})$ for diatoms and $f = \min(f_{nitrog} \times f_{phosph})$ for greens 2 enables $f = \min(f_{silica} \times f_{nitrog} \times f_{phosph})$ for diatoms and $f = \min(f_{nitrog} \times f_{phosph})$ for greens 3 enables $f = f_{silica} \times f_{phosph}$ for diatoms and $f = f_{phosph}$ for greens	1	None
TMEND	Day number of last hydrodynamic record	9999.9	Days
TMSTRT	Day number of first hydrodynamic record	0.0	Days
START_DAY_SCALE_LOAD_94	Starting day number for the scaling of 1994 loads	0.0	Days
END_DAY_SCALE_LOAD_94	Ending day number for the scaling of 1994 loads	9999.9	Days
START_DAY_SCALE_LOAD_95	Starting day number for the scaling of 1995 loads	0.0	Days
END_DAY_SCALE_LOAD_95	Ending day number for the scaling of 1995 loads	9999.9	Days

*s = seconds; W = watts; m = meters

Table 3.3 States for General Program Parameter *SCALE

State	Description
DOC	Dissolved organic carbon
DIA	Diatom carbon
GRE	Greens
ZOO	Zooplankton
LOC	Labile organic carbon
ROC	Refractory organic carbon
SRP	Soluble reactive phosphorus
DOP	Dissolved organic phosphorus
LOP	Labile organic phosphorus
ROP	Refractory organic phosphorus
NH_4	Ammonia
NO_3	Nitrate
DON	Dissolved organic nitrogen
LON	Labile organic nitrogen
RON	Refractory organic nitrogen
SA	Available silica
SU	Particulate biogenic silica
PCB_5+8	PCB congener 5+8
PCB_15+17	PCB congener 15+17
PCB_16+32	PCB congener 16+32
PCB_18	PCB congener 18
PCB_26	PCB congener 26
PCB_28+31	PCB congener 28+31
PCB_33	PCB congener 33
PCB_37+42	PCB congener 37+42
PCB_44	PCB congener 44
PCB_49	PCB congener 49
PCB_52	PCB congener 52
PCB_56+60	PCB congener 56+60
PCB_66	PCB congener 66
PCB_70+76	PCB congener 70+76
PCB_74	PCB congener 74
PCB_77+110	PCB congener 77+110
PCB_81	PCB congener 81
PCB_87	PCB congener 87
PCB_84+92	PCB congener 84+92
PCB_89	PCB congener 89
PCB_99	PCB congener 99
PCB_101	PCB congener 101
PCB_118	PCB congener 118
PCB_123+149	PCB congener 123+149
PCB_105+132+153	PCB congener 105+132+153
PCB_151	PCB congener 151
PCB_138+163	PCB congener 138+163
PCB_170+190	PCB congener 170+190
PCB_172+197	PCB congener 172+197
PCB_180	PCB congener 180
PCB_182+187	PCB congener 182+187
PCB_195+208	PCB congener 195+203
PCB_196+203	PCB congener 196+203
PCB_201	PCB congener 201
PCB_146	PCB congener 146
PCB_85	PCB congener 85
ATZ	Atrazine
CL	Chloride
CLAY	Clay

Table 3.4. Load Types for Program Parameter *SCALE

Load Type	Description
LOAD_MON	Monitored loads
LOAD_UNMON	Unmonitored loads
LOAD_ATM	Atmospheric loads
LOAD_MENOMINEE	Menominee River load
LOAD_FOX	Fox River load
LOAD_SHEBOYGAN	Sheboygan River load
LOAD_MILWAUKEE	Milwaukee River load
LOAD_GRAND_CALUMET	Grand Calumet River load
LOAD_ST_JOSEPH	St. Joseph River load
LOAD_KALAMAZOO	Kalamazoo River load
LOAD_GRAND	Grand River load
LOAD_MUSKEGON	Muskegon River load
LOAD_PERE_MARQUETTE	Pere Marquette River load
LOAD_MANISTIQUE	Manistique River load
LOAD_CEDAR	Cedar River load
LOAD_PESHTIGO	Peshtigo River load
LOAD_OCONTO	Oconto River load
LOAD_PENSAUKEE	Pensaukee River load
LOAD_DUCK	Duck River load
LOAD_KEWAUNEE	Kewaunee River load
LOAD_EAST_TWIN	East Twin River load
LOAD_WEST_TWIN	West Twin River load
LOAD_MANITOWOC	Manitowoc River load
LOAD_ROOT	Root River load
LOAD_BLACK_SH	Black (SH) River load
LOAD_BLACK_HD	Black (HD) River load
LOAD_PIGEON	Pigeon River load
LOAD_WHITE	White River load
LOAD_PENTWATER	Pentwater River load
LOAD_BIG_SABLE	Big Sable River load
LOAD_MANISTEE	Manistee River load
LOAD_BETSIE	Betsie River load
LOAD_BOARDMAN	Boardman River load
LOAD_JORDAN	Jordan River load
LOAD_STURGEON	Sturgeon River load
LOAD_WHITEFISH	Whitefish River load
LOAD_RAPID	Rapid River load
LOAD_ESCANABA	Escanaba River load
LOAD_FORD	Ford River load
LOAD_CHICAGO_RIVER	Outflow only: no loads
LOAD_STRAITS_MACKINAC	Outflow only: no loads
LOAD_WET_ATMOSPHERIC	Wet atmospheric load
LOAD_SED	Sediment load
LOAD_UNMONITORED_KALAMAZOO	Unmonitored Kalamazoo River load
LOAD_UNMONITORED_PERE_MARQUETTE	Unmonitored Pere Marquette River load
LOAD_MILECOQUINS	Milecoquins River load
LOAD_GALIEN	Galien River load
LOAD_SHORE_EROSION	Shoreline erosion load
LOAD_DRY_ATMOSPHERIC	Dry atmospheric load

Table 3.5. *DO Key General Program Parameters

Key Name	Description	Default Value
ATM_LOADS_ELIMINATION	Enable the elimination of phosphorus loads	FALSE
ATZ	Enable atrazine calculations	FALSE
CL	Enable chloride calculations	FALSE
CLAY	Enable clay calculations	FALSE
EUTRO	Enable eutrophication kinetics calculations	FALSE
EXP_DECAY_LOAD	Enable an exponential decay of phosphorus loads	FALSE
GRLIM	Write eutrophication growth limitation factors	FALSE
LOADS_INCREASE	Enable the increase of phosphorus loads	FALSE
MASSBALANCE	Enable mass balance calculations	FALSE
OCONNOR	Enable atrazine O'Connor volatilization	FALSE
PCB_5+18	Enable PCB 5+8 kinetic calculations	FALSE
PCB_15+17	Enable PCB 15+17 kinetic calculations	FALSE
PCB_16+32	Enable PCB 16+32 kinetic calculations	FALSE
PCB_18	Enable PCB 18 kinetic calculations	FALSE
PCB_26	Enable PCB 26 kinetic calculations	FALSE
PCB_28+31	Enable PCB 28+31 kinetic calculations	FALSE
PCB_33	Enable PCB 33 kinetic calculations	FALSE
PCB_37+42	Enable PCB 37+42 kinetic calculations	FALSE
PCB_44	Enable PCB 44 kinetic calculations	FALSE
PCB_49	Enable PCB 49 kinetic calculations	FALSE
PCB_52	Enable PCB 52 kinetic calculations	FALSE
PCB_56+60	Enable PCB 56+60 kinetic calculations	FALSE
PCB_66	Enable PCB 66 kinetic calculations	FALSE
PCB_70+76	Enable PCB 70+76 kinetic calculations	FALSE
PCB_74	Enable PCB 74 kinetic calculations	FALSE
PCB_77+110	Enable PCB 77+110 kinetic calculations	FALSE
PCB_81	Enable PCB 81 kinetic calculations	FALSE
PCB_87	Enable PCB 87 kinetic calculations	FALSE
PCB_84+92	Enable PCB 84+92 kinetic calculations	FALSE
PCB_89	Enable PCB 89 kinetic calculations	FALSE
PCB_99	Enable PCB 99 kinetic calculations	FALSE
PCB_101	Enable PCB 101 kinetic calculations	FALSE
PCB_118	Enable PCB 118 kinetic calculations	FALSE
PCB_123+149	Enable PCB 123+149 kinetic calculations	FALSE
PCB_105+132+153	Enable PCB 105+132+153 kinetic calculations	FALSE
PCB_151	Enable PCB 151 kinetic calculations	FALSE
PCB_138+163	Enable PCB 138+163 kinetic calculations	FALSE
PCB_170+190	Enable PCB 170+190 kinetic calculations	FALSE
PCB_172+197	Enable PCB 172+197 kinetic calculations	FALSE
PCB_180	Enable PCB 180 kinetic calculations	FALSE
PCB_182+187	Enable PCB 182+187 kinetic calculations	FALSE
PCB_195+208	Enable PCB 195+208 kinetic calculations	FALSE
PCB_196+203	Enable PCB 196+203 kinetic calculations	FALSE
PCB_201	Enable PCB 201 kinetic calculations	FALSE
PCB_146	Enable PCB 146 kinetic calculations	FALSE
PCB_85	Enable PCB 85 kinetic calculations	FALSE
RESCALE_PLOADS_ATM_TRIBS	Enable the rescaling of phosphorus loads	FALSE
SEDIMENT	Enable sediment transport calculations	FALSE
SED_LOADS_INCREASE	Enable the increase of sediment loads	FALSE
SED_MASS_BALANCE	Enable sediment bed mass balance calculations	FALSE
SHEAR	Enable bottom shear calculations	FALSE
TRIB_LOADS_ELIMINATION	Enable the elimination of tributary phosphorus loads	FALSE
TRANS	Enable water column transport	FALSE
VOLAT	Enable volatilization calculations	FALSE

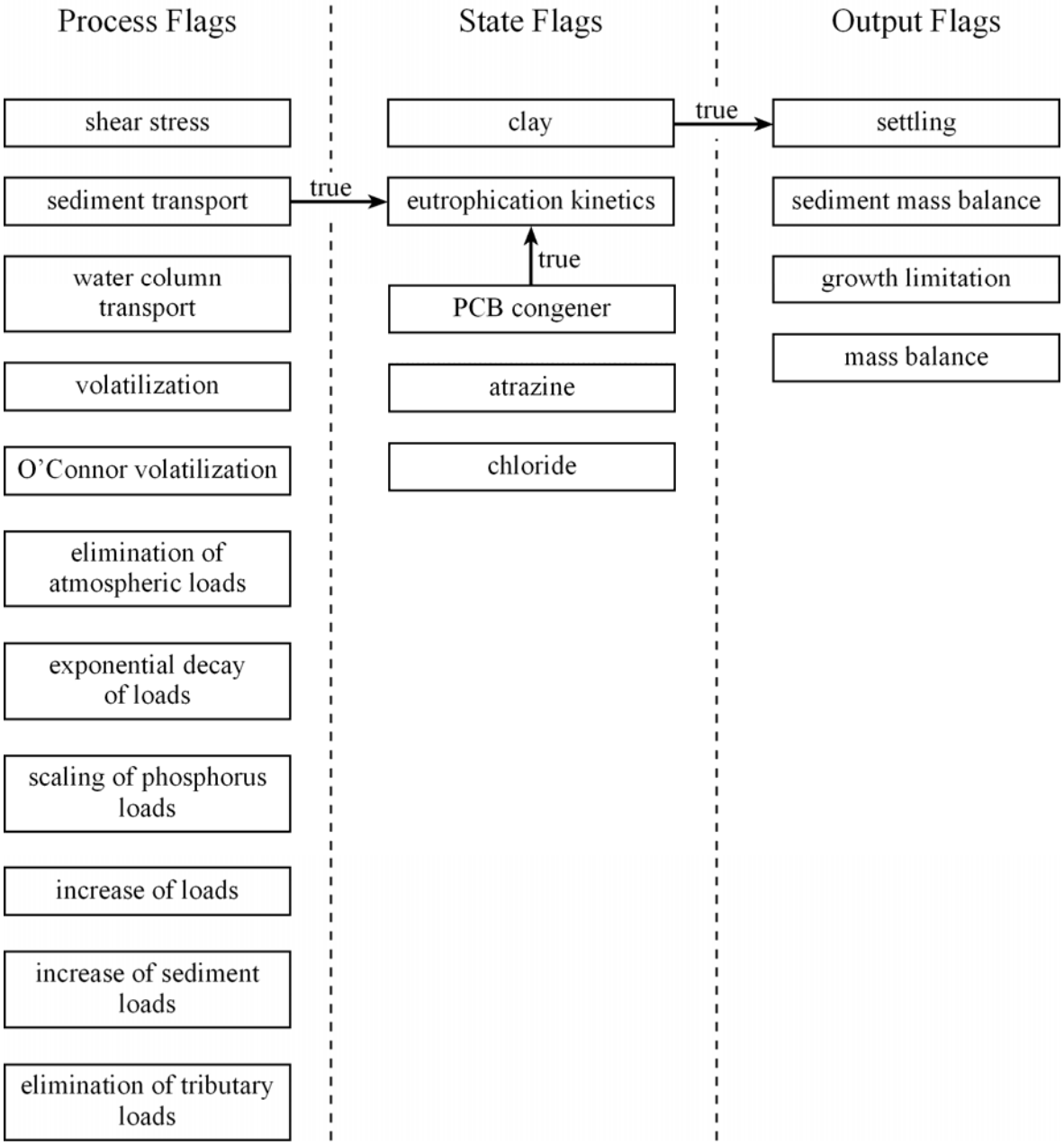


Figure 3.1. Logical associations among the LM3 program flags. All flags are FALSE by default and *DO FIREAD keys are used to enable the desired options.

Water column transport, atrazine calculation, and PCB calculations can be enabled independently of one another. However, enabling sediment transport will cause the program to automatically turn on eutrophication calculations. Enabling eutrophication calculations will cause the program to automatically perform settling calculations. Enabling PCB calculations will cause the program to turn on the eutrophication. There is a shear stress calculation that can be performed independently of the eutrophication, atrazine, and PCB calculations. For example, the following FIREAD input key line:

*DO EUTRO

will enable eutrophication and settling calculations. See Chapter 13 for examples of different simulations that show the use of the *DO key.

3.5 Eutrophication and Chemical-Specific Parameters

The input parameters used for eutrophication, atrazine, and PCB kinetic calculations are not discussed nor are they listed in this chapter. The discussion of kinetic calculations is complex and involves many equations. Thus separate chapters have been devoted to discuss kinetics for the cases of eutrophication, PCBs, and atrazine.

Eutrophication kinetics is discussed in detail in Chapter 6. A listing of all the parameters associated with the kinetics is shown in Table 6.1. See Section 13.4 for a sample input deck that demonstrates the use of eutrophication input parameters.

PCB kinetics is discussed in detail in Chapter 8. Listings of all the parameters associated with the kinetics are shown in Tables 8.1 through 8.4. See Section 13.5 for a sample input deck that demonstrates the use of PCB congener parameters.

Atrazine kinetics is discussed in detail in Chapter 9. A listing of all the parameters associated with the kinetics is shown in Table 9.1. See Section 13.2 for a sample input deck that demonstrates the use of atrazine parameters.

There is no chloride kinetics chapter because chloride does not undergo any kinetics that we know of. Thus there are no input parameters associated with chloride modeling. Nevertheless, see Section 13.3 for a chloride sample input deck.

3.6 General Transport Parameters

There are several input parameters that can affect the water column and sediment bed transport simulation of LM3. These parameters are used to perform various tasks such as scaling diffusion coefficients, specifying settling velocities, enabling boundary conditions, specifying the Crank-Nicholson parameter, enabling diagnostics, and so on. The full list of available water column transport parameters in LM3 is given in Table 3.6. The full list of available sediment bed transport parameters in LM3 is given in Table 3.7.

Table 3.6. Water Transport Parameters

Key Name	Description	Default Value	Units
BCC	Enable boundary conditions; 1 enables, 0 disables	0	None
DIAC	Enable diagnostics; 1 enables, 0 disables	0	None
DLTFTN	Water column transport limiting Courant number	0.9	None
FLC	Enable flows; 1 enables, 0 disables	1	None
KNIK	Nikuradse roughness parameter	0.11	m
SLC	Transport algorithm; 1 = QUICKEST, 2 = Upwind	1	None
TH	Implicit scheme θ-weighting value	0.5	None
VBC	Enable volume balance checking; 1 enables, 0 disables	1	None
VDIA	Diatom settling velocity	2.3×10^{-6}	$m \cdot s^{-1}$
VGRE	Green algae settling velocity	1.7×10^{-6}	$m \cdot s^{-1}$
VLOC	Labile detrital carbon settling velocity	1.7×10^{-6}	$m \cdot s^{-1}$
VLON	Labile organic nitrogen settling velocity	1.7×10^{-6}	$m \cdot s^{-1}$
VLOP	Labile organic phosphorus settling velocity	1.7×10^{-6}	$m \cdot s^{-1}$
VROC	Refractory detrital carbon settling velocity	1.7×10^{-6}	$m \cdot s^{-1}$
VRON	Refractory organic nitrogen settling velocity	1.7×10^{-6}	$m \cdot s^{-1}$
VROP	Refractory organic phosphorus settling velocity	1.7×10^{-6}	$m \cdot s^{-1}$
VSU	Unavailable silica settling velocity	1.7×10^{-6}	$m \cdot s^{-1}$
VCLAY	Clay settling velocity	1.0×10^{-5}	$m \cdot s^{-1}$
XYDFC	Enable horizontal dispersion; 1 enables, 0 disables	1	None
XYDFMAX	Maximum horizontal dispersion	$1.0 \times 10^{+29}$	$m^2 \cdot s^{-1}$
XYDFMUL	Horizontal dispersion scaling factor	1.0	None
ZDFC	Enable vertical dispersion; 1 enables, 0 disables	1	None
ZDFMAX	Maximum vertical dispersion	10^{29}	$m^2 \cdot s^{-1}$
ZDFMUL	Vertical dispersion scale factor	1	None

Table 3.7. Sediment Transport Parameters

Key Name	Description	Default Value	Units
DENSITY_LOC	Labile organic carbon particle density	1200.00	$kg \cdot m^{-3}$
DENSITY_ROC	Refractory organic carbon particle density	1200.00	$kg \cdot m^{-3}$
DENSITY_SRP	Soluble reactive phosphorus particle density	1200.00	$kg \cdot m^{-3}$
DENSITY_LOP	Labile organic phosphorus particle density	1200.00	$kg \cdot m^{-3}$
DENSITY_ROP	Refractory organic phosphorus particle density	1200.00	$kg \cdot m^{-3}$
DENSITY_LON	Labile organic nitrogen particle density	1200.00	$kg \cdot m^{-3}$
DENSITY_RON	Refractory organic nitrogen particle density	1200.00	$kg \cdot m^{-3}$
DENSITY_CLAY	Clay particle density	2200.00	$kg \cdot m^{-3}$
MAX_RESP_VEL	Maximum resuspension velocity allowed	1.0×10^{-4}	$m \cdot s^{-1}$
QMUL_SED	Sediment flow scaling factor	1.0	None
SLC_SED	Transport algorithm 1 = QUICKEST 2 = UPWIND	1	None
TH_SED	Implicit scheme θ-weighting value	0.55	None
TIME_STEP_SED_SCALE	Sediment bed time step scaling factor	1.0	None
ZDF	Sediment vertical diffusion	1.0×10^{-8}	$m^2 \cdot s^{-1}$
ZDFMAX_SED	Sediment maximum vertical dispersion	1.0×10^{4}	$m^2 \cdot s^{-1}$
ZDFMUL_SED	Sediment vertical dispersion scale factor	1.0	None

All the transport parameters shown in Tables 3.6 and 3.7 have default values and some of them have associated units while others do not. The user does not have to specify these parameters in the input deck if the intention is to rely on the default values. However, if the user wants to use non-default values for some of the parameters, then one needs to set the desired values in the input deck. See Chapter 13 for examples of different simulations that show the use of these transport parameters.

CHAPTER 4

LM3 NEW FEATURES

LM3 has undergone many changes since its origins as an In-place Pollutant eXport (IPX) application. New state variables and algorithms were added to expand the modeling capabilities of the application. Some of the new states added were atrazine, chloride, polychlorinated biphenyls (PCBs), *Diporeia*, zebra mussels, *Mysis relicta, Bythotrephes longimanus*, and blue-green algae, among others.

These new features were added in stages over a long period of time. Each stage of development is represented by a different Revision Control System (RCS) branch number as shown in the next section. Each branch can also be seen as a different kind of LM3 model because some of the branches contain features that are not necessarily found in some of the others. This allows for the development of parallel sub-models in LM3.

In the discussion that follows, the main new LM3 features are outlined under their corresponding RCS development branches. The goal is to highlight some of the unique features that were incorporated into LM3. Some of these branches, especially the later ones, are still under testing and/or development. Thus the new features discussed in this chapter are the ones contained in LM3 as of this writing.

4.1 New Features

The following new features were included in Version 3.2.8 of the LM3 calculation.

1. Eutrophication loads were implemented.

2. A shear stress algorithm was added. The algorithm was taken from the application known as SEDZL.

3. A sediment bed transport algorithm was added. The algorithm is the same one used in SEDZL and is known as the age-layer algorithm. However, this algorithm has never been tested within LM3.

4. Atrazine and sand state variables were added.

The following new features were included in Version 3.2.9 of the LM3 calculation.

1. Source code modifications were made to allow the program to read in a new hydrodynamics input format. The new input format has the hydrodynamics data stored in separate Common Data Format (CDF) files for each sigma layer.

2. New source code was added to generate phytoplankton carbon production for the level 2 toxic model (also known as LM2).

The following new features were included in Version 3.2.10 of the LM3 calculation.

1. New Waterways Experiment Station (WES) modifications to hydrodynamics were made to accommodate tributary flows. This is the first version of LM3 that uses hydrodynamics with tributary flows.

2. Vertical transport was improved using ULTIMATE QUICKEST (UQ) for the explicit term of Crank-Nicholson. Settling calculation was added to the transport calculation.

The following new features were included in Version 3.2.11 of the LM3 calculation.

1. Tributary flows were added to the hydrodynamic fields.

2. The hydrodynamics format was modified to accommodate larger input data sets with shorter update intervals.

3. A state variable for chloride was added.

The following new features were included in Version 3.2.12 of the LM3 calculation.

1. The program was modified to enable long-term simulations, that is, simulations that span time periods of more than two years. Long-term simulations are made possible by recycling a subset of the 1994-1995 hydrodynamic and load data. The hydrodynamic and load data that are recycled consist of the last 10 days of 1994 and all of 1995 giving the user one full calendar year worth of values. Most of the modifications that support these long-term simulations were made in the routine hydro.F.

2. Shoreline erosion (bluff) loads for particulate organic carbon (POC) were added. These POC loads are believed to consist mostly of refractory organic carbon (ROC); thus ROC is the only carbon state that receives bluff loads in the program.

3. The light limitation factor was modified to account for shorter update intervals.

The following new features were included in Version 3.2.13 of the LM3 calculation.

1. New Ray Chapman modifications to the UQ transport routine were incorporated (see Appendix A). This is the same transport routine that Version 3.2.10 used. Settling is part of the transport calculation.

2. A number of control flags were added in support of various eutrophication load management scenarios. These flags are used to trigger an increase or a decrease of nutrient loads starting for a specified year when running the program for 10 or more years.

3. More than two-year simulations were implemented by looping over the 1994-1995 hydrodynamic and load data as many times as required. The hydrodynamic output, for the 1994-1995 calibration period, covers the period from January 1, 1994 to December 21, 1995. If simulations extend beyond December 21, 1995, the program uses the hydrodynamic data of the last 10 days of 1994 for the missing last 10 days of 1995 in order to get a full 1995 calendar year. To allow simulations to extend beyond 1995, the program re-reads the hydrodynamic output starting on January 1, 1994 onward through December 21, 1995. To complete the calendar date of 1995, the data corresponding to the last 10 days of 1994 are used as surrogate values for the last 10 missing days of 1995 once again. This entire process is repeated as many times as needed to cover the desired number of simulation years. This new implementation replaced the one-year loop of Version 3.2.12.

The following new feature was included in Version 3.2.14 of the LM3 calculation.

Conditional statements were added to bypass the recycling of soluble reactive phosphorus (SRP) and labile organic phosphorus (LOP) load values for the Fox and Grand Rivers in one case and for the Betsie and Grand Rivers in the other case. SRP and LOP load values were generated for a 20-year time period.

The following new features were included in Version 3.2.15 of the LM3 calculation.

1. A volatilization calculation and a lumped-sum decay factor were added for atrazine.

2. PCB kinetics, including volatilization, was added to the program. PCBs are modeled as seven phases in the water column and three phases in the sediment column. The phases represent the partitioning of PCBs with respect to the eutrophication carbon states in addition to the PCB congener purely dissolved phase. A sediment-water column diffusion process for PCBs was also added.

3. A sediment bed transport algorithm was added in support of PCB simulations (see Appendix A). The algorithm uses a simple UQ calculation.

4. Clay was added as a state variable, while sand was removed from the list of states modeled.

5. Multi-year simulations were implemented by looping over the 1994-1995 hydrodynamic and load data as many times as required. The hydrodynamic data is looped over in the same way as explained in feature three of Version 3.2.13 above.

The following new features were included in Version 3.2.16 of the LM3 calculation.

1. A new enhanced eutrophication was implemented by adding three additional zooplankton groups to the original one. The original zooplankton was herbivorous zooplankton; the

three additional groups added were: *Mysis relicta, Bythotrephes longimanus,* and carnivorous zooplankton.

2. A sediment compartment was implemented in which nutrient sediment concentrations were stratified into zones according to a correlation between water column depth and observed sediment nutrient concentrations. Sediment concentrations were populated for each 5 km cell using the inverse square method of interpolation within each zone. A sediment resuspension flux was calculated using a user-defined resuspension velocity together with the interpolated sediment concentrations. There is no limit on the amount of mass that can be resuspended for a given nutrient; in other words, the sediment bed is treated as an infinite reservoir of nutrients.

The following new features were included in Version 3.2.17 of the LM3 calculation.

1. The eutrophication model was further enhanced by dividing the green algae into four groups. The four groups are: greens, nitrogen-fixing blue-greens, non-nitrogen-fixing blue-greens, and other green algae.

2. Two new species were added, zebra mussels and *Diporeia*, in support of an ecosystem model.

3. This branch is still being tested as of this writing.

The following new feature was included in Version 3.2.18 of the LM3 calculation.

Threshold concentration input parameters were added to the source code to address the problem of the occurrence of negative concentrations in the eutrophication kinetics. The new input parameters apply only to dissolved silica (SA), SRP, dissolved ammonia (NH_4), and nitrate (NO_3).

The following new features were included in Version 3.2.19 of the LM3 calculation.

1. This branch, an enhanced version of 3.2.15, includes modifications to support a eutrophication hindcast. Recycling of monitored phosphorus loads and phosphorus sediment loads was eliminated to allow the use of historical phosphorus loads from the 1970s. New logical flags were added within the program to aid in deciding which eutrophication states and which loads were not going to be recycled during the hindcast. These flags were added within the subroutine hydro.F. Logical flags were added within the subroutine loads.F to help in deciding which load locations and states did not need any time adjustment during the hindcast run. The hindcast periods considered were 1960-2011 and 1976-2011. Because there was no hydrodynamic data available for the entire hindcast periods, the 1994-1995 hydrodynamics was looped over in the same way as explained in feature three of Version 3.2.13 (see above).

2. Several new input CDF files were created in support of the hindcast runs.

The following new feature was included in Version 3.2.20 of the LM3 calculation.

> This branch is an extension of Version 3.2.16, with some source code modifications made to allow for long-term forecasts. Only a single year of the 1994-1995 hydrodynamic data is used for projections. These projections are program runs that extend beyond the 1994-1995 study period. This branch is still being tested as of this writing.

CHAPTER 5

WATER COLUMN TRANSPORT

Considerable attention has been paid to correctly simulating water column transport in the LM3 simulation (Chapman *et al*., 1997). The correct implementation of hydrodynamic flow and dispersion information (Schwab and Beletsky, 1998) as well as the simulation of concentration gradients have been identified as key elements of water quality analysis for the Lake Michigan system and have been given particular emphasis throughout the development of the LM3 model.

Numerical diffusion has been reduced by implementing an explicit, third-order accurate Non-Uniform-Grid ULTIMATE QUICKEST (UQ) transport scheme in the horizontal and vertical dimensions, and a second-order semi-implicit Crank-Nicholson transport scheme in the vertical dimension.

UQ uses an additional spatial term to estimate concentrations used in computing horizontal and vertical mass fluxes. A non-uniform grid UQ scheme was developed using a three-point Lagrangian interpolation function to estimate constituent values at grid cell interfaces. Specifically, advective multipliers for each of three upstream weighted grid cells are derived in terms of cell lengths and the local cell interface velocity. Time invariant parts of the interpolation functions are calculated once thus minimizing computations for additional constituents.

The vertical transport algorithm incorporates settling velocities and, as stated above, employs the semi-implicit method of Crank and Nicholson. An excellent discussion of this technique is included in Carnahan *et al*. (1969). The implicit solution algorithm makes use of a Thomas tri-diagonal matrix solver, which is also discussed in Carnahan *et al*. (1969).

The vertical transport scheme can be made fully explicit by setting a parameter known as "THETA" equal to zero or fully implicit by setting the same parameter equal to one. If "THETA" is set to a value between zero and one, the vertical transport is time-weighted between the explicit (using UQ) and implicit results and in this case we have a semi-implicit calculation. It is recommended to use a value of 0.55 for "THETA" when running LM3 simulations.

The details of the transport algorithm employed in this work are complex. A derivation of the UQ transport algorithm in one dimension and its implementation in the LM3 simulation is detailed in Appendix E. Although the UQ results are derived in one dimension, the transport calculations in LM3 account for all three dimensions. The one-dimensional results apply to each dimension because transport is performed as a one-dimensional exchange between adjacent cells through an individual cell face, irrespective of the dimensionality of the model. An outline

of the vertical transport derivation is given in Appendix A. A brief summary of the UQ and vertical transport main results is given below.

To aid in visualizing the derivation of the UQ transport algorithm in Appendix E, a brief discussion of the model grid is presented in the following section. This discussion also serves as a reference for the subsequent sections.

5.1 Model Grid

The Lake Michigan Mass Balance Project (LMMBP) model grid is three-dimensional. The horizontal segmentation of the grid consists of 2,318 cells with a scale of 5 km. Thus the horizontal area of each cell is 5 km x 5 km (Figure 5.1). The vertical segmentation consists of 20 sigma levels of the depth at each respective grid point. The number of sigma layers is 19; that is, one less than the number of sigma levels.

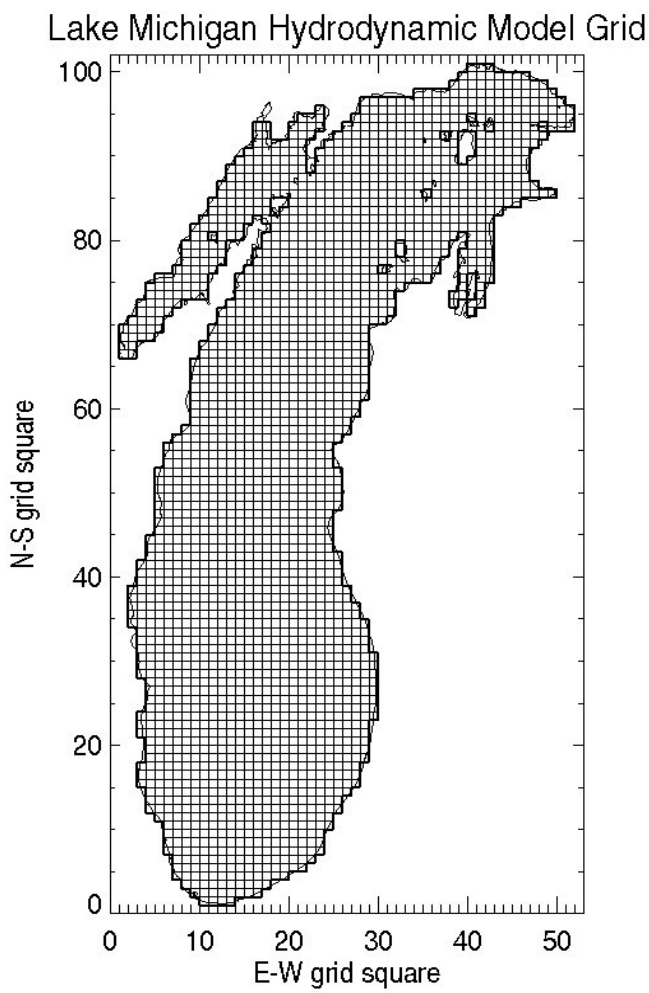


Figure 5.1. The LMMBP high-resolution grid. Grid scale is 5 km.

The sigma levels are not equally spaced; they are closely packed at grid points with shallow depths and more spaced out at grid points with larger depths. Each sigma layer varies in thickness depending on the water column depth of each grid point. In nearshore areas, where water depth can be as low as 2.5 meters, some of the sigma layers can have thicknesses under 0.1 meters. In offshore areas, where water depth can exceed 250 meters, some of the sigma layers can have thicknesses exceeding 30 meters.

The number of cells for each sigma layer is 2,318. Thus the total number of grid cells is 2,318 x 19 = 44,042.

5.2 The ULTIMATE QUICKEST Transport Scheme

The UQ transport scheme developed by Leonard (1979) was augmented for use with variable grid sizes by Chapman (Chapman *et al.*, 1997). The resulting transport algorithm was coded in FORTRAN and applied to the Chesapeake Bay (ICM) model (Cerco and Cole, 1994; 1995). The same transport scheme was adapted for use in the LMMBP and the In-place Pollutant eXport (IPX) pollutant calculation (Velleux *et al*., 2001; Chapman *et al.*, 1997).

The box model calculation performs numerical integration of spatially varying concentrations using quadratic interpolations of the concentration to infer its value at flow faces and analytic integration over the space- and time-variables to account for changes in the concentration at the cell wall during the course of the time step.

The discussion that follows is only a summary of the UQ main results. See Appendix E for a detailed derivation of the results given below.

The general constituent transport equation that is solved is:

$$\frac{\partial \phi}{\partial t} + \frac{\partial u\phi}{\partial x} + \frac{\partial w\phi}{\partial z} - \frac{\partial}{\partial x}\left(\Gamma_x \frac{\partial \phi}{\partial x}\right) - \frac{\partial}{\partial z}\left(\Gamma_z \frac{\partial \phi}{\partial z}\right) = 0 \tag{5.1}$$

ϕ = constituent concentration
u = horizontal cell face velocity
w = vertical cell face velocity
Γ_x = horizontal cell face diffusion coefficient
Γ_z = vertical cell face diffusion coefficient
x = horizontal dimension
z = vertical dimension
t = time

For the case of one-dimensional horizontal transport, Equation 5.1 reduces to:

$$\frac{\partial \phi}{\partial t} + \frac{\partial u\phi}{\partial x} - \frac{\partial}{\partial x}\left(\Gamma_x \frac{\partial \phi}{\partial x}\right) = 0 \tag{5.2}$$

Integration of 5.2 over the time step Δt gives the result:

$$\phi^{n+1} = \phi^{n} - c\left(\bar{\phi}_r - \bar{\phi}_l\right) + \Delta x \bullet \alpha\left(GRAD_r - GRAD_l\right) \tag{5.3}$$

$\bar{\phi}_r$ = time-averaged concentration at right cell wall

$\bar{\phi}_l$ = time-averaged concentration at left cell wall

$$c = \frac{u\Delta t}{\Delta x}\left(\text{See Equation E.13}\right)$$

$$\alpha = \Gamma_x \frac{\Delta t}{\Delta x^2}$$

$GRAD_r$ is given by Equation E.17.
$GRAD_l$ is given by Equation E.18.

The solution for ϕ^{n+1} is found by calculating the time-averaged concentrations and gradients or, equivalently, the diffusive fluxes at the cell faces in (5.3) and integrating them over one time step. The transactional symmetry of the cells (the left face of cell i is the same as the right cell face of cell i-1) is exploited so that time-averaged concentrations and diffusive fluxes only need to be calculated at either the left or the right face of each cell.

The UQ corrected concentration at the right cell face, $\bar{\phi}_r$, can be expressed as a weighted sum of the concentration being interpolated. There are two ways to express $\bar{\phi}_r$ depending on the sign of the velocity u. For the case of u > 0 we have:

$$\bar{\phi}_r = T^{+}_{i-1}\phi^{n}_{i-1} + T^{+}_{i}\phi^{n}_{i} + T^{+}_{i+1}\phi^{n}_{i+1} \tag{5.4}$$

For the case of u < 0 we have:

$$\bar{\phi}_r = T^{-}_{i}\phi^{n}_{i} + T^{-}_{i+1}\phi^{n}_{i+1} + T^{-}_{i+2}\phi^{n}_{i+2} \tag{5.5}$$

The factors T^{+}_{i-1}, T^{+}_{i}, T^{+}_{i+1}, T^{-}_{i}, T^{-}_{i+1}, and T^{-}_{i+2} are advective multipliers which weigh the contribution of three adjacent grid point concentrations; they are given by Equations E.38, E.37, E.39, E.48, E.49, and E.50 of Appendix E, respectively. They are obtained by collecting terms associated with each constituent defined by the UQ advection operator.

The contribution of dispersion to the transport is solved in a similar fashion as for advection. The dispersive contribution to the change of mass within a cell can be expressed in terms of left and right dispersive fluxes. From Appendix E we have that:

$$\left(m^{n+1} - m^{n}\right)_{Dispersive} = \Delta t\left(G_r - G_l\right) \tag{5.6}$$

where

$$G_r \equiv A_r \Gamma_r \partial_x \phi_r^n \tag{5.7}$$

$$G_l \equiv A_l \Gamma_l \partial_x \phi_l^n \tag{5.8}$$

Equations 5.6, 5.7, and 5.8 are Equations E.63, E.64, and E.65, respectively. Because of the transactional symmetry of the cells we only need to calculate either the left or right face of each cell. In analogy to Equations 5.4 and 5.5 G_r can be written as a linear combination of concentrations. For the case of u > 0 we have that:

$$G_r = G_{i-1}^{+}\phi_{i-1}^n + G_i^{+}\phi_i^n + G_{i+1}^{+}\phi_{i+1}^n \tag{5.9}$$

For the case of u < 0 we have:

$$G_r = G_i^{-}\phi_i + G_{i+1}^{-}\phi_{i+1} + G_{i+2}^{-}\phi_{i+2} \tag{5.10}$$

The factors $G_{i-1}^{+}, G_i^{+}, G_{i+1}^{+}, G_i^{-}, G_{i+1}^{-}$, and G_{i+2}^{-} are the dispersive multipliers; they are given by Equations E.71, E.72, E.73, E.79, E.80, and E.81, respectively.

In the horizontal, the advection and dispersion contributions to the mass derivative for each cell can be calculated as:

$$\partial_t m = Q_l \bar{\phi}_l - Q_r \bar{\phi}_r + G_r - G_l \tag{5.11}$$

where Q_l and Q_r are the flows through the left and right faces of the cell.

Equation 5.11 above is Equation E.85 in Appendix E.

5.3 Vertical Implicit Transport

An outline of the derivation of vertical transport is presented in Appendix A. The discussion that follows is only a summary of the main results.

In the vertical, advective and diffusive transport is also present. The constituent transport equation can be written (see Equation 5.1):

$$\frac{\partial \phi}{\partial t} + \frac{\partial w\phi}{\partial z} - \frac{\partial}{\partial z}\left(\Gamma_z \frac{\partial \phi}{\partial z}\right) = \text{RHS} \tag{5.12}$$

where RHS represents all horizontal transport, and all sources/sinks. The terms w and Γ_z are the vertical velocity and vertical dispersion, respectively. Equation 5.12 is Equation A.15 in Appendix A. If incorporating a velocity due to gravitational settling, replace the term w by w - v_{set} in Equation 5.12. Integrating Equation 5.12 vertically and over time, and utilizing theta-weighting for advection yields:

$$H\phi^{n+1} + \theta\Delta t\delta_z\left(w\phi^{n+1}\right) - \Delta t\delta_z\left(\Gamma_z \frac{\partial\phi^{n+1}}{\partial z}\right) = H\phi^* \tag{5.13}$$

where ϕ^* denotes all n-time transport and H is the cell height. Equation 5.13 is Equation A.16 in Appendix A. Equation 5.13 can be recast as follows:

$$A_{i+1}\phi_{i+1}^{n+1} + A_i\phi_i^{n+1} + A_{i-1}\phi_{i-1}^{n+1} = \phi^* \tag{5.14}$$

where the coefficients A_{i+1}, A_i, and A_{i-1} are given by Equations A.18, A.19, and A.20 of Appendix A, respectively. The explicit portion of the algorithm ϕ^* is given by Equation A.21. These coefficients are computed once, stored in arrays, and used to update each constituent. This is accomplished by loading the theta-weighted explicit part of the algorithm ϕ^* for each successive constituent and inverting the resulting matrix *via* a Thomas tri-diagonal solver (Carnahan *et al*., 1969).

CHAPTER 6

EUTROPHICATION KINETICS

Eutrophication refers to the process by which an aquatic system becomes enriched with dissolved nutrients, resulting in increased growth of algae and other microscopic plants. This process occurs naturally over many years but can be accelerated by human activities such as fertilizer runoff from farms and sewage input. The nutrients consist mostly of nitrates and phosphates.

Understanding eutrophication is critical in water quality studies. For instance, too many nutrients entering an aquatic system can cause large algal blooms followed by natural die-off and decay which results in low oxygen levels. Low oxygen levels can be detrimental to fish, particularly when levels become hypoxic or even worse, anoxic.

Another critical aspect of eutrophication is that it affects the recycling of toxic chemicals that bind to organic particles in aquatic systems. Polychlorinated biphenyls (PCBs) are examples of such toxic chemicals. Because they are found in Lake Michigan and have adverse effects on fish, they were the primary focus in the Lake Michigan Mass Balance Project (LMMBP).

Eutrophication models are used to address environmental factors such as the aforementioned ones. Thus this chapter presents a detailed description of the eutrophication model used for the LMMBP. The kinetics equations are based on the Corps of Engineers Water Quality Integrated Compartment Model (CE-QUAL-ICM) (Cerco and Cole, 1995). The mathematical equations are constructed on the basis of the principle of conservation of mass. This is an elementary physical law which is satisfied by macroscopic natural systems.

The water quality constituents of primary interest of this model are nutrients and phytoplankton. Phytoplankton growth is modeled as a function of available nutrients, temperature, and light; other processes such as phytoplankton mortality and herbivorous zooplankton grazing are included. There are 17 constituents or state variables that are described by this model. Dissolved oxygen has not been incorporated into this model as of this writing.

6.1 Development of Eutrophication Equations

A characteristic of eutrophication modeling is the many interactions between nutrients, plankton, and sediments and the transformation reactions describing the conversions between dissolved and particulate phases. In a modeling framework, each interaction is described as a

mathematical equation, and the challenge is to define a relatively simple expression to approximate more complex biochemical processes.

6.1.1 Phytoplankton Growth

Although several phytoplankton groups are present in Lake Michigan, the lake is dominated mainly by diatoms and flagellates (Makarewicz *et al*., 1998). For the discussion that follows, algae that are not considered diatoms (non-diatoms) are denominated as green algae; thus flagellates are grouped under green algae. The major difference between these classes are silica dependence by the diatoms and differences in settling rates, carbon content, and growth rates at different times of the year. It has been speculated that diatoms grow faster than "non-diatoms" and that they grow better at colder temperatures because one usually observes diatom blooms during the spring in the Great Lakes (Schelske and Stoermer, 1971, 1972; Rousar, 1973; Brooks and Torke, 1977; Bartone and Schelske, 1982; Schelske *et al*., 1986; Fahnenstiel and Scavia, 1987a,b; Scavia and Fahnenstiel, 1987; Rockwell *et al*., 1989; Stoermer, 1993). The kinetic equations used in this model are based on the WASP family of models (Thomann and Di Toro, 1975; Di Toro and Connolly, 1980; Rodgers and Salisbury, 1981a,b) and CE-QUAL-ICM, developed by the United States Army Corps of Engineers (USACE) (Cerco and Cole, 1993). However, note that unlike those found in the WASP family of models, the kinetic equations presented in this chapter do not display a settling rate term. The settling rates are included within the transport algorithm of LM3. Thus settling is accounted for as a transport and not a kinetic process, as it should be. The basic phytoplankton growth equation can be written as:

NET PRODUCTION = GROSS PRODUCTION – MORTALITY (6.1)

Equation 6.1 can be represented in mathematical terms by:

$$\frac{dPHYTO}{dt} = \left(k_g - k_d\right) \bullet PHYTO - k_{gz} \bullet Z \tag{6.2}$$

where:

PHYTO = phytoplankton concentration ($kg \cdot m^{-3}$)
t = time (s)
k_g = phytoplankton growth rate (s^{-1})
k_d = phytoplankton mortality rate (s^{-1})
k_{gz} = predation rate (s^{-1})
Z = zooplankton concentration ($kg \cdot m^{-3}$)

The growth rate can be written as:

$$k_g = k_{gmax} \bullet f(N) \bullet f(T) \bullet f(I) \tag{6.3}$$

where:

k_{gmax} = optimum growth rate (s^{-1})
f(N) = nutrient growth dependency

f(I) = light growth dependency
f(T) = temperature growth dependency

The optimum growth rate, k_{gmax}, has different values for different types of phytoplankton. Thus diatoms and non-diatoms (greens) have slightly different optimum growth rates. This parameter is represented in the LM3 source code with two variable names: PMD, for diatoms, and PMG, for greens. Table 6.1 shows default values for the aforementioned variables.

6.1.1.1 Nutrient Growth Dependency

For the nutrient growth dependency, f(N), we use the standard Monod equation (Monod, 1949) and variations of it depending whether nitrogen is considered a growth limiting factor or not. Diatoms are treated slightly different from non-diatoms because of the former's silica dependence in addition to nitrogen and phosphorus.

For the non-diatoms, the Liebig's law of minimum (Odum, 1971) applies with no silica dependency and assumes nitrogen is a growth limiting factor.

$$f(N) = \min\left[\left(\frac{NH_4 + NO_3}{k_{sat-N} + NH_4 + NO_3}\right), \left(\frac{P_{av}}{k_{sat-P} + P_{av}}\right)\right] \tag{6.4}$$

where:

min = minimum of the two expressions (separated by a comma) within the square brackets.
k_{sat-N} = half-saturation coefficient for nitrogen uptake (kg · m^{-3}) = KHNG in Table 6.1
k_{sat-P} = half-saturation coefficient for phosphorus uptake (kg · m^{-3}) = KHPG in Table 6.1
P_{av} = available phosphorus (kg · m^{-3}), given by Equation 6.6 (see below)
NH_4 = ammonia concentration (kg · m^{-3})
NO_3 = nitrate concentration (kg · m^{-3})

If nitrogen is assumed not to be a limiting factor, then Equation 6.4 simplifies to:

$$f(N) = \frac{P_{av}}{k_{sat-P} + P_{av}} \tag{6.5}$$

Because we assume that a fraction of the dissolved organic phosphorus (DOP) is readily available for algal uptake, we define the term available phosphorus, P_{av}, as follows:

$$P_{av} = SRP + f_{DOP} \bullet DOP \tag{6.6}$$

where:

f_{DOP} = fraction of available DOP = AVFRAC in Table 6.1
DOP = dissolved organic phosphorus concentration (kg · m^{-3})
SRP = soluble reactive phosphorus concentration (kg · m^{-3})

Table 6.1. Eutrophication Parameters

Key Name	Description	Default Value	Units
ANCP	N:C ratio for plankton	2.0×10^{-1}	None
APCP	P:C ratio for plankton	1.0×10^{-2}	None
ASCD	Si:C ratio for diatoms	8.0×10^{-1}	None
AVFRAC	Available fraction of DOP	0.0	None
BMRD	Diatom base metabolic rate	8.6×10^{-7}	s^{-1}
BMRG	Green base metabolic rate	8.6×10^{-7}	s^{-1}
CCHLD	Carbon:chlorophyll ratio for diatoms	5.0×10^{1}	None
CCHLG	Carbon:chlorophyll ratio for green algae	5.0×10^{1}	None
CGZ	Zooplankton maximum growth rate	4.6×10^{-6}	s^{-1}
FCDD	Fraction of basal metabolism exuded as DOC by diatoms	1.0×10^{-1}	None
FCDG	Fraction of basal metabolism exuded as DOC by greens	1.0×10^{-1}	None
FCDP	Fraction of DOC produced by predation	1.0×10^{-1}	None
FCDZ	Fraction of DOC from zooplankton mortality	1.0×10^{-1}	None
FCLD	Fraction of labile POC produced by diatoms metabolism	1.0×10^{-1}	None
FCLG	Fraction of labile POC produced by greens metabolism	1.0×10^{-1}	None
FCLP	Fraction of labile PDC from predation	5.5×10^{-1}	None
FCLZ	Fraction of labile PDC from zooplankton mortality	5.5×10^{-1}	None
FCRD	Fraction of refractory POC produced by diatoms metabolism	1.0×10^{-1}	None
FCRG	Fraction of refractory POC produced by greens metabolism	1.0×10^{-1}	None
FCRP	Fraction of refractory PDC from predation	3.5×10^{-1}	None
FCRZ	Fraction of refractory PDC from zooplankton mortality	3.5×10^{-1}	None
FNDD	Fraction of DON produced by diatom metabolism	8.0×10^{-1}	None
FNDG	Fraction of DON produced by green algae metabolism	8.0×10^{-1}	None
FNDP	Fraction of DON produced by predation	1.0×10^{-1}	None
FNDZ	Fraction of DON produced by zooplankton mortality	4.0×10^{-1}	None
FNID	Fraction of inorganic nitrogen produced by diatom metabolism	1.0×10^{-1}	None
FNIG	Fraction of Inorganic nitrogen produced by green algae metabolism	1.0×10^{-1}	None
FNIP	Fraction of inorganic nitrogen produced by predation	0.0	None
FNIZ	Fraction of inorganic nitrogen produced by zooplankton mortality	1.0×10^{-1}	None
FNLD	Fraction of labile particulate nitrogen produced by diatom metabolism	1.0×10^{-1}	None
FNLG	Fraction of labile particulate nitrogen produced by green algae metabolism	1.0×10^{-1}	None
FNLP	Fraction of labile particulate nitrogen produced by predation	5.5×10^{-1}	None
FNLZ	Fraction of LON produced by zooplankton mortality	3.5×10^{-1}	None
FNRD	Fraction of refractory particulate nitrogen produced by diatom metabolism	0.0	None
FNRG	Fraction of refractory particulate nitrogen produced by green algae metabolism	0.0	None
FNRP	Fraction of RON produced by predation	3.5×10^{-1}	None
FNRZ	Fraction of RON produced by zooplankton mortality	1.5×10^{-1}	None
FPDD	Fraction of DOP produced by diatom metabolism	8.0×10^{-1}	None
FPDG	Fraction of DOP produced by green algae metabolism	8.0×10^{-1}	None
FPDP	Fraction of DOP produced by predation	1.0×10^{-1}	None
FPDZ	Fraction of DOP produced by zooplankton mortality	4.0×10^{-1}	None
FPID	Fraction of inorganic phosphorus produced by diatom metabolism	1.0×10^{-1}	None
FPIG	Fraction of inorganic phosphorus produced by green algae metabolism	1.0×10^{-1}	None
FPIP	Fraction of inorganic phosphorus produced by predation	0.0	None
FPIZ	Fraction of inorganic phosphorus produced by zooplankton mortality	1.0×10^{-1}	None
FPLD	Fraction of LOP produced by diatom metabolism	1.0×10^{-1}	None
FPLG	Fraction of LOP produced by green algae metabolism	1.0×10^{-1}	None
FPLP	Fraction of LOP produced by predation	5.5×10^{-1}	None
FPLZ	Fraction of LOP produced by zooplankton mortality	3.5×10^{-1}	None
FPRD	Fraction of ROP produced by diatom metabolism	0.0	None
FPRG	Fraction of ROP produced by green algae metabolism	0.0	None
FPRP	Fraction of ROP produced by predation	3.5×10^{-1}	None

Table 6.1. Eutrophication Parameters (Continued)

Key Name	Description	Default Value	Units
FPRZ	Fraction of ROP produced by zooplankton mortality	1.5×10^{-1}	None
FSAP	Fraction of silica made available through predation	0.0	None
GREFF	Zooplankton grazing efficiency	6.0×10^{-1}	None
ILUM0	Constant value of illumination (this is a user-defined value)	0.0	$W\text{-}m^{-2}$
ISMIN	Minimum optical illumination factor	6.05×10^{-1}	$W\text{-}m^{-2}$
KDC	Minimum DOC mineralization rate	1.2×10^{-7}	s^{-1}
KDCALG	DOC mineralization rate algal dependence	0.0	$m^3 \cdot kg^{-1} \cdot s^{-1}$
KDN	Minimum DON mineralization rate	2.3×10^{-7}	s^{-1}
KDNALG	DON mineralization rate algal dependence	0.0	$m^3 \cdot kg^{-1} \cdot s^{-1}$
KDP	Minimum DOP mineralization rate	5.0×10^{-7}	s^{-1}
KDPALG	DOP mineralization rate algal dependence	2.3×10^{-3}	$m^3 \cdot kg^{-1} \cdot s^{-1}$
KE	Background light attenuation	1.5×10^{-1}	m^{-1}
KECHL	Light attenuation factor for chlorophyll *a*	1.7×10^{4}	$m^2 \cdot kg^{-1}$
KHN	Organic nitrogen decay half-saturation constant	2.5×10^{-5}	$kg \cdot m^{-3}$
KHND	Mean half-saturation constant for nitrogen uptake by diatoms	2.5×10^{-5}	$kg \cdot m^{-3}$
KHNG	Mean half-saturation constant for nitrogen uptake by green algae	2.5×10^{-5}	$kg \cdot m^{-3}$
KHNNT	Half-saturation concentration of NH_4 required for nitrification	1.0×10^{-4}	$kg \cdot m^{-3}$
KHP	Organic phosphorus decay half-saturation constant	1.0×10^{-6}	$kg \cdot m^{-3}$
KHPD	Mean half-saturation constant for diatom phosphorus uptake	1.0×10^{-6}	$kg \cdot m^{-3}$
KHPG	Mean half-saturation constant for green algae phosphorus uptake	1.0×10^{-6}	$kg \cdot m^{-3}$
KHSD	Mean half-saturation constant for diatom silica uptake	3.0×10^{-5}	$kg \cdot m^{-3}$
KLC	Minimum hydrolysis rate of LOC	8.6×10^{-7}	s^{-1}
KLCALG	LOC hydrolysis rate algal dependence	0.0	$m^3 \cdot kg^{-1} \cdot s^{-1}$
KLN	Minimum hydrolysis rate of LON	2.3×10^{-7}	s^{-1}
KLNALG	LON hydrolysis rate algal dependence	0.0	$m^3 \cdot kg^{-1} \cdot s^{-1}$
KLP	Minimum hydrolysis rate of LOP	2.3×10^{-7}	s^{-1}
KLPALG	LOP hydrolysis rate algal dependence	2.3×10^{-3}	$m^3 \cdot kg^{-1} \cdot s^{-1}$
KRC	Minimum hydrolysis rate of ROC	5.0×10^{-8}	s^{-1}
KRCALG	ROC hydrolysis rate algal dependence	0.0	$m^3 \cdot kg^{-1} \cdot s^{-1}$
KRN	Minimum hydrolysis rate of RON	5.7×10^{-8}	s^{-1}
KRNALG	RON hydrolysis rate algal dependence	0.0	$m^3 \cdot kg^{-1} \cdot s^{-1}$
KRP	Minimum hydrolysis rate of ROP	5.7×10^{-8}	s^{-1}
KRPALG	ROP hydrolysis rate algal dependence	2.3×10^{-3}	$m^3 \cdot kg^{-1} \cdot s^{-1}$
KSUA	Particulate silica dissolution rate constant	5.0×10^{-8}	s^{-1}
KSZ	Half-saturation coefficient of zooplankton for phytoplankton	7.5×10^{-4}	$kg \cdot m^{-3}$
KTBD	Metabolism temperature dependence factor for diatoms	6.9×10^{-2}	$°C^{-1}$
KTBG	Metabolism temperature dependence factor for greens	6.9×10^{-2}	$°C^{-1}$
KTGD1	Effect of temperature below optimal temperature for diatoms	2.5×10^{-3}	$°C^{-2}$
KTGD2	Effect of temperature above optimal temperature for diatoms	6.0×10^{-3}	$°C^{-2}$
KTGG1	Effect of temperature below optimal temperature for greens	2.5×10^{-3}	$°C^{-2}$
KTGG2	Effect of temperature above optimal temperature for greens	1.0×10^{-2}	$°C^{-2}$
KTHDR	Hydrolysis temperature dependence	6.9×10^{-2}	$°C^{-1}$
KTMNL	Mineralization temperature dependence	6.9×10^{-2}	$°C^{-1}$
KTNT1	Effect of temperature below optimal temperature nitrification	9.0×10^{-2}	$°C^{-2}$
KTNT2	Effect of temperature above optimal temperature for nitrification	1.1×10^{-6}	$°C^{-2}$
KTSUA	Silica dissolution temperature rate constant	1.1×10^{-6}	$°C^{-1}$
NTM	Nitrification rate coefficient at optimal temperatures	1.2×10^{-12}	$kg \cdot m^{-3} \cdot s^{-1}$
PMD	Diatom production under optimal conditions	2.4×10^{-5}	s^{-1}
PMG	Green algae production under optimal conditions	2.3×10^{-5}	s^{-1}
TMD	Temperature of optimum growth for diatoms	2.0×10^{-1}	°C
TMG	Temperature of optimum growth for greens	2.0×10^{-1}	°C
TMNT	Optimal temperature for nitrification	2.0×10^{-1}	°C

Table 6.1. Eutrophication Parameters (Continued)

Key Name	Description	Default Value	Units
TRD	Metabolism reference temperature for diatoms	2.0×10^{1}	°C
TRG	Metabolism reference temperature for greens	2.0×10^{1}	°C
TRHDR	Reference temperature for hydrolysis	2.0×10^{1}	°C
TRMNL	Reference temperature for mineralization	2.0×10^{1}	°C
TRSUA	Silica dissolution reference temperature	2.0×10^{1}	°C
TZREF	Predation reference temperature	2.0×10^{1}	°C
ZDTH	Zooplankton death/die-off coefficient	3.5×10^{-7}	s^{-1}
ZTHET	Temperature coefficient for predation	1.05	None

The user has to specify which of the two nutrient growth dependency equations, Equation 6.4 or Equation 6.5, to use by setting the general parameter SILIM to specific values as shown in Table 3.2. The parameter SILIM is set by the user in the input deck. It is strongly recommended to use Equation 6.4 because it applies to all situations, regardless whether nitrogen is a growth limiting factor or not. Equation 6.5 is too restrictive to use because the user has to prove, using statistical analysis of field data, that nitrogen is not a growth limiting factor in the aquatic system being studied/simulated.

The diatoms are described using one of three different formulas. The first one uses the product of the silica limitation and the minimum of nitrogen and phosphorus:

$$f(N) = \left(\frac{Si}{k_{sat-Si} + Si}\right) \bullet \min\left[\left(\frac{NH_4 + NO_3}{k_{sat-N} + NH_4 + NO_3}\right), \left(\frac{P_{av}}{k_{sat-P} + P_{av}}\right)\right] \tag{6.7}$$

The second one uses the minimum of silica, nitrogen, and phosphorus:

$$f(N) = \min\left[\left(\frac{Si}{k_{sat-Si} + Si}\right), \left(\frac{NH_4 + NO_3}{k_{sat-N} + NH_4 + NO_3}\right), \left(\frac{P_{av}}{k_{sat-P} + P_{av}}\right)\right] \tag{6.8}$$

The last one uses the product of silica and phosphorus limitation only:

$$f(N) = \left(\frac{Si}{k_{sat-Si} + Si}\right) \bullet \left(\frac{P_{av}}{k_{sat-P} + P_{av}}\right) \tag{6.9}$$

k_{sat-Si} = half-saturation coefficient for silica (Si) uptake ($kg \cdot m^{-3}$) = KHSD in Table 6.1
min = minimum of the two expressions (separated by a comma) within the square brackets.
k_{sat-N} = half-saturation coefficient for nitrogen uptake ($kg \cdot m^{-3}$) = KHND in Table 6.1
k_{sat-P} = half-saturation coefficient for phosphorus uptake ($kg \cdot m^{-3}$) = KHPD in Table 6.1
Si = dissolved available silica concentration ($kg \cdot m^{-3}$)

Only one equation of the three listed above is used during a particular simulation. The choice of the equation depends on what assumptions are made about the lake. The user specifies which one to use by using the general parameter SILIM as shown in Table 3.2.

6.1.1.2 Temperature Growth Dependency

The temperature dependency is expressed using a Gaussian probability function. It is different from most Arrhenius temperature correction factors (Arrhenius, 1889) in that it is two-sided around Tm and has a different shape than the standard $\theta^{(T-20)}$. Thus:

$$f(T)=\begin{cases} \exp\left[-ktg1 \bullet \left(T - Tm\right)^2\right] & \text{where } T \leq Tm \\ \exp\left[-ktg2 \bullet \left(Tm - T\right)^2\right] & \text{where } T > Tm \end{cases} \tag{6.10}$$

where:

ktg1 = temperature effect below optimum temperature ($°C^{-2}$)
ktg2 = temperature effect above optimum temperature ($°C^{-2}$)
Tm = optimum temperature for phytoplankton growth (°C)
T = water temperature (°C)

The parameters ktg1, ktg2, and Tm are different for diatoms and greens. The parameter ktg1 is represented in the LM3 source code with the variable name KTGD1 for diatoms and KTGG1 for greens. The parameter ktg2 is represented with the variable name KTGD2 for diatoms and KTGG2 for greens. The parameter Tm is represented with the variable name TMD for diatoms and TMG for greens. Table 6.1 shows default values for the aforementioned parameters.

6.1.1.3 Light Growth Dependency

A number of equations have been proposed to describe the effect of light intensity on phytoplankton production. Steele's (1962) equation is one of the most commonly used expressions, while a light saturation equation (similar to the Monod equation) is also frequently used (Di Toro *et al*., 1971). We described light dependency in this model according to Steele's equation.

$$f(I) = \frac{I}{I_s} \exp\left[-\left(\frac{I}{I_s}\right) + 1\right] \tag{6.11}$$

where:

f(I) = light limitation (fraction between 0 and 1)
I = incident solar light intensity ($w \cdot m^{-2}$)
I_s = saturating light intensity ($w \cdot m^{-2}$) = ISMIN in Table 6.1

The Beer-Lambert equation (Parker, 1993) was used to estimate light penetration into the water.

$$I_z = I_o \bullet \exp\left(-k_{ess} \bullet z\right) \tag{6.12}$$

where:

I_z = the light intensity at depth z ($w \cdot m^{-2}$)
I_o = surface light intensity ($w \cdot m^{-2}$)
k_{ess} = light extinction coefficient (m^{-1})
z = depth (m)

Substituting Equation 6.12 into Equation 6.11:

$$f\left(I(z)\right) = \frac{I_o \bullet \exp\left(-k_{ess} \bullet z\right)}{I_s} \bullet \exp\left[-\frac{I_o \bullet \exp\left(-k_{ess} \bullet z\right)}{I_s} + 1\right] \tag{6.13}$$

This equation calculates the light limitation at an instantaneous time and at a specific depth. However, for models like ours, we need to estimate the light limitation at a certain cell (with a given depth range) and over a time period (the time step). We thus need to integrate this equation over time and depth. Di Toro *et al*. (1971) formulated an equation assuming a constant light intensity over the photoperiod. They integrated Steele's equation (Equation 6.14) over a 24-hour period and the total depth of a segment.

$$f\left(I_a\right) = \frac{2.718 \bullet fd}{k_{ess} \bullet \Delta z} \bullet \left[\exp\left(-\alpha_1\right) - \exp\left(-\alpha_0\right)\right] \tag{6.14}$$

where:

$$\alpha_0 = \frac{I_a}{I_s} \bullet \exp\left(-k_{ess} \bullet z_1\right) \tag{6.15}$$

$$\alpha_1 = \frac{I_a}{I_s} \bullet \exp\left(-k_{ess} \bullet z_2\right) \tag{6.16}$$

fd = the photoperiod = fraction day with sunlight (unitless)
I_a = average light intensity over the photoperiod ($w \cdot m^{-2}$)
Δz = thickness of water column segment = z_2 - z_1 (m)
z_1 = vertical distance from the water surface to the top face of water segment (m)
z_2 = vertical distance from the water surface to the bottom face of water segment (m)

This approach is commonly used, although it has been criticized for losing the power to represent midday surface inhibition (Di Toro *et al*., 1971; Kremer and Nixon, 1974).

The LM3 light formalism allows modelers to use different light intensity update intervals, from hourly to 12-hours. If one wants to estimate the light limitation for less than a day and the average light intensity of that period is known, one can solve Steele's equation by averaging it over the depth of a grid cell. Let z_1 and z_2 be the water depth of the top and bottom, respectively, of a grid cell. The averaging is performed as:

$$f(I_a) = \frac{1}{z_2 - z_1} \int_{z_1}^{z_2} \left(\frac{I_0 \bullet \exp(-k_{ess} \bullet z)}{I_s} \exp\left[-\frac{I_0 \bullet \exp(-k_{ess} \bullet z)}{I_s} + 1 \right] \right) dz \tag{6.17}$$

The solution is almost the same as Equation 6.14, without the fraction of daylight in the equation.

$$f(I_a) = \frac{2.718}{k_{ess} \bullet \Delta z} \bullet \left[\exp(-\alpha_1) - \exp(-\alpha_0) \right] \tag{6.18}$$

The average light intensity (I_a) here can be calculated as follows:

$$I_a = \frac{\int I_0(t)dt}{\int dt} \tag{6.19}$$

where:

$I_0(t)$ = measured incident solar radiation at time t ($w \cdot m^{-2}$)
t = time (hours)

and can thus be approximated by:

$$I_a = \frac{1}{n} \sum_{i=1}^{n} I_0(t_i) \tag{6.20}$$

where:

n = number of discrete time intervals at which I_0 is measured.

The light extinction coefficient, k_{ess}, as it appears in Equation 6.18 requires some further explanation. This coefficient actually represents the light attenuation in the water column. It is composed of two fractions: a background value dependent on both water color and concentration of suspended particles, and extinction due to light absorption by ambient chlorophyll.

$$k_{ess} = k_e + k_{echl} \sum_{i=d,g} \frac{PHYTO_i}{(CCHL)_i} \tag{6.21}$$

where:

k_e = background light attenuation (m^{-1})
k_{echl} = light attenuation factor for chlorophyll-*a* ($m^2 \cdot kg^{-1}$)

$PHYTO_i$ = phytoplankton concentration ($kg \cdot m^{-3}$)
$(CCHL)_i$ = carbon-to-chlorophyll ratio in phytoplankton group
d,g = diatoms, greens

The parameters, k_e and k_{echl} are constant input values specified by the user. Suggested values can be found in Table 6.1 under the source code names KE and KECHL. The variable $(CCHL)_i$ takes on two parameter values, one for diatoms and one for greens. The diatoms value is denoted by the source code name of CCHLD and the corresponding one for greens is denoted by the name CCHLG. These parameters are also constants and specified by the user. Suggested values can be found in Table 6.1.

Many of the eutrophication parameters default values shown in Table 6.1 were taken from the Chesapeake Bay eutrophication report (Cerco and Cole, 1994). Others were estimated values that did not have any association with that report or with any calibration exercise. The user is ultimately responsible for determining whether default values are appropriate for some of the parameters of a given simulation run. Strictly speaking, the default values are initialization values within the LM3 source code and are not meant to be a mandatory or even a recommended value.

6.1.2 Zooplankton Kinetics

Zooplankton predation is important in regulating the phytoplankton densities in Lake Michigan, especially during the stratified summer months (Scavia *et al*., 1988). Lake Michigan zooplankton are dominated by herbivorous species, with copepods making up the majority of the total biomass for most of the year. However, cladocerans exhibit significant peaks in the late summer and fall. Due to the limited zooplankton data reported for the lake and the lack of kinetic laboratory and field studies for Lake Michigan zooplankton populations, we avoided complex zooplankton equations (e.g., Bowie *et al*., 1985) and chose a relatively simple formulation. The equations describing herbivorous zooplankton growth are based on formulations from the literature (Bowie *et al*., 1985; Di Toro and Connolly, 1980; Di Toro and Matystik, 1980; Thomann and Mueller, 1987). Carnivorous zooplankton are not simulated in most versions of LM3 except 3.2.16 and 3.2.17; however, the LM3 versions 3.2.16 and 3.2.17 are still being developed/tested as of this writing. Thus for the present discussion, no carnivorous zooplankton is simulated.

The following equation was used:

$$\frac{dZ}{dt} = (\varepsilon \bullet k_{gz} - k_{dz}) \bullet Z \tag{6.22}$$

where:

Z = zooplankton concentration ($kg \cdot m^{-3}$)
t = time (s)
k_{gz} = zooplankton predation rate (s^{-1})
k_{dz} = mortality rate (s^{-1}) = ZDTH in Table 6.1
ε = assimilation efficiency = GREFF in Table 6.1

$$k_{gz} = k_{gzmax} \bullet \left(\frac{PHYTO_T}{ks + PHYTO_T} \right) \bullet \theta^{T-T_{rz}} \tag{6.23}$$

where:

k_{gzmax} = maximum growth rate (s^{-1}) = CGZ in Table 6.1
$PHYTO_T$ = total phytoplankton concentration = the sum of diatom and greens concentration ($kg \cdot m^{-3}$)
ks = half-saturation coefficient ($kg \cdot m^{-3}$) = KSZ in Table 6.1
θ = temperature correction factor = ZTHET in Table 6.1
T_{rz} = reference temperature (°C) = TZREF in Table 6.1

The maximum growth rate is a term that lumps the filtration and assimilation rates into a single term. The mortality term lumps respiration, excretion, and higher predation into a single term. We can thus write the overall equation:

$$\frac{dZ}{dt} = \left[\varepsilon \bullet k_{gzmax} \bullet \frac{PHYTO}{ks + PHYTO_T} \bullet \theta^{T-T_{rz}} - k_{dz} \right] \bullet Z \tag{6.24}$$

6.1.3 Carbon Interactions

Several carbon interactions are described in the model, including phytoplankton and zooplankton carbon, carbon loads from tributaries, shoreline erosion, and detrital carbon from plankton. The carbon state variables in this model are diatom, non-diatom, and zooplankton carbon; labile particulate organic carbon (LPOC); refractory particulate organic carbon (RPOC); and dissolved organic carbon (DOC). Carbon dioxide (CO_2) was not simulated, although a mineralization reaction was included. Diatom and non-diatom carbon were simulated, as described in the previous section. LPOC refers to the organic detrital carbon from the phytoplankton species which breaks down, as the name implies, relatively rapidly. In contrast, the RPOC is the combination of the fraction of the plankton breaking down, as well as other forms of organic carbon in the system (carbon from tributaries, the sediments, etc). These forms of carbon break down more slowly but are not totally refractory. In the equations, we specify the fractions of labile and refractory carbon. We assume that phytoplankton utilize CO_2 as the carbon source during photosynthesis and release carbon as dissolved (CO_2 and DOC) and particulate (RPOC and LPOC).

6.1.3.1 Phytoplankton Mortality and Decay

Phytoplankton respiration and non-predatory mortality were grouped together in the model as a "mortality" term. The release of carbon from these processes was split into different fractions of dissolved and particulate organic carbon.

$$\frac{dDOC}{dt} = f_{cdd} \bullet k_{bmd} \bullet DIA + f_{cdg} \bullet k_{bmg} \bullet GRE \tag{6.25}$$

$$\frac{dRPOC}{dt} = f_{crd} \bullet k_{bmd} \bullet DIA + f_{crg} \bullet k_{bmg} \bullet GRE \tag{6.26}$$

$$\frac{dLPOC}{dt} = f_{cld} \bullet k_{bmd} \bullet DIA + f_{clg} \bullet k_{bmg} \bullet GRE \tag{6.27}$$

where:

DIA = diatoms concentration (kg · m^{-3})
GRE = greens concentration (kg · m^{-3})
f_{cdd} = fraction of basal metabolism exuded as DOC by diatoms = FCDD in Table 6.1
f_{cdg} = fraction of basal metabolism exuded as DOC by greens = FCDG in Table 6.1
f_{crd} = fraction of refractory POC produced by diatoms metabolism = FCRD in Table 6.1
f_{crg} = fraction of refractory POC produced by greens metabolism = FCRG in Table 6.1
f_{cld} = fraction of labile POC produced by diatoms metabolism = FCLD in Table 6.1
f_{clg} = fraction of labile POC produced by greens metabolism = FCLG in Table 6.1
k_{bmd} = base metabolic rate for diatoms (s^{-1})
k_{bmg} = base metabolic rate for greens (s^{-1})

The base metabolic rates, k_{bmd} and k_{bmg}, correspond to the term k_d of Equation 6.2. They are calculated using the following equations:

$$k_{bmd} = b_{mrd} \bullet \exp\left[k_{tbd} \bullet \left(T - T_{rd}\right)\right] \tag{6.28}$$

$$k_{bmg} = b_{mrg} \bullet \exp\left[k_{tbg} \bullet \left(T - T_{rg}\right)\right] \tag{6.29}$$

where:

b_{mrd} = diatom base metabolic rate (s^{-1}) = BMRD in Table 6.1
b_{mrg} = green base metabolic rate (s^{-1}) = BMRG in Table 6.1
k_{tbd} = metabolism temperature dependence factor for diatoms (°C^{-1}) = KTBD in Table 6.1
k_{tbg} = metabolism temperature dependence factor for greens (°C^{-1}) = KTBG in Table 6.1
T = water temperature (°C)
T_{rd} = metabolism reference temperature for diatoms (°C) = TRD in Table 6.1
T_{rg} = metabolism reference temperature for greens (°C) = TRG in Table 6.1

However, phytoplankton carbon is also converted to detrital and DOC through predation (messy feeding) and zooplankton-imposed mortality.

$$\frac{dDOC}{dt} = f_{cdp} \bullet \left(1 - \varepsilon\right) \bullet k_{gz} \bullet Z \tag{6.30}$$

$$\frac{dRPOC}{dt} = f_{crp} \bullet (1-\varepsilon) \bullet k_{gz} \bullet Z \tag{6.31}$$

$$\frac{dLPOC}{dt} = f_{clp} \bullet (1-\varepsilon) \bullet k_{gz} \bullet Z \tag{6.32}$$

where:

f_{cdp} = fraction of DOC from predation = FCDP in Table 6.1
f_{crp} = fraction of RPOC from predation = FCRP in Table 6.1
f_{clp} = fraction of LPOC from predation = FCLP in Table 6.1

6.1.3.2 Zooplankton Mortality and Decay

The zooplankton mortality term included respiration, non-predatory mortality, and predatory mortality. We assume that the detrital zooplankton carbon consists of dissolved, labile particulate, and refractory particulate fractions.

$$\frac{dDOC}{dt} = f_{cdz} \bullet k_{dz} \bullet Z \tag{6.33}$$

$$\frac{dLPOC}{dt} = f_{clz} \bullet k_{dz} \bullet Z \tag{6.34}$$

$$\frac{dRPOC}{dt} = f_{crz} \bullet k_{dz} \bullet Z \tag{6.35}$$

where:

f_{cdz} = fraction of DOC from zooplankton mortality = FCDZ in Table 6.1
f_{clz} = fraction of LPOC from zooplankton mortality = FCLZ in Table 6.1
f_{crz} = fraction of RPOC from zooplankton mortality = FCRZ in Table 6.1

6.1.3.3 Hydrolysis and Mineralization

Particulate fractions (both labile and refractory) are hydrolyzed to DOC, while DOC mineralizes to CO_2.

Because we do not explicitly model bacteria in this model, their breakdown of carbon was modeled by including a dependency on the phytoplankton, which acted as a surrogate of the heterotrophic bacterial activity in the lake. We also calculated a temperature limitation to the hydrolysis and mineralization. The equations can be written as follows:

$$Tf_{mnl} = \exp\left[Tk_{mnl} \cdot \left(T - Tr_{mnl}\right)\right] \tag{6.36}$$

$$Tf_{hdr} = \exp\left[Tk_{hdr} \cdot \left(T - Tr_{hdr}\right)\right] \tag{6.37}$$

where:

Tf_{mnl} = temperature correction for mineralization (unitless)
Tf_{hdr} = temperature correction for hydrolysis (unitless)
Tk_{mnl} = mineralization temperature coefficient ($°C^{-1}$) = KTMNL in Table 6.1
Tk_{hdr} = hydrolysis temperature coefficient ($°C^{-1}$) = KTHDR in Table 6.1
Tr_{mnl} = optimum temperature correction for mineralization (°C) = TRMNL in Table 6.1
Tr_{hdr} = optimum temperature correction for hydrolysis (°C) = TRHDR in Table 6.1

$$\begin{aligned}\frac{dDOC}{dt} = Tf_{hdr} \cdot \Big[&\left(k_{lc} + k_{lcp} \cdot PHYTO_T\right) \cdot LPOC \\ &+ \left(k_{rc} + k_{rcp} \cdot PHYTO_T\right) \cdot RPOC\Big] \\ &- Tf_{mnl} \cdot \left(k_{dc} + k_{dcp} \cdot PHYTO_T\right) \cdot DOC\end{aligned} \tag{6.38}$$

$$-\frac{dRPOC}{dt} = Tf_{hdr} \cdot \left(k_{rc} + k_{rcp} \cdot PHYTO_T\right) \cdot RPOC \tag{6.39}$$

$$-\frac{dLPOC}{dt} = Tf_{hdr} \cdot \left(k_{lc} + k_{lcp} \cdot PHYTO_T\right) \cdot LPOC \tag{6.40}$$

where:

DOC = dissolved organic carbon concentration ($kg \cdot m^{-3}$)
LPOC = labile particulate organic carbon concentration ($kg \cdot m^{-3}$)
RPOC = refractory particulate organic carbon concentration ($kg \cdot m^{-3}$)
k_{dc} = DOC minimum mineralization rate (s^{-1}) = KDC in Table 6.1
k_{dcp} = DOC mineralization relating to phytoplankton ($m^3 \cdot kg\ C^{-1} \cdot s^{-1}$) = KDCALG in Table 6.1
k_{rc} = RPOC minimum hydrolysis rate (s^{-1}) = KRC in Table 6.1
k_{rcp} = RPOC hydrolysis relating to phytoplankton ($m^3 \cdot kg\ C^{-1} \cdot s^{-1}$) = KRCALG in Table 6.1
k_{lc} = LPOC minimum hydrolysis rate (s^{-1}) = KLC in Table 6.1
k_{lcp} = LPOC hydrolysis relating to phytoplankton ($m^3 \cdot kg\ C^{-1} \cdot s^{-1}$) = KLCALG in Table 6.1

From the last two equations, note that the gain in DOC equals the sum of the loss of RPOC and LPOC.

6.1.4 Phosphorus

In LM3, phosphorus can exist as one of four species (in addition to being tied up in the phytoplankton). Note that all four forms are in the same oxidation state; thus no oxidation reactions occur. The forms are SRP, DOP, and two forms of particulate organic phosphorus (POP), a labile (LPOP), and a refractory (RPOP) form. SRP and a small fraction of the DOP are taken up by the phytoplankton during production (photosynthesis). They are released due to mortality and predation. Particulate phosphorus is hydrolyzed to DOP, and DOP is hydrolyzed to SRP.

6.1.4.1 Phosphorus Uptake by Phytoplankton

SRP Uptake:

$$\frac{dSRP}{dt} = -r_{pc} \bullet \left(k_{g,dia} \bullet DIA + k_{g,gre} \bullet GRE\right) \bullet \frac{SRP}{SRP + f_{dop} \bullet DOP} \tag{6.41}$$

DOP Uptake:

$$\frac{dDOP}{dt} = -r_{pc} \bullet \left(k_{g,dia} \bullet DIA + k_{g,gre} \bullet GRE\right) \bullet \frac{f_{dop} \bullet DOP}{f_{dop} \bullet DOP + SRP} \tag{6.42}$$

where:

r_{pc} = the P:C ratio (unitless) = APCP in Table 6.1
$k_{g,dia}$ = diatoms growth rate (s^{-1})
$k_{g,gre}$ = greens growth rate (s^{-1})
DOP = dissolved organic phosphorus concentration ($kg \cdot m^{-3}$)
SRP = soluble reactive phosphorus concentration ($kg \cdot m^{-3}$)
f_{dop} = available fraction of DOP = AVFRAC in Table 6.1

The growth rates for diatoms and greens are given by Equation 6.3.

An interesting concept, common in many phytoplankton models, is the way in which the nutrients, including phosphorus, are accounted for within the phytoplankton. The model keeps track of the carbon and uses a constant carbon:nutrient relationship to make these determinations.

6.1.4.2 Phosphorus Release Due to Phytoplankton Metabolism/Mortality

Phytoplankton in the water column are hydrolyzed and mineralized to all four phosphorus forms. During algal metabolic/mortality processes, phytoplankton-P is converted to particulate and dissolved organic forms as well as directly to SRP.

$$\frac{dSRP}{dt} = r_{pc} \bullet \left(f_{srp,dia} \bullet k_{bmd} \bullet DIA + f_{srp,gre} \bullet k_{bmg} \bullet GRE\right) \tag{6.43}$$

$$\frac{dDOP}{dt} = r_{pc} \bullet \left(f_{dop,dia} \bullet k_{bmd} \bullet DIA + f_{dop,gre} \bullet k_{bmg} \bullet GRE\right) \tag{6.44}$$

$$\frac{dLPOP}{dt} = r_{pc} \bullet \left(f_{lpop,dia} \bullet k_{bmd} \bullet DIA + f_{lpop,gre} \bullet k_{bmg} \bullet GRE\right) \tag{6.45}$$

$$\frac{dRPOP}{dt} = r_{pc} \bullet \left(f_{rpop,dia} \bullet k_{bmd} \bullet DIA + f_{rpop,gre} \bullet k_{bmg} \bullet GRE\right) \tag{6.46}$$

where:

$f_{srp,dia}$ = fraction of SRP from diatoms metabolism/mortality = FPID in Table 6.1
$f_{srp,gre}$ = fraction of SRP from greens metabolism/mortality = FPIG in Table 6.1
$f_{dop,dia}$ = fraction of DOP from diatoms metabolism/mortality = FPDD in Table 6.1
$f_{dop,gre}$ = fraction of DOP from greens metabolism/mortality = FPDG in Table 6.1
$f_{lpop,dia}$ = fraction of LPOP from diatoms metabolism/mortality = FPLD in Table 6.1
$f_{lpop,gre}$ = fraction of LPOP from greens metabolism/mortality = FPLG in Table 6.1
$f_{rpop,dia}$ = fraction of RPOP from diatoms metabolism/mortality = FPRD in Table 6.1
$f_{rpop,gre}$ = fraction of RPOP from greens metabolism/mortality = FPRG in Table 6.1
DIA = diatoms concentration ($kg \cdot m^{-3}$)
GRE = greens concentration ($kg \cdot m^{-3}$)
k_{bmd} = base metabolic rate for diatoms, given by Equation 6.28
k_{bmg} = base metabolic rate for greens, given by Equation 6.29

6.1.4.3 Phosphorus Release Due to Zooplankton Predation

During phytoplankton predation, zooplankton assimilate only a fraction of the phytoplankton and the remainder of the detrital phytoplankton is released directly to the water. This process is commonly referred to as "messy feeding." The phosphorus is released in both the dissolved and particulate forms.

$$\frac{dSRP}{dt} = f_{pip} \bullet (1-\varepsilon) \bullet r_{pc} \bullet k_{gz} \bullet Z \tag{6.47}$$

$$\frac{dDOP}{dt} = f_{pdp} \bullet (1-\varepsilon) \bullet r_{pc} \bullet k_{gz} \bullet Z \tag{6.48}$$

$$\frac{dLPOP}{dt} = f_{plp} \bullet (1-\varepsilon) \bullet r_{pc} \bullet k_{gz} \bullet Z \tag{6.49}$$

$$\frac{dRPOP}{dt} = f_{prp} \bullet (1-\varepsilon) \bullet r_{pc} \bullet k_{gz} \bullet Z \tag{6.50}$$

where:

f_{pip} = fraction of SRP from predation = FPIP in Table 6.1
f_{pdp} = fraction of DOP from predation = FPDP in Table 6.1
f_{plp} = fraction of LPOP from predation = FPLP in Table 6.1
f_{prp} = fraction of RPOP from predation = FPRP in Table 6.1
k_{gz} = zooplankton predation rate (s^{-1}), given by Equation 6.23
Z = zooplankton concentration ($kg \cdot m^{-3}$)

6.1.4.4 Phosphorus Release Due to Zooplankton Mortality

The model also includes equations to describe zooplankton mortality. Phosphorus is released to the water column in both the dissolved and particulate forms.

$$\frac{dSRP}{dt} = f_{piz} \bullet r_{pc} \bullet k_{dz} \bullet Z \tag{6.51}$$

$$\frac{dDOP}{dt} = f_{pdz} \bullet r_{pc} \bullet k_{dz} \bullet Z \tag{6.52}$$

$$\frac{dLPOP}{dt} = f_{plz} \bullet r_{pc} \bullet k_{dz} \bullet Z \tag{6.53}$$

$$\frac{dRPOP}{dt} = f_{prz} \bullet r_{pc} \bullet k_{dz} \bullet Z \tag{6.54}$$

where:

f_{piz} = fraction of SRP from zooplankton mortality = FPIZ in Table 6.1
f_{pdz} = fraction of DOP from zooplankton mortality = FPDZ in Table 6.1
f_{plz} = fraction of LPOP from zooplankton mortality = FPLZ in Table 6.1
f_{prz} = fraction of RPOP from zooplankton mortality = FPRZ in Table 6.1

6.1.4.5 Hydrolysis and Mineralization of Phosphorus

Particulate phosphorus is hydrolyzed to DOP and DOP is mineralized back to SRP as follows:

$$\frac{dSRP}{dt} = \left(k_{dp} + k_{dpa} \bullet PHYTO_T \bullet \frac{k_{sat-pt}}{SRP + k_{sat-pt}} \right) \bullet Tf_{mnl} \bullet DOP \tag{6.55}$$

$$\frac{dDOP}{dt} = \left(k_{lp} + k_{lpa} \bullet PHYTO_T \bullet \frac{k_{sat-pt}}{SRP + k_{sat-pt}} \right) \bullet Tf_{hdr} \bullet LPOP$$
$$+ \left(k_{rp} + k_{rpa} \bullet PHYTO_T \bullet \frac{k_{sat-pt}}{SRP + k_{sat-pt}} \right) \bullet Tf_{hdr} \bullet RPOP$$
$$- \left(k_{dp} + k_{dpa} \bullet PHYTO_T \bullet \frac{k_{sat-pt}}{SRP + k_{sat-pt}} \right) \bullet Tf_{mnl} \bullet DOP \qquad (6.56)$$

$$-\frac{dLPOP}{dt} = \left(k_{lp} + k_{lpa} \bullet PHYTO_T \bullet \frac{k_{sat-pt}}{SRP + k_{sat-pt}} \right) \bullet Tf_{hdr} \bullet LPOP \qquad (6.57)$$

$$-\frac{dRPOP}{dt} = \left(k_{rp} + k_{rpa} \bullet PHYTO_T \bullet \frac{k_{sat-pt}}{SRP + k_{sat-pt}} \right) \bullet Tf_{hdr} \bullet RPOP \qquad (6.58)$$

where:

DOP = dissolved organic phosphorus concentration ($kg \cdot m^{-3}$)
LPOP = labile particulate organic phosphorus concentration ($kg \cdot m^{-3}$)
RPOP = refractory particulate organic phosphorus concentration ($kg \cdot m^{-3}$)
k_{dp} = DOP mineralization coefficient (s^{-1}) = KDP in Table 6.1
k_{lp} = LPOP hydrolysis coefficient (s^{-1}) = KLP in Table 6.1
k_{rp} = RPOP hydrolysis coefficient (s^{-1}) = KRP in Table 6.1
k_{dpa} = DOP mineralization coefficient algal dependence ($m^3 \cdot kg^{-1} \cdot s^{-1}$) = KDPALG in Table 6.1
k_{lpa} = LPOP hydrolysis coefficient algal dependence ($m^3 \cdot kg^{-1} \cdot s^{-1}$) = KLPALG in Table 6.1
k_{rpa} = RPOP hydrolysis coefficient algal dependence ($m^3 \cdot kg^{-1} \cdot s^{-1}$) = KRPALG in Table 6.1
k_{sat-pt} = mean saturation coefficient of algal classes for SRP ($kg \cdot m^{-3}$) = KHP in Table 6.1
Tf_{mnl} = temperature correction for mineralization (unitless), given by Equation 6.36
Tf_{hdr} = temperature correction for hydrolysis (unitless), given by Equation 6.37

6.1.5 Nitrogen

The transformation of nitrogen is similar to phosphorus, although nitrogen exists in more than one oxidation state. Algal nitrogen is released as two forms of particulate organic nitrogen (PON) (labile [LPON], refractory [RPON]), dissolved organic nitrogen (DON), and ammonia-nitrogen (NH_4). Particulate forms are hydrolyzed to DON. DON is further mineralized to NH_4, and NH_4 is oxidized to NO_3.

6.1.5.1 Nitrogen Uptake by Phytoplankton

In our model, we assume that phytoplankton have no preference between NH_4 and NO_3 as a nitrogen source.

$$\frac{dDIN}{dt} = -r_{nc} \bullet \left(k_{g,dia} \bullet DIA + k_{g,gre} \bullet GRE\right) \tag{6.59}$$

where:

DIN = $NH_4 + NO_3$
NH_4 = ammonia concentration (kg · m^{-3})
NO_3 = nitrate concentration (kg · m^{-3})
r_{nc} = N:C ratio = ANCP in Table 6.1
$k_{g,dia}$ = diatoms growth rate (s^{-1}), given by Equation 6.3
$k_{g,gre}$ = greens growth rate (s^{-1}), given by Equation 6.3

Because we assume no preference, then

$$\frac{dNH_4}{dt} = -\left[\frac{NH_4}{DIN}\right] \bullet r_{nc} \bullet \left(k_{g,dia} \bullet DIA + k_{g,gre} \bullet GRE\right) \tag{6.60}$$

$$\frac{dNO_3}{dt} = -\left[\frac{NO_3}{DIN}\right] \bullet r_{nc} \bullet \left(k_{g,dia} \bullet DIA + k_{g,gre} \bullet GRE\right) \tag{6.61}$$

6.1.5.2 Nitrogen Release Due to Phytoplankton Metabolism/Mortality

Similar to phosphorus, nitrogen bound to phytoplankton can be released as particulate organic, dissolved organic, and NH_4 forms.

$$\frac{dNH_4}{dt} = r_{nc} \bullet \left(f_{nh4,dia} \bullet k_{bmd} \bullet DIA + f_{nh4,gre} \bullet k_{bmg} \bullet GRE\right) \tag{6.62}$$

$$\frac{dDON}{dt} = r_{nc} \bullet \left(f_{don,dia} \bullet k_{bmd} \bullet DIA + f_{don,gre} \bullet k_{bmg} \bullet GRE\right) \tag{6.63}$$

$$\frac{dLPON}{dt} = r_{nc} \bullet \left(f_{lpon,dia} \bullet k_{bmd} \bullet DIA + f_{lpon,gre} \bullet k_{bmg} \bullet GRE\right) \tag{6.64}$$

$$\frac{dRPON}{dt} = r_{nc} \bullet \left(f_{rpon,dia} \bullet k_{bmd} \bullet DIA + f_{rpon,gre} \bullet k_{bmg} \bullet GRE\right) \tag{6.65}$$

where:

$f_{nh4,dia}$ = fraction of NH_4 produced from diatoms metabolic process = FNID in Table 6.1
$f_{nh4,gre}$ = fraction of NH_4 produced from greens metabolic process = FNIG in Table 6.1
$f_{don,dia}$ = fraction of DON from diatoms metabolic process = FNDD in Table 6.1
$f_{don,gre}$ = fraction of DON from greens metabolic process = FNDG in Table 6.1
$f_{lpon,dia}$ = fraction of LPON from diatoms metabolic process = FNLD in Table 6.1
$f_{lpon,gre}$ = fraction of LPON from greens metabolic process = FNLG in Table 6.1
$f_{rpon,dia}$ = fraction of RPON from diatoms metabolic process = FNRD in Table 6.1
$f_{rpon,gre}$ = fraction of RPON from greens metabolic process = FNRG in Table 6.1

6.1.5.3 Nitrogen Release Due to Zooplankton Predation

As described for carbon and phosphorus, the nitrogen balance is affected by the zooplankton through "messy feeding" and zooplankton mortality. For zooplankton grazing:

$$\frac{dDON}{dt} = f_{ndp} \bullet (1-\varepsilon) \bullet r_{nc} \bullet k_{gz} \bullet Z \tag{6.66}$$

$$\frac{dNH_4}{dt} = f_{nip} \bullet (1-\varepsilon) \bullet r_{nc} \bullet k_{gz} \bullet Z \tag{6.67}$$

$$\frac{dLPON}{dt} = f_{nlp} \bullet (1-\varepsilon) \bullet r_{nc} \bullet k_{gz} \bullet Z \tag{6.68}$$

$$\frac{dRPON}{dt} = f_{nrp} \bullet (1-\varepsilon) \bullet r_{nc} \bullet k_{gz} \bullet Z \tag{6.69}$$

where:

f_{nip} = fraction of NH_4 from predation = FNIP in Table 6.1
f_{ndp} = fraction of DON from predation = FNDP in Table 6.1
f_{nlp} = fraction of LPON from predation = FNLP in Table 6.1
f_{nrp} = fraction of RPON from predation = FNRP in Table 6.1

6.1.5.4 Nitrogen Release Due to Zooplankton Mortality

The release of nitrogen during zooplankton mortality can be expressed similarly to that of phosphorus.

$$\frac{dNH_4}{dt} = f_{niz} \bullet r_{nc} \bullet k_{dz} \bullet Z \tag{6.70}$$

$$\frac{dDON}{dt} = f_{ndz} \bullet r_{nc} \bullet k_{dz} \bullet Z \tag{6.71}$$

$$\frac{dLPON}{dt} = f_{nlz} \bullet r_{nc} \bullet k_{dz} \bullet Z \tag{6.72}$$

$$\frac{dRPON}{dt} = f_{nrz} \bullet r_{nc} \bullet k_{dz} \bullet Z \tag{6.73}$$

where:

f_{niz} = fraction of NH_4 from zooplankton mortality = FNIZ in Table 6.1
f_{ndz} = fraction of DON from zooplankton mortality = FNDZ in Table 6.1
f_{nlz} = fraction of LPON from zooplankton mortality = FNLZ in Table 6.1
f_{nrz} = fraction of RPON from zooplankton mortality = FNRZ in Table 6.1

6.1.5.5 Hydrolysis and Mineralization of Nitrogen

Particulate nitrogen is hydrolyzed to DON and DON is mineralized back to NH_4.

$$\frac{dNH_4}{dt} = \left(k_{dn} + k_{dna} \bullet PHYTO_T \bullet \frac{k_{sat-decayN}}{NH_4 + NO_3 + k_{sat-decayN}} \right) \bullet Tf_{mnl} \bullet DON \tag{6.74}$$

$$\begin{aligned} \frac{dDON}{dt} = & \left(k_{ln} + k_{lna} \bullet PHYTO_T \bullet \frac{k_{sat-decayN}}{NH_4 + NO_3 + k_{sat-decayN}} \right) \bullet Tf_{hdr} \bullet LPON \\ & + \left(k_{rn} + k_{rna} \bullet PHYTO_T \bullet \frac{k_{sat-decayN}}{NH_4 + NO_3 + k_{sat-decayN}} \right) \bullet Tf_{hdr} \bullet RPON \\ & - \left(k_{dn} + k_{dna} \bullet PHYTO_T \bullet \frac{k_{sat-decayN}}{NH_4 + NO_3 + k_{sat-decayN}} \right) \bullet Tf_{mnl} \bullet DON \end{aligned} \tag{6.75}$$

$$-\frac{dLPON}{dt} = \left(k_{ln} + k_{lna} \bullet PHYTO_T \bullet \frac{k_{sat-decayN}}{NH_4 + NO_3 + k_{sat-decayN}} \right) \bullet Tf_{hdr} \bullet LPON \tag{6.76}$$

$$-\frac{dRPON}{dt} = \left(k_{rn} + k_{rna} \bullet PHYTO_T \bullet \frac{k_{sat-decayN}}{NH_4 + NO_3 + k_{sat-decayN}} \right) \bullet Tf_{hdr} \bullet RPON \tag{6.77}$$

where:

DON	=	dissolved organic nitrogen concentration (kg · m^{-3})
LPON	=	labile particulate organic nitrogen concentration (kg · m^{-3})
RPON	=	refractory particulate organic nitrogen concentration (kg · m^{-3})
k_{dn}	=	minimum DON mineralization rate (s^{-1}) = KDN in Table 6.1
k_{dna}	=	DON mineralization rate algal dependence (m^3 · kg^{-1} · s^{-1}) = KDNALG in Table 6.1
k_{ln}	=	minimum hydrolysis rate of LPON (s^{-1}) = KLN in Table 6.1
k_{lna}	=	LPON hydrolysis rate algal dependence (m^3 · kg^{-1}· s^{-1}) = KLNALG in Table 6.1
k_{rn}	=	minimum hydrolysis rate of RPON (s^{-1}) = KRN in Table 6.1
k_{rna}	=	RPON hydrolysis rate algal dependence (m^3 · kg^{-1} · s^{-1}) = KRNALG in Table 6.1
$k_{sat\text{-}decayN}$	=	organic nitrogen decay half-saturation constant (kg · m^{-3}) = KHN in Table 6.1
Tf_{mnl}	=	temperature correction for mineralization (unitless), given by Equation 6.36
Tf_{hdr}	=	temperature correction for hydrolysis (unitless), given by Equation 6.37

6.1.5.6 Nitrification

Nitrification is the biological oxidation of ammonia into nitrite followed by the oxidation of these nitrites into nitrates. In LM3, the modeling of the oxidation of NH_4 is simplified by assuming that ammonia, NH_4, oxidizes directly into nitrate, NO_3. The mathematical representation of the nitrification process used in LM3 is:

$$Tf_n = \begin{cases} \exp\left[-ktnt1 \bullet \left(T - Tnt \right)^2 \right] & \text{where } T < Tnt \\ \exp\left[-ktnt2 \bullet \left(Tnt - T \right)^2 \right] & \text{where } T \geq Tnt \end{cases} \tag{6.78}$$

where

Tf_n	=	nitrification temperature dependence function (unitless)
T	=	water temperature (°C)
Tnt	=	optimal temperature for nitrification (°C) = TMNT in Table 6.1
ktnt1	=	effect of temperature below optimal temperature for nitrification (°C^{-2}) = KTNT1 in Table 6.1
ktnt2	=	effect of temperature above optimal temperature for nitrification (°C^{-2}) = KTNT2 in Table 6.1

$$\frac{dNH_4}{dt} = -k_{ntm} \bullet Tf_n \bullet \frac{NH_4}{NH_4 + k_{sat-nitN}} \tag{6.79}$$

$$\frac{dNO_3}{dt} = k_{ntm} \bullet Tf_n \bullet \frac{NH_4}{NH_4 + k_{sat-nitN}} \tag{6.80}$$

where

k_{ntm} = nitrification rate coefficient at optimal temperature ($kg \cdot m^{-3} \cdot s^{-1}$)•= NTM in Table 6.1
NH_4 = ammonia concentration ($kg \cdot m^{-3}$)
$k_{sat-nitN}$ = half-saturation concentration of NH_4 required for nitrification ($kg \cdot m^{-3}$) = KHNNT in Table 6.1•

6.1.6 Silica

The behavior of silica is similar to that of phosphorus and nitrogen. Two silica species, particulate biogenic silica or unavailable silica (SU) and dissolved available silica (SA), are simulated in the lake. Dissolved silica is utilized by phytoplankton, while biogenic silica is released *via* phytoplankton mortality. Dissolved and biogenic silica are released *via* predation upon phytoplankton by zooplankton. Biogenic silica is assumed to mineralize to dissolved silica. The major difference from the other nutrients is that only diatoms have a silica dependency.

The diatom silica consumption can be written as follows:

$$\frac{dSA}{dt} = -r_{sc} \bullet k_{g,dia} \bullet DIA \tag{6.81}$$

where:

r_{sc} = Si:C ratio = ASCD in Table 6.1
$k_{g,dia}$ = diatoms growth rate (s^{-1}); see Equation 6.3
DIA = diatoms concentration ($kg \cdot m^{-3}$)

Like the other nutrients, silica is released *via* diatom mortality.

$$\frac{dSU}{dt} = r_{sc} \bullet k_{bmd} \bullet DIA \tag{6.82}$$

Both classes of silica can be released *via* zooplankton predation.

$$\frac{dSA}{dt} = f_{sap} \bullet r_{sc} \bullet k_{gz} \bullet Z \bullet \frac{DIA}{PHYTO_T} \tag{6.83}$$

$$\frac{dSU}{dt} = f_{sup} \bullet r_{sc} \bullet k_{gz} \bullet Z \bullet \frac{DIA}{PHYTO_T} \tag{6.84}$$

where:

f_{sap} = fraction of SA from predation = FSAP in Table 6.1
f_{sup} = 1 - f_{sap} = fraction of SU from predation
k_{gz} = zooplankton predation rate (s^{-1}), given by Equation 6.23
Z = zooplankton concentration ($kg \cdot m^{-3}$)

Biogenic silica is mineralized to dissolved silica. This process is described by the following equations:

$$\frac{dSA}{dt} = k_{sua} \bullet SU \bullet EXP\left[kt_{sua} \bullet \left(T - Tr_{sua}\right)\right] \tag{6.85}$$

$$\frac{dSU}{dt} = -k_{sua} \bullet SU \bullet EXP\left[kt_{sua} \bullet \left(T - Tr_{sua}\right)\right] \tag{6.86}$$

where:

k_{sua} = particulate silica dissolution rate constant (s^{-1}) = KSUA in Table 6.1
kt_{sua} = silica dissolution temperature rate constant ($°C^{-1}$) = KTSUA in Table 6.1
Tr_{sua} = silica dissolution reference temperature (°C) = TRSUA in Table 6.1
SU = particulate biogenic silica concentration ($kg \cdot m^{-3}$)

We assume that no silica accumulates within the zooplankton. Thus there are no terms for silica release from zooplankton mortality.

CHAPTER 7

SEDIMENT TRANSPORT

Sediment transport is a very important component of water quality modeling. It describes the physical processes that are behind the movement of particles to and from the water column and sediment bed.

One of the sediment transport processes that describes the movement of particles from the water column into the sediment bed is gravitational settling. In simple terms, gravitational settling is the downward movement of particles caused by the gravitational pull that the Earth exerts on particles. The description of settling is not exactly a simple one because particles are not free falling, that is, they are not falling in a vacuum under the influence of the gravitational force; instead, particles are falling through a fluid, water, with various properties such as viscosity and temperature affecting downward movement. Thus gravitational settling in water is in reality a complex function of temperature, viscosity, particle density, particle geometry, and water currents; water is always in a state of motion in Lake Michigan.

There is an equation that has been derived to describe the settling of particles in water, but that equation assumes that the particles are spherical in shape. In nature, we seldom find things that are perfect spheres. The equation is known as Stokes' Law and is used to get rough estimates of settling velocities for different types of particles. In practice, modelers prefer to specify a settling velocity for each settling particle as an input parameter. The input value is adjusted until a decent fit to the observed data is achieved. In LM3, settling is a user-specified velocity value with units of m/s. Each type of settling particle is assigned its own value. The settling velocities are time- and space-independent; that is, each settling particle is assigned a settling velocity that is the same for all the grid cells in the model and is constant throughout the simulation. Generally speaking settling velocities are not constant because in the real world they are functions of water temperature and particle size. However, modelers at the United States Environmental Protection Agency (USEPA), Large Lakes Research Station (LLRS) decided to use constant settling velocities, which could be adjusted during calibration, because of lack of laboratory experiments and *in situ* settling tests in Lake Michigan.

Another important process of sediment transport is sediment resuspension. This process refers to the physical movement of particles as they are lifted off the sediment bed and transported into the water column. Sediment resuspension is triggered by the continuous action of shear stress upon the sediment bed. Shear stress is a force per unit area. This force per unit area is created by the combined action of waves and currents in the water column. However, this combined action is not a simple linear superposition of these two disturbances in the water; instead, it is a complicated non-linear interaction between waves and current energy. When

shear stress exceeds a critical value, also known as critical shear stress, particles in the sediment bed will start resuspending. Shear stress is used in conjunction with sediment properties to calculate resuspension velocities in units of m/s. These resuspension velocities are used with sediment concentrations to calculate the resuspension mass rate for each cell in the surficial sediment grid.

The following section contains an in-depth discussion of shear stress. The discussion is somewhat technical, but it offers an insight on the calculation done within the application known as SEDZL. This application is used to generate resuspension fluxes. These fluxes are then converted to velocities which, in turn, are used as input values to LM3. Only the branches 3.2.15, 3.2.18, and 3.2.19 of LM3 are set up to use resuspension fluxes generated by SEDZL; the other LM3 branches have not been modified to make use of these fluxes as of this writing. See Section 7.3.4.1 for a detailed description of how the resuspension fluxes are converted to resuspension velocities.

7.1 Outline of the Bottom Shear Stress Calculation

7.1.1 Introduction

The following discussion outlines the calculation of the bottom shear stress used in the simulation programs SEDZL and the branches 3.2.15, 3.2.18, and 3.2.19 of LM3. This outline is taken from Christoffersen and Jonsson (1985). The main point was to calculate, among other quantities, the bottom shear stress from a combined current and wave motion. This section does not address any derivation of the equations used in calculating the shear stress; it is meant to be an outline of the procedure used in the computer programs mentioned above. Anyone interested in the derivations needs to refer to the paper by Christoffersen and Jonsson (1985).

Our goal was to calculate the bottom shear stress due to the combined motion of currents and waves. Bottom shear stress is a force per unit area that is produced by a combination of both currents and waves. This force is responsible for the resuspension of bottom material in Lake Michigan and, in principle, any other major lake. The resuspension of bottom material is an important process because it accounts for the recycling of toxics and other chemicals previously deposited at the bottom of the lake into the water column, where they are likely to have an impact on the water quality of Lake Michigan.

7.1.2 Total Bed Shear Stress

The calculation of bottom shear stress requires the input of various parameters such as the x and y current velocity components, wave heights, wave periods, and wave directions. Consequently, a mathematical expression for bed shear stress should depend on those parameters, and it is expected to be complex. In order to describe the most general expression for bottom shear stress, the following vector expression for total shear stress due to currents and waves is used.

$$\vec{\tau} = \vec{\tau}_w + \vec{\tau}_c \tag{7.1}$$

where:

$\vec{\tau}_w$ = the shear component due to waves ($N \cdot m^{-2}$)[1]
$\vec{\tau}_c$ = the shear component due to currents ($N \cdot m^{-2}$)
$\vec{\tau}$ = total shear stress ($N \cdot m^{-2}$)

Equation 7.1 is also valid at the lake bed giving:

$$\vec{\tau}_b = \vec{\tau}_{wb} + \vec{\tau}_{cb} \tag{7.2}$$

The subscript b stands for bed (lake bottom).

Note that Equations 7.1 and 7.2 do not include a wave-current interaction term. Strictly speaking, these interaction terms contribute to the shear stress but they are usually very small and are being neglected in the sediment transport model. The biggest effect of the interaction terms is to increase the "effective roughness length" or "effective drag coefficient" "felt" by the current horizontal velocity. This increase in "effective drag coefficient" or "effective roughness length" for the horizontal current is being accounted for (see Section 7.1.3.1).

We can give a more explicit expression for these vector expressions by stating that (see Christoffersen and Jonsson, 1985) $\vec{\tau}_w$ and $\vec{\tau}_c$ have the same directions as the wave and current directions, namely $\vec{\tau}_w$ and $\vec{\tau}_c$ have the same direction as $\vec{u}_w$ and $\vec{u}_c$, respectively, where $\vec{u}_w$ is the horizontal wave velocity and $\vec{u}_c$ is the horizontal current velocity. We expect the shear due to waves to be in the direction of the wave propagation and the shear due to currents to be in the direction of the current. Note that the current and wave velocities are the horizontal components because the shear for resuspension acts along the sediment bed; the perpendicular component of the shear with respect to the bed is ignored. Then, $\vec{\tau}_{wb}$ and $\vec{\tau}_{cb}$ are expressed as follows:

$$\vec{\tau}_{wb} \equiv \frac{\vec{k}}{k}\tau_{wb} = \frac{\vec{k}}{k}\tau_{wbm}e^{i(\theta+\phi_b)} \tag{7.3}$$

$$\vec{\tau}_{cb} \equiv \frac{\vec{U}}{U}\tau_{cb} \tag{7.4}$$

where:

$\frac{\vec{k}}{k}$ = ($\cos\alpha$, $\sin\alpha$) is a unit vector in the horizontal direction of wave propagation (dimensionless)

α = the angle between the horizontal direction of wave propagation and the x_1 axis (x_1 and x_2 are the horizontal Cartesian coordinates) (radians)

---

[1]N-Newtons (this is the unit of force in the metric system).

$\frac{\vec{U}}{U}$ = (cos δ, sin δ) is also a unit vector in the direction of the current (horizontal component) (dimensionless). The current is averaged over the water depth

δ = the angle between the horizontal component of the current direction and the x_1 axis (radians)

τ_{wbm} = the maximum bottom shear stress due to waves. The reason for a maximum value is that the bottom shear stress due to waves is oscillatory in nature and so τ_{wbm} is nothing else but the amplitude of $\vec{\tau}_{wb}$ in Equation 7.3 ($N \cdot m^{-2}$)

ϕ_b = the phase lead relative to the wave particle velocity $\vec{u}_w$. The wave particle velocity $\vec{u}_w$ solution is time-harmonic which results in a time-harmonic wave bed shear stress $\vec{\tau}_{wb}$ (radians)

θ = the phase function given by $\theta = \omega_a t - \vec{k} \cdot \vec{x}$ (radians)

ω_a = absolute angular frequency = $\frac{2\pi}{T_a}$

T_a = absolute wave period

As we have seen, it is possible to split the shear stress into a current component and an oscillating wave component. It is also convenient and natural to define a current friction factor f_c and a wave friction factor f_w in the following way:

$$\tau_{cb} \equiv \frac{1}{2} f_c \rho U^2 \tag{7.5}$$

$$\tau_{wbm} \equiv \frac{1}{2} f_w \rho u_{wov}^2 \tag{7.6}$$

where:

ρ = the fluid density. In our case, the fluid is water and its density value is $10^3 \cdot kg/m^3$

U = the average over-depth current velocity. Some caution should be taken when calculating U because the average of the current velocity is not necessarily performed over the entire water column. In nearshore regions, we take the average of the current velocity over all the sigma layers. For intermediate regions, the average is performed over the last three or four sigma layers, while for deep regions, just the lowermost sigma layer value of the current velocity is used ($m \cdot s^{-1}$)

u_{wov} = the bottom orbital velocity ($m \cdot s^{-1}$) which is determined from linear wave theory and is given by:

$$u_{wov} = \frac{\pi H_s}{T_a \sinh(2\pi h_0 / L_d)} \tag{7.7}$$

where:

L_d = the wavelength (m) and is given by:

$$L_d = \frac{g T_a^2}{2\pi} \tanh(2\pi h_0 / L_d) \tag{7.8}$$

where:

H_s = the wave height (m)
T_a = the absolute wave period (s)
h_0 = the average water depth (m)
g = the constant of acceleration due to gravity ($m \cdot s^{-2}$)

Note that Equation 7.8 is a transcendental equation which means in order to solve for L_d, we need to use a numerical method known as Newton-Ralphson, which is frequently employed to find the roots of an algebraic equation. The wave information needed by Equations 7.7 and 7.8 was generated by a wave model adapted to Lake Michigan (Schwab and Beletsky, 1998).

Looking at Equations 7.5 and 7.6, note that the term corresponding to one-half times the fluid density times the velocity squared is actually the kinetic energy per unit volume. Therefore, the friction factors f_c and f_w can be interpreted as the fraction of kinetic energy per unit volume that is being dissipated at the lake bottom.

We are now in position to give an explicit expression for the total bed shear stress (Equation 7.2). By combining Equations 7.3, 7.4, 7.5, 7.6, and 7.2, we have

$$\begin{aligned} \vec{\tau} &= \frac{\vec{U}}{U}\tau_{cb} + \frac{\vec{k}}{k}\tau_{wbm} e^{i(\theta+\phi_b)} \\ &= \tau_{wbm}\left(\sigma\cos\delta + \cos\left(\theta+\phi_b\right)\cos\alpha, \sigma\sin\delta + \cos\left(\theta+\phi_b\right)\sin\alpha\right) \end{aligned} \tag{7.9}$$

where:

$\vec{\tau}$ has units of $N \cdot m^{-2}$

$$\sigma \equiv \frac{\tau_{cb}}{\tau_{wbm}} = \frac{f_c}{f_w}\left(\frac{U}{u_{wov}}\right)^2 \tag{7.10}$$

σ is dimensionless

From Equation 7.9, the maximum (total) bed shear stress is found to be

$$\tau = \frac{1}{2} f_w \rho u_{wov}^2 m \tag{7.11}$$

τ has units of $N \cdot m^{-2}$

with

$$m = \sqrt{1 + \sigma^2 + 2\sigma\left|\cos(\delta - \alpha)\right|} \tag{7.12}$$

m is dimensionless.

Even though the current friction factor f_c is defined in Equation 7.5 for a pure current motion and the wave friction factor is defined in Equation 7.6 for a pure wave motion, they depend on each other for a combined waves and currents motion. Note that in the limit when U approaches zero, σ will also approach zero which then implies that m will approach the value of one and consequently Equation 7.11 reduces to Equation 7.6. Similarly, in the limit that u_{wov} approaches zero, σ approaches infinity, which then implies that $m \approx \sigma$ and consequently Equation 7.11 reduces to Equation 7.5.

7.1.3 Friction Factors

After defining the current friction factor in Equation 7.5 and the wave friction factor in Equation 7.6, we still have to give expressions for these two factors because the equations just mentioned do not tell us how to determine them. The answer comes from hydrodynamics where one solves the momentum equations satisfying horizontal equilibrium (Christoffersen and Jonsson, 1985). The results of these calculations are discussed below.

7.1.3.1 Current Friction Factor

The result for the current friction factor is:

$$\sqrt{\frac{2}{f_c}} = \frac{1}{\kappa}\ln\left(\frac{30h}{e k_N}\right) - \frac{1}{\kappa}\ln\left(\frac{k_A}{k_N}\right) \tag{7.13}$$

where:

κ = this dimensionless constant is known as the von Karman constant and its value is 0.4. This constant is considered one of the most fundamental in turbulence theory. It occurs in the log-law governing the velocity profile adjacent to a boundary in a turbulent boundary layer. In other words, this constant arises in the analysis of the average velocity profile for the flow between two parallel plates driven by a pressure gradient in one direction. The resulting velocity profile equation is logarithmic, and the Karman constant is related to the slope of the logarithmic curve

k_N = this parameter is known as Nikuradse roughness. Its value may vary from one location of the lake to another and is related to the "roughness" of the lake bottom; that is, its value quantifies how rough the bottom of the lake is or how much friction the bottom of the lake presents to the moving fluid (water in our case). Nikuradse roughness is in units of length (meters or centimeters)

h = this parameter represents the water depth, that is, the distance from the lake bottom to the water surface for a given location. Water depth in the lake may vary with the passage of time but in our calculation we use the value measured at the beginning of the study period

e = the natural base of logarithms and whose value is equal to exp(1) = 2.718

k_A = known as the apparent roughness and is given by the following equation:

$$\frac{k_A}{k_N} = 30\frac{\delta_w}{k_N}\exp\left(-\frac{\kappa\delta_w}{\beta k_N}\sqrt{\frac{\sigma}{m}}\right) \tag{7.14}$$

where:

β = the turbulence constant whose value is 0.0747. This constant is proportional to the eddy viscosity in the wave boundary layer

δ_w, which is known as the bottom layer thickness, is given by the following equation:

$$\delta_w = r\frac{\pi}{\sqrt{2}}k_N\sqrt{\frac{u_{wov}T_a\beta}{k_N 2\pi}}\left(\frac{mf_w}{2}\right)^{\frac{1}{4}} \tag{7.15}$$

where r is the dimensionless factor which later will be taken to be equal to $\sqrt{2/\pi}$ = 0.45 in the calculation. The physical interpretation of the apparent roughness is related to the fact that because there are waves in the lake, the current meets a greater bed shear stress than a pure current (no waves) and thus will "feel" a greater roughness than the "geometrical" Nikuradse roughness k_N. The concept of apparent roughness was introduced by Grant and Madsen (1979). On the other hand, the variable δ_w represents a wave boundary layer thickness that is a certain distance with respect to the lake bottom over which the wave energy gets dissipated; it is

a bottom layer where the wave "feels" the lake bottom. Another way of putting it is by saying that outside this bottom boundary layer, the wave energy does not get dissipated which means that $\tau \simeq 0$.

7.1.3.2 Wave Friction Factor

The expression for the wave friction factor f_w that results from the solution of momentum equations is:

$$f_w = \left(\frac{4\sqrt{2}\beta \pi k_N m^{\frac{1}{3}}}{u_{wov} T_a} \right)^{\frac{2}{3}} \tag{7.16}$$

where:

m = is given by Equation 7.12

The other variables have already been discussed.

7.1.4 Calculation Scheme

The calculation scheme is an iterative one in order to determine f_c and f_w, and it proceeds as follows:

1. Compute f_c with $k_A = k_N$ using Equation 7.13. This is the case for a pure current; that is, no waves. In other words, for a pure current motion the apparent roughness k_A is the same as the Nikuradse roughness k_N. When there are no waves, this is expected because the friction experienced by the current does not have a wave component. After equating k_A to k_N, Equation 7.13 reduces to:

$$\sqrt{\frac{2}{f_c}} = \frac{1}{\kappa} \ln\left(\frac{30h}{e k_N} \right) \tag{7.17}$$

   Equation 7.17 can be simplified to:

$$\sqrt{\frac{2}{f_c}} = 2.5 \ln\left(\frac{11\ h}{k_N} \right) \tag{7.18}$$

2. Compute f_w with m = 1; that is, the case where there are no currents. The case m = 1 corresponds to the case when $\sigma = 0$ (see Equations 7.12 and 7.10); that is, when U = 0 which means the currents are negligible and we are left with a pure wave motion. Substituting m = 1 in Equation 7.16, we get:

$$f_w = \left(\frac{4\sqrt{2}\beta\pi k_N}{u_{wov} T_a} \right)^{\frac{2}{3}} \tag{7.19}$$

3. The next step is to iterate through Equations 7.10, 7.12, and 7.16, keeping f_c fixed, until the values of σ, m, and f_w converge within a numerical accuracy of single precision. On most computers, single precision accuracy usually means that the value of a real variable has an accuracy of six or seven significant digits. We determined that five iterations are sufficient to determine the values of σ, m, and f_w with a precision of six significant digits.

4. Compute bottom layer thickness δ_w and apparent roughness k_A from Equations 7.14 and 7.15, respectively with r = 0.450 and β = 0.0747.

5. Compute a new f_c using Equation 7.13. In this case, we do not use the condition $k_A = k_N$ anymore.

6. Repeat steps 2-5 until numerical convergence with single precision accuracy has been obtained for f_c. We determined that five iterations are sufficient to achieve numerical convergence for f_c with a precision of six significant digits.

7. Calculate bottom shear stress from Equation 7.11 because we now know what σ, m, and f_w are. We are assuming that all the units are in the MKS (meter-kilogram-second) system.

7.1.5 Figures

This section provides several examples of the shear stress calculation. Included are figures that show the frequency distribution of shear stress over a period of 60 days using the meteorological data of 1982 and 1994, for several locations in the lake with different Nikuradse roughness k_N values. The shear stress was calculated using the algorithm described in the previous section, and the program used was SEDZL.

A brief explanation is required to understand the axes of the frequency distributions shown in Figures 7.2 to 7.25. The x-axis represents the range of shear stress values. This range is divided into intervals of equal length, with each interval representing the width of the rectangular columns. The y-axis represents the percentage of shear stress values that fall within an interval. For example, if an interval shows a column height of 80%, this means that 80% of the shear stress values fall within that interval.

The frequency distribution of bottom shear stress was calculated for three locations in Lake Michigan. These locations are labeled A, B, and C (Figure 7.1). These locations were chosen to capture the behavior of shear stress at different zones and different water depths in the lake. Location C is over the deepest part of the southern basin with a water depth of 163.5 m; location B is over the deepest part of the northern basin with a water depth of 261.5 m (which is the deepest location in the lake); and location A is located at the Fox River mouth in Green Bay with a water depth of 2.5 m, which happens to be the shallowest cell on Lake Michigan. The calculation was performed for the years 1982 and 1994 and for four different values of

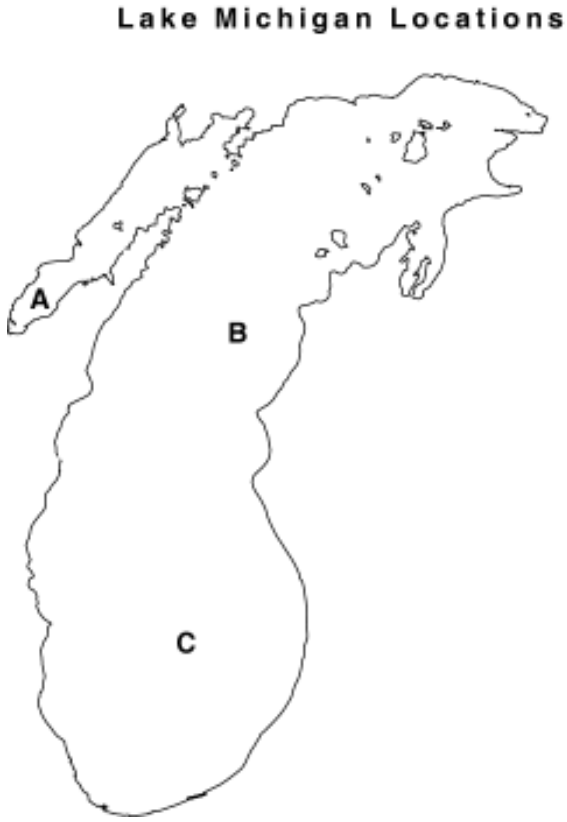


Figure 7.1. Locations for the three selected areas where shear stress was calculated. The locations are denoted by the capital letters A, B, and C.

Nikuradse roughness k_N. The four values that were selected for k_N were 0.2 cm, 1.0 cm, 2.0 cm, and 5.0 cm. One observation to keep in mind is that in SEDZL the units used are in the CGS (centimeter-gram-second) system. Therefore, shear stress units are in dynes/cm^2 and Nikuradse roughness parameters are in centimeters. Please note that Figures 7.2 to 7.13 correspond to a 60-day period starting on March 31, 1982 and Figures 7.14 to 7.25 correspond to a 60-day period starting on January 1, 1994.

One of the main observations made from these figures is that for the two deep locations, B and C, the shear stress distribution does not change much as a function of the Nikuradse roughness parameter. This result is probably not at all surprising.

For water depths of more than 30 m, the contribution of waves to shear is almost negligible, and most of the shear is due to currents. On the other hand, for location A, the change in the shear distribution is very dramatic as k_N varies. In this particular case, we are dealing with a very shallow area of the lake and the effects of both waves and currents are very important. Most of the resuspension of Lake Michigan sediments occurs in shallow waters.

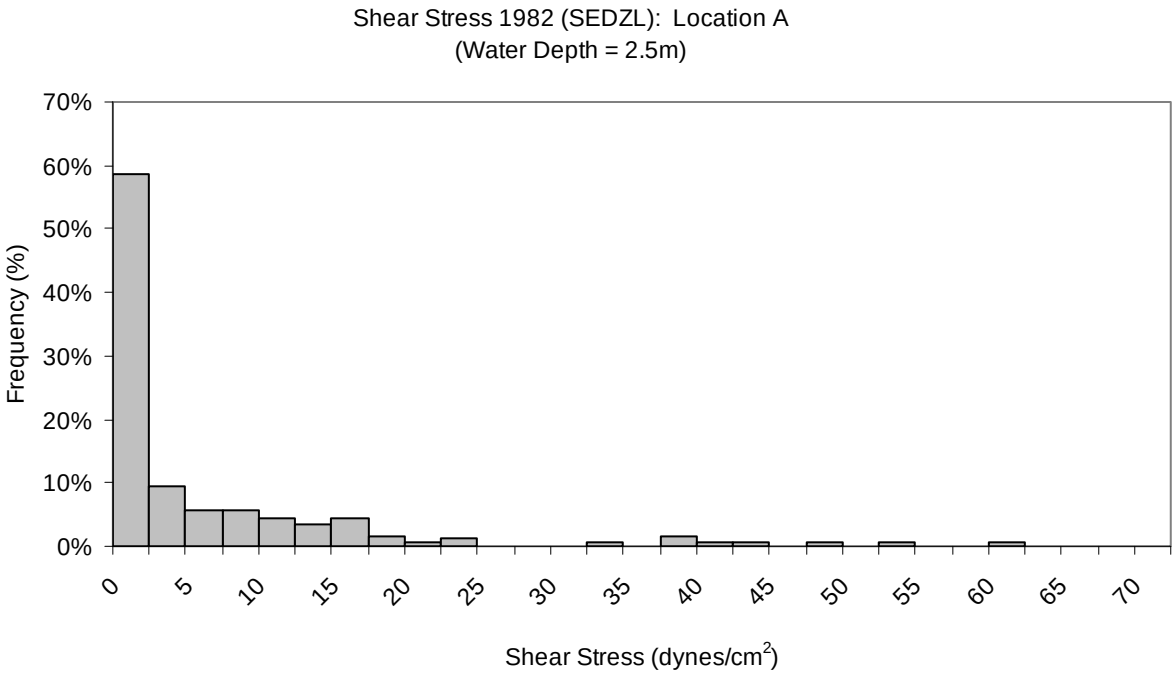


Figure 7.2. Distribution of shear stress for location A, 1982, k_N = 0.2 cm.

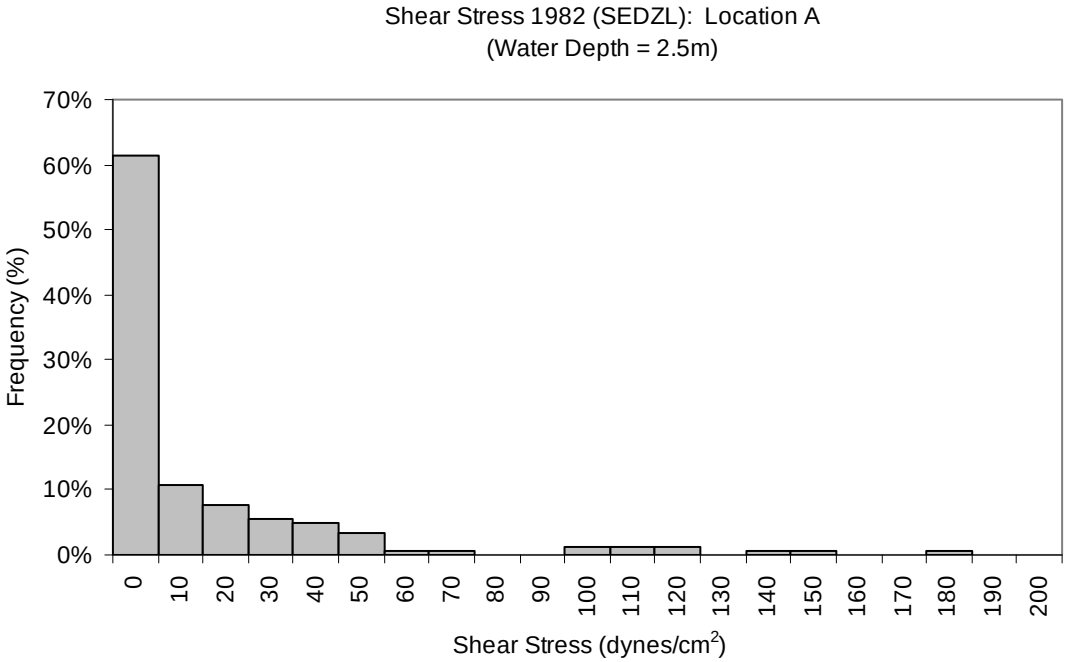


Figure 7.3. Distribution of shear stress for location A, 1982, k_N = 1.0 cm.

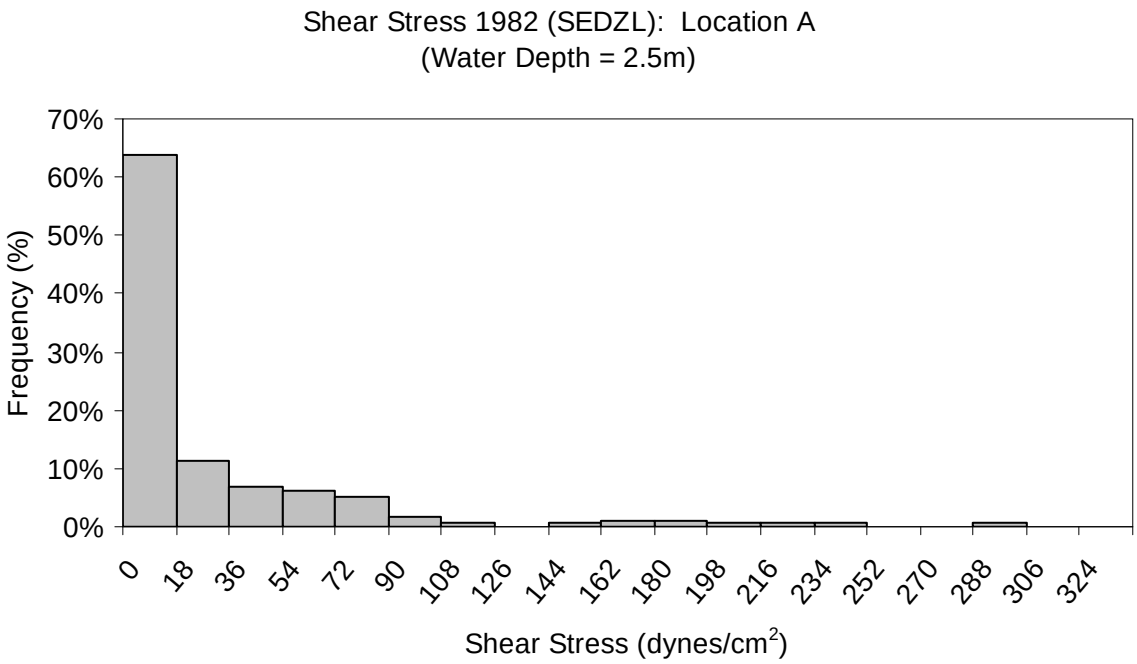


Figure 7.4. Distribution of shear stress for location A, 1982, k_N = 2.0 cm.

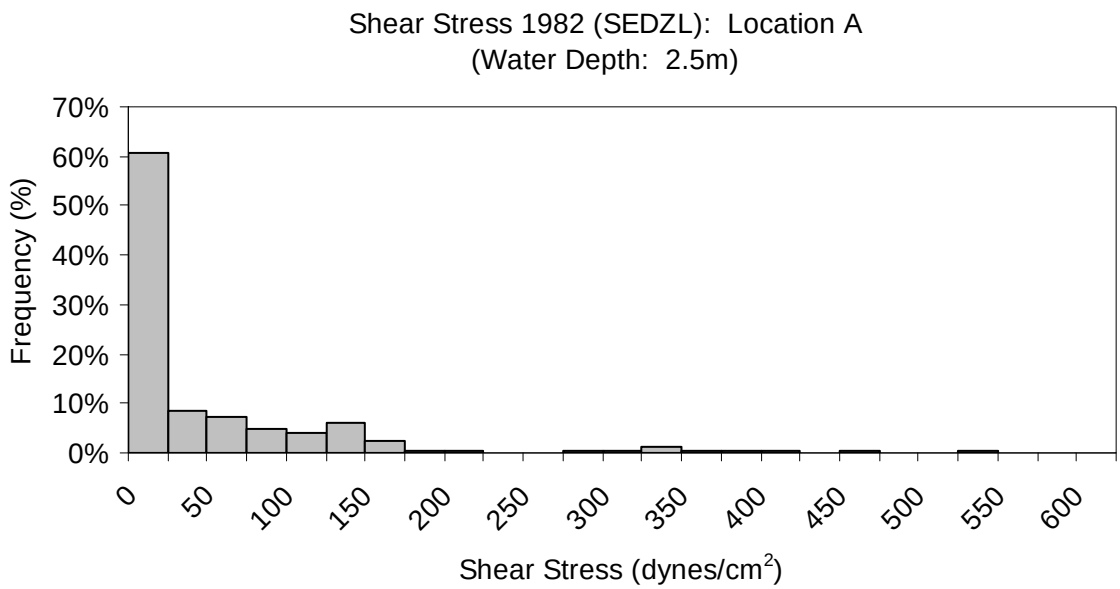


Figure 7.5. Distribution of shear stress for location A, 1982, k_N = 5.0 cm.

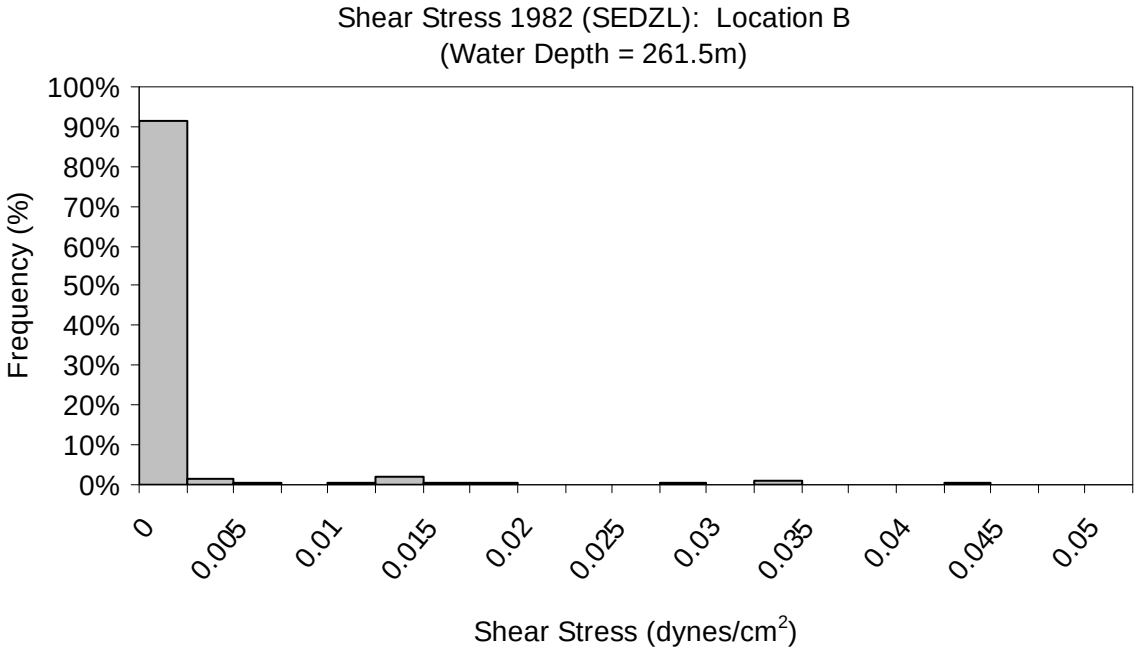


Figure 7.6. Distribution of shear stress for location B, 1982, k_N = 0.2 cm.

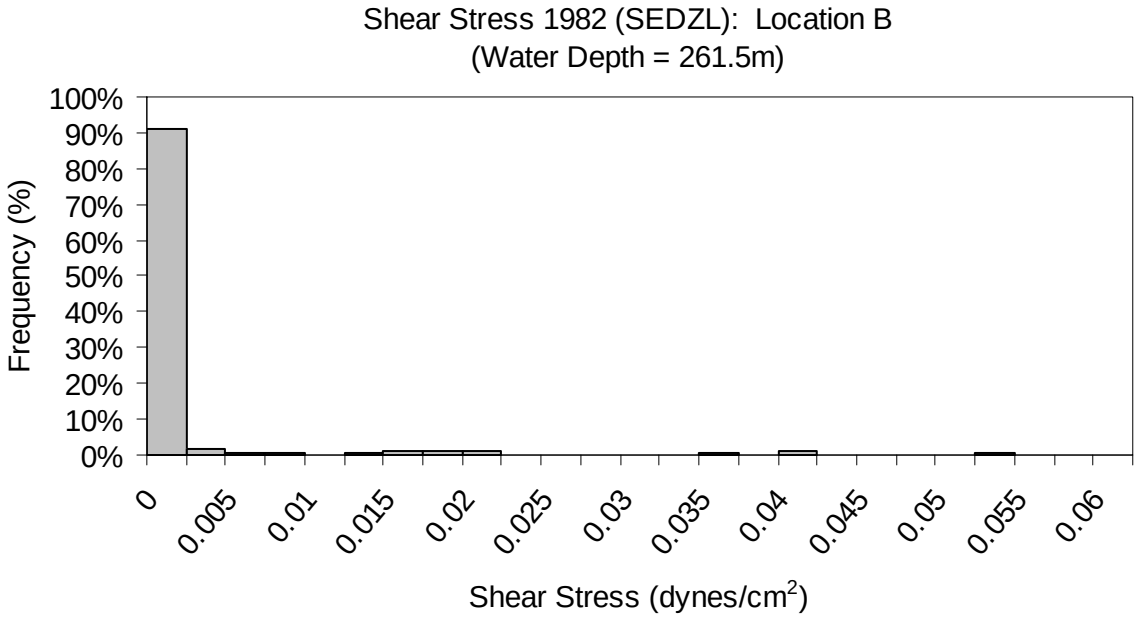


Figure 7.7. Distribution of shear stress for location B, 1982, k_N = 1.0 cm.

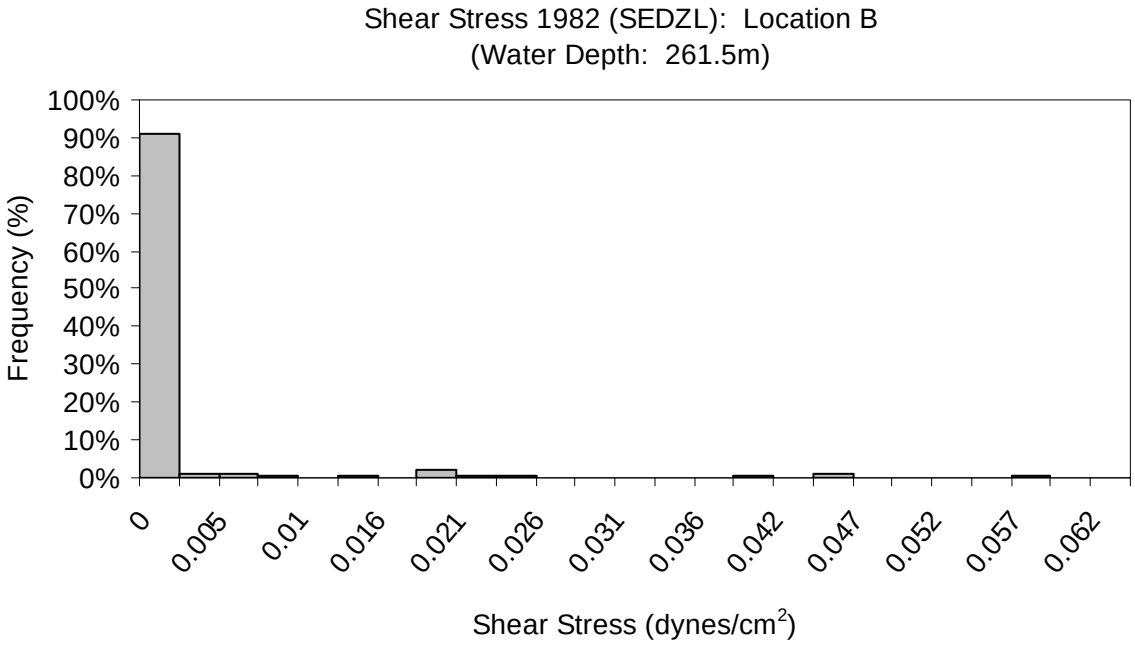


Figure 7.8. Distribution of shear stress for location B, 1982, k_N = 2.0 cm.

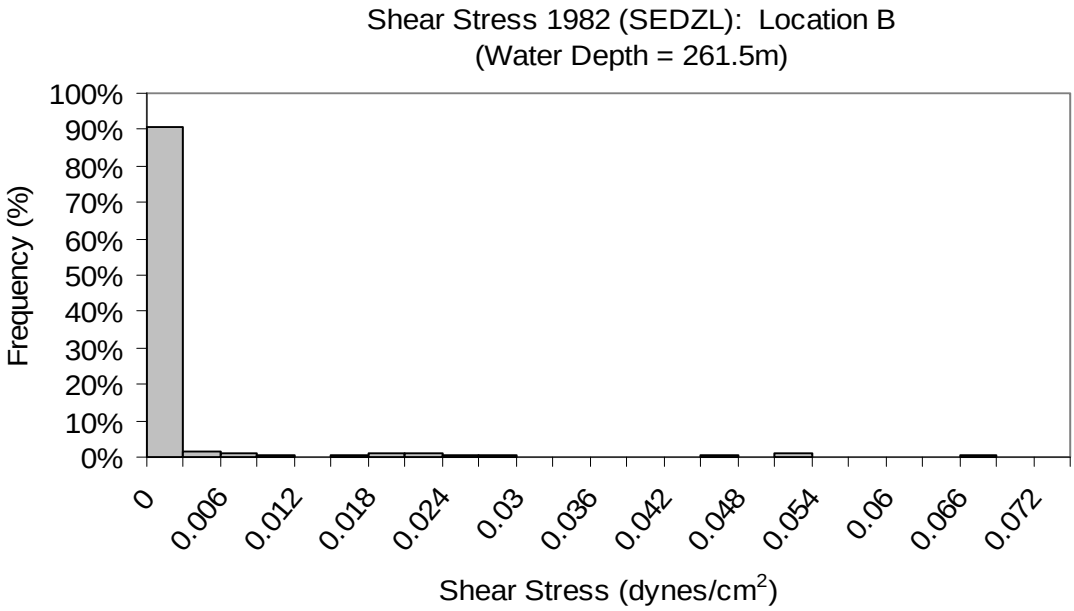


Figure 7.9. Distribution of shear stress for location B, 1982, k_N = 5.0 cm.

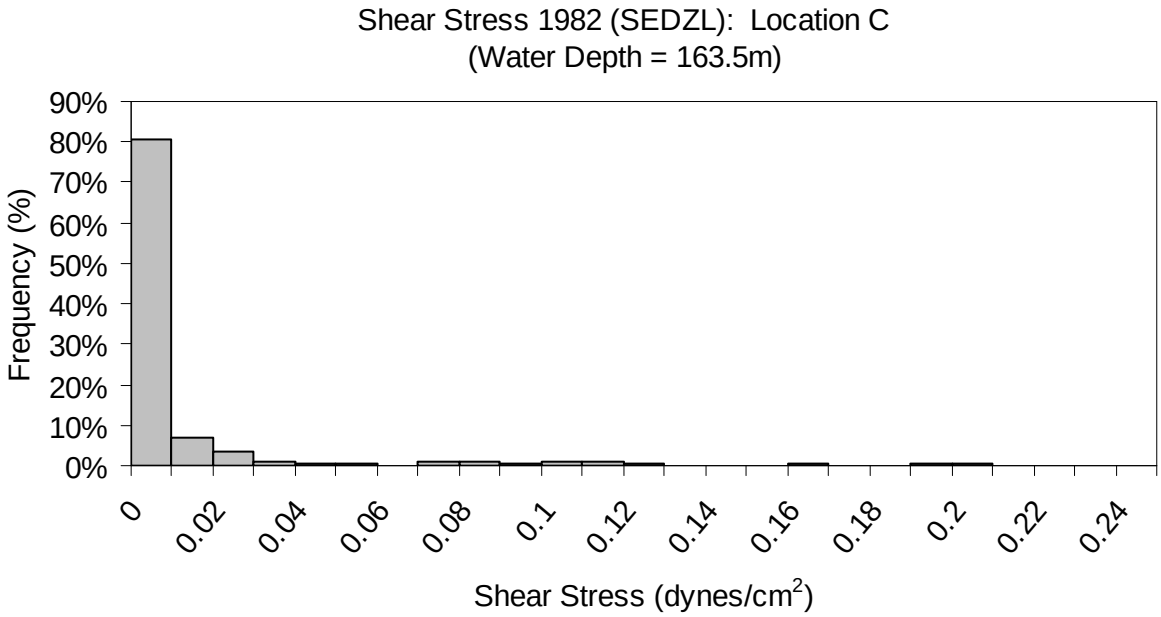


Figure 7.10. Distribution of shear stress for location C, 1982, k_N = 0.2 cm.

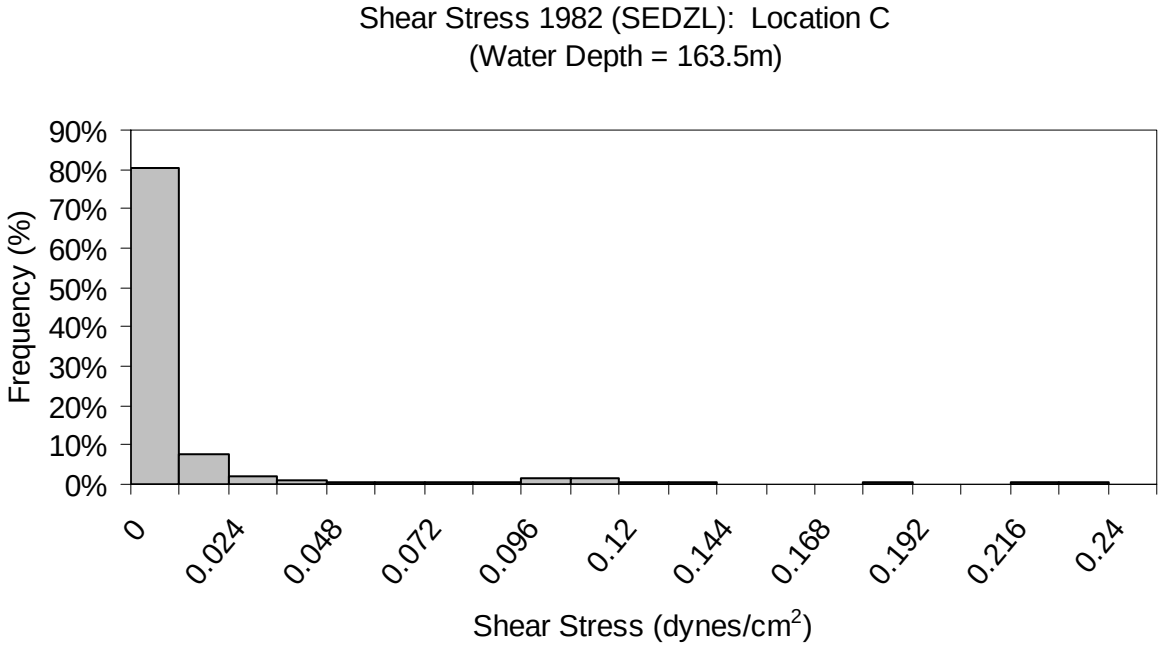


Figure 7.11. Distribution of shear stress for location C, 1982, k_N = 1.0 cm.

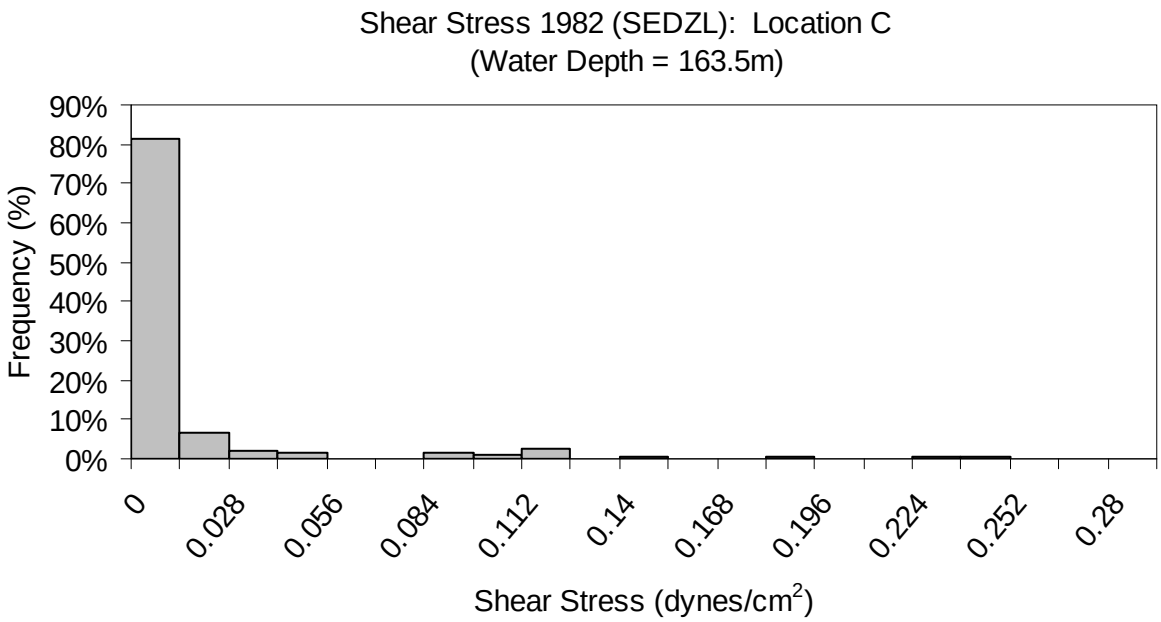


Figure 7.12. Distribution of shear stress for location C, 1982, k_N = 2.0 cm.

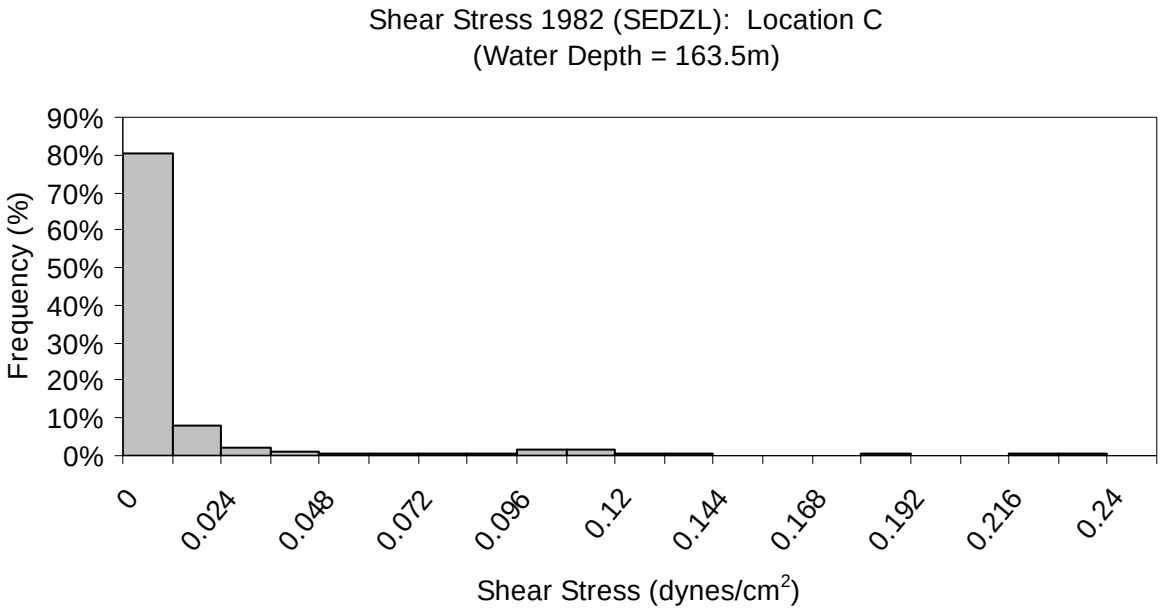


Figure 7.13. Distribution of shear stress for location C, 1982, k_N = 5.0 cm.

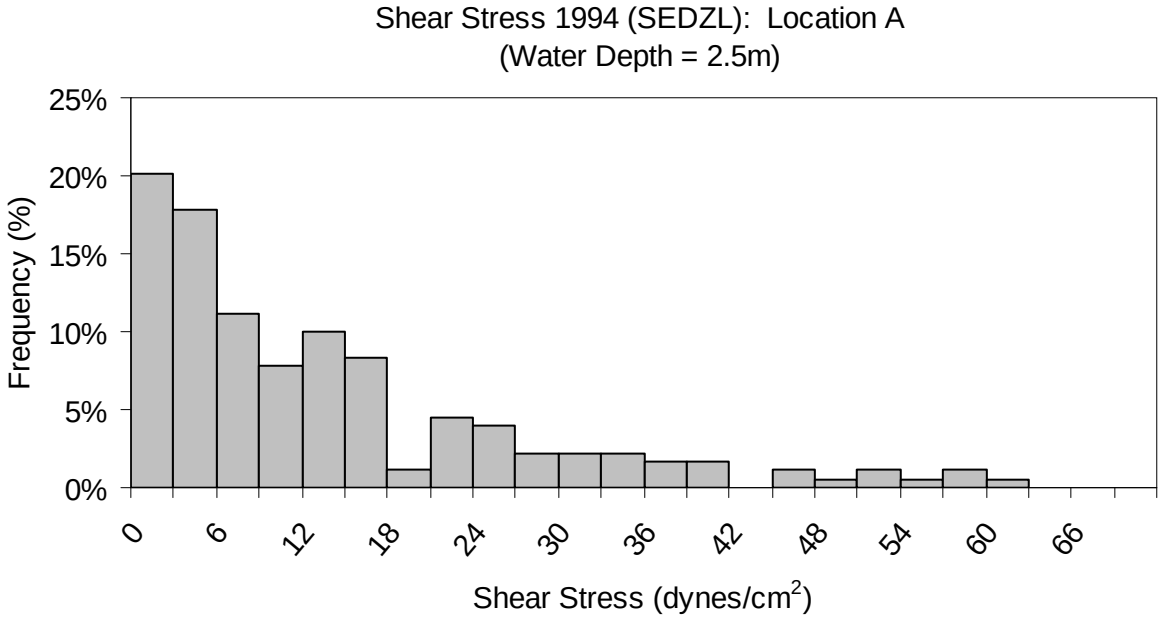


Figure 7.14. Distribution of shear stress for location A, 1994, k_N = 0.2 cm.

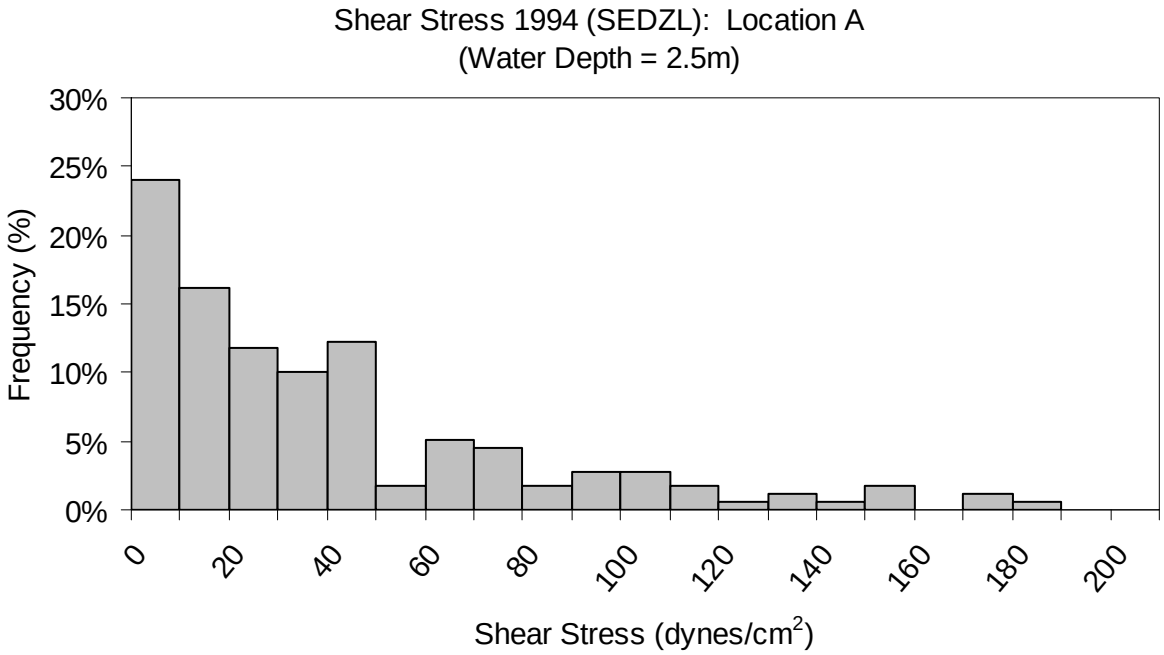


Figure 7.15. Distribution of shear stress for location A, 1994, k_N = 1.0 cm.

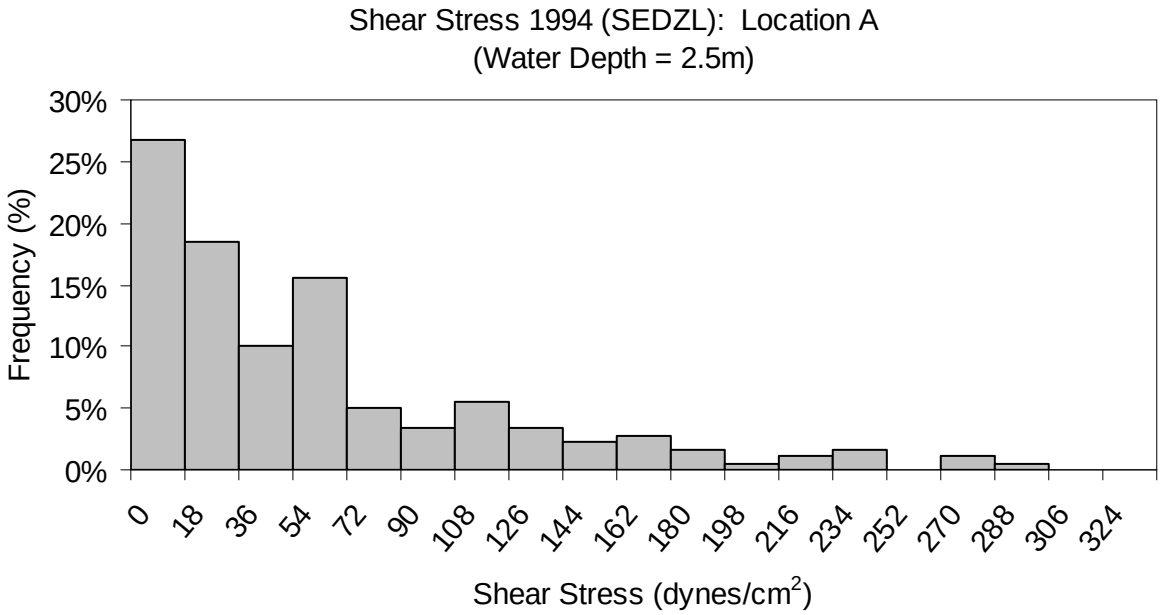


Figure 7.16. Distribution of shear stress for location A, 1994, k_N = 2.0 cm.

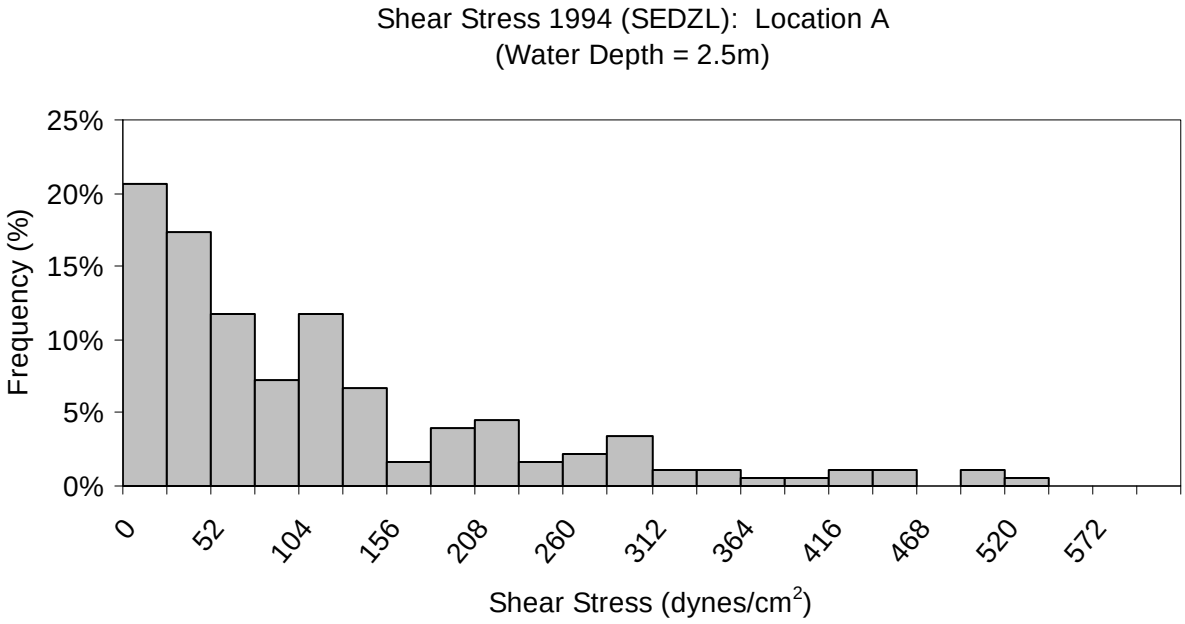


Figure 7.17. Distribution of shear stress for location A, 1994, k_N = 5.0 cm.

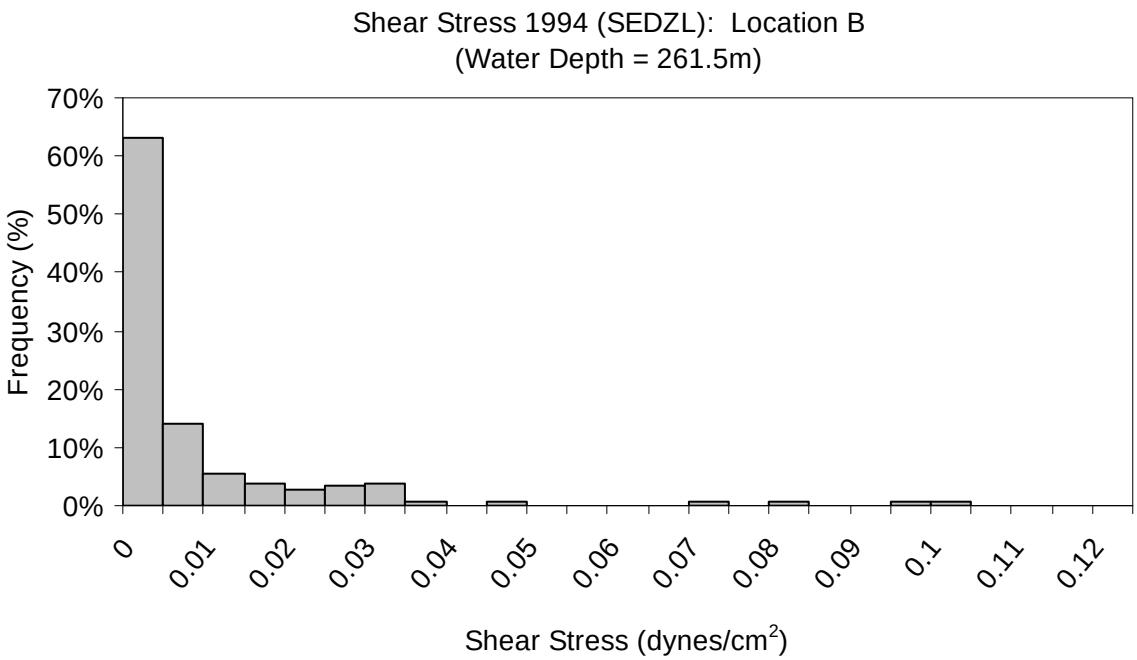


Figure 7.18. Distribution of shear stress for location B, 1994, k_N = 0.2 cm.

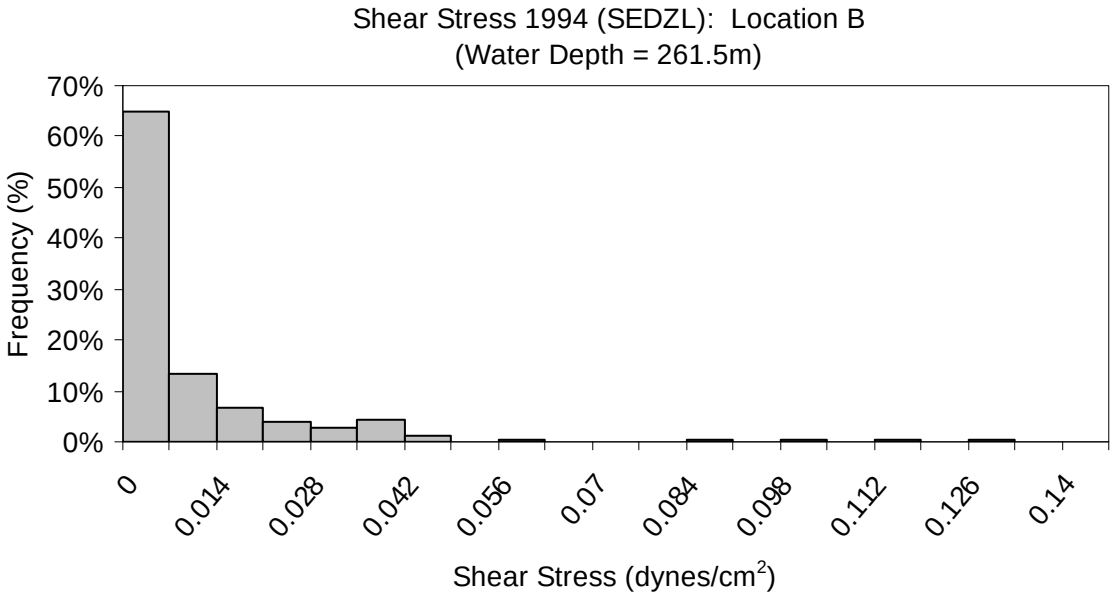


Figure 7.19. Distribution of shear stress for location B, 1994, k_N = 1.0 cm.

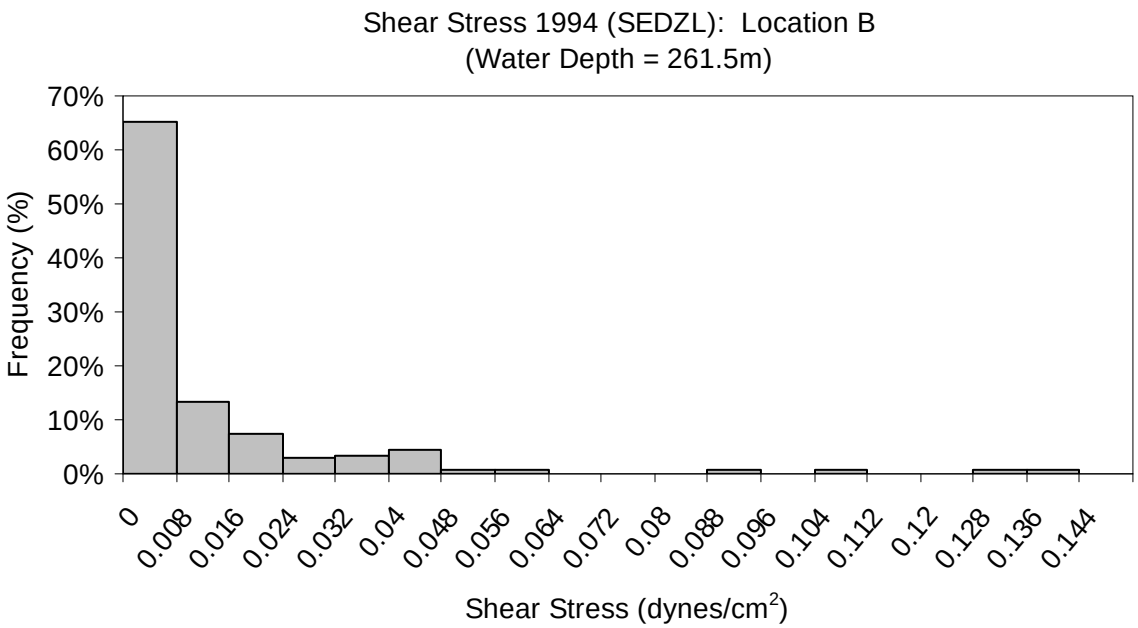


Figure 7.20. Distribution of shear stress for location B, 1994, k_N = 2.0 cm.

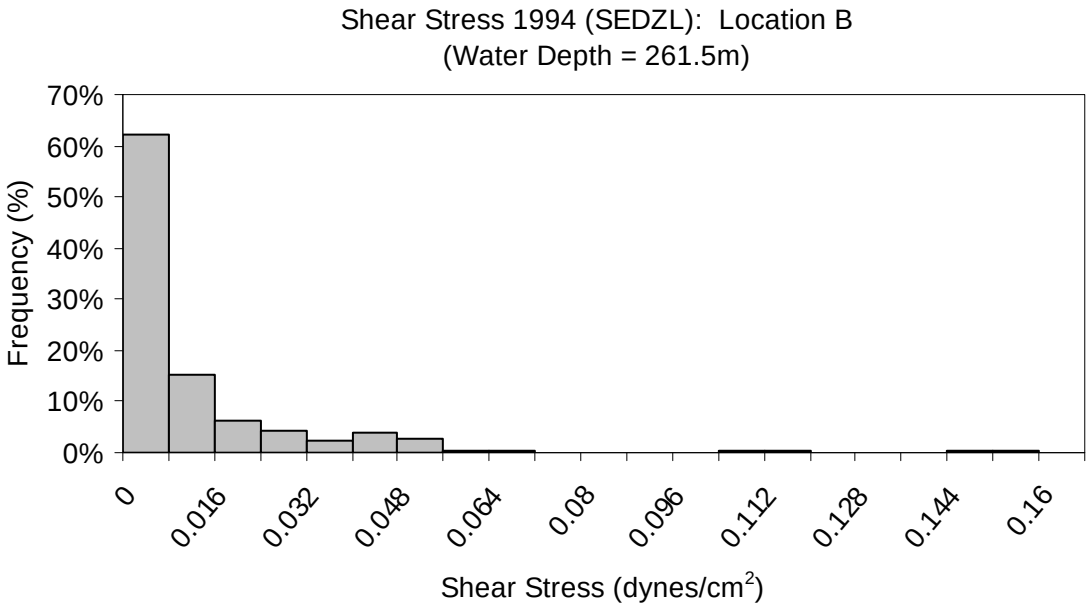


Figure 7.21. Distribution of shear stress for location B, 1994, k_N = 5.0 cm.

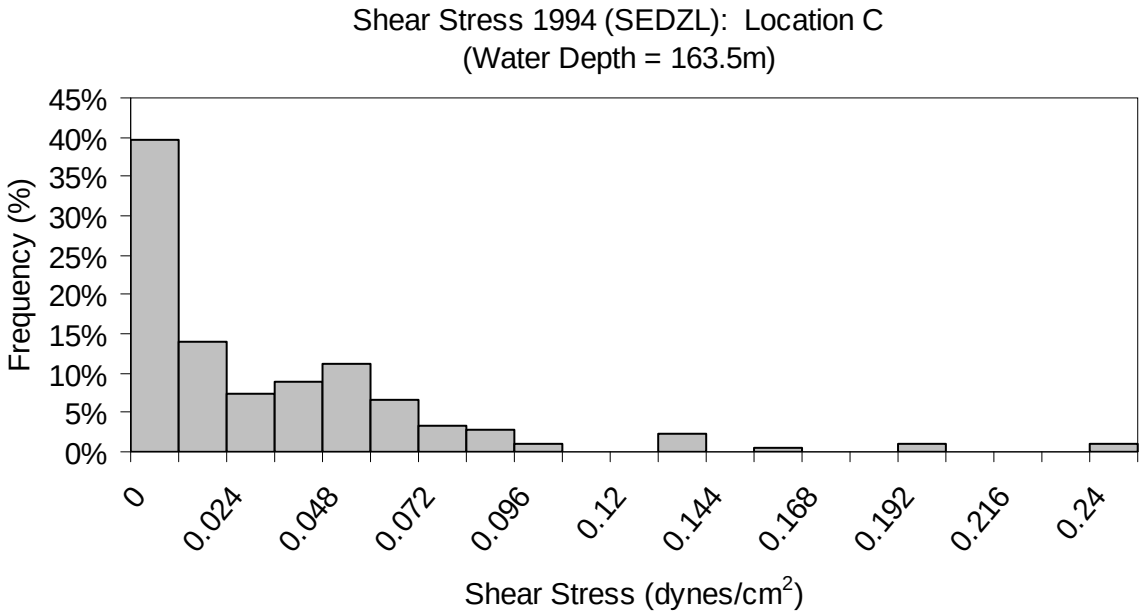


Figure 7.22. Distribution of shear stress for location C, 1994, k_N = 0.2 cm.

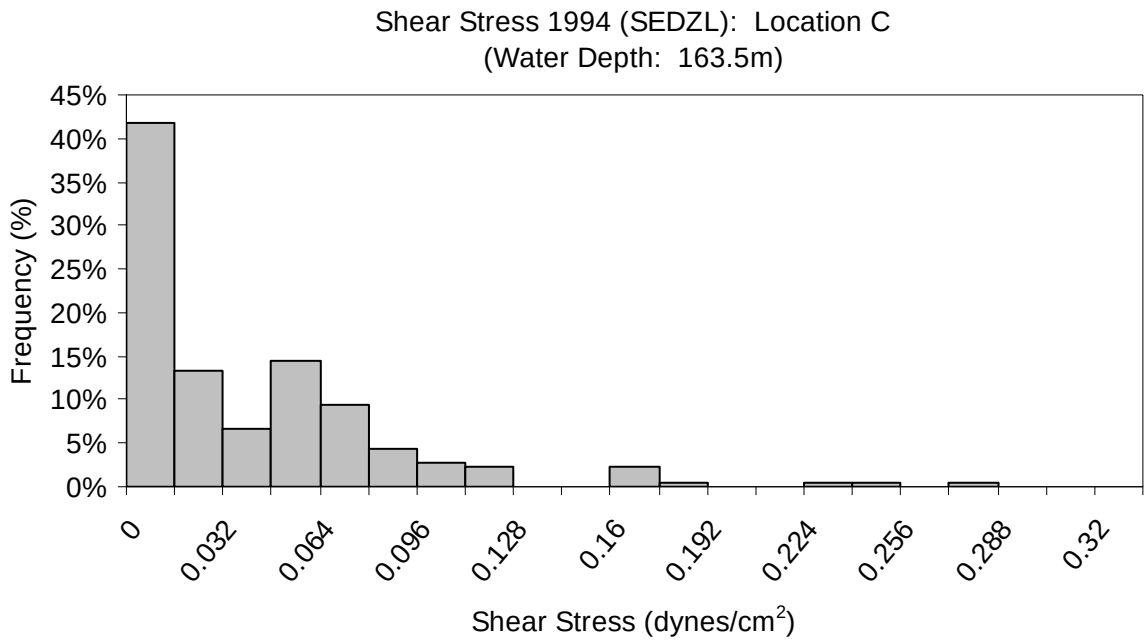


Figure 7.23. Distribution of shear stress for location C, 1994, k_N = 1.0 cm.

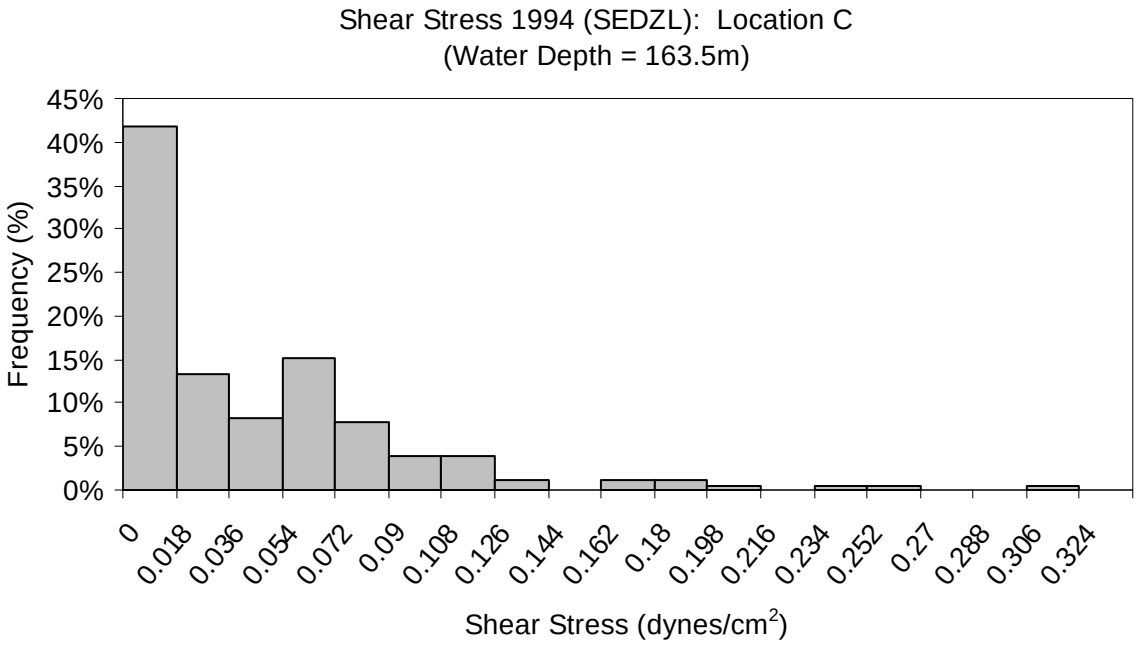


Figure 7.24. Distribution of shear stress for location C, 1994, k_N = 2.0 cm.

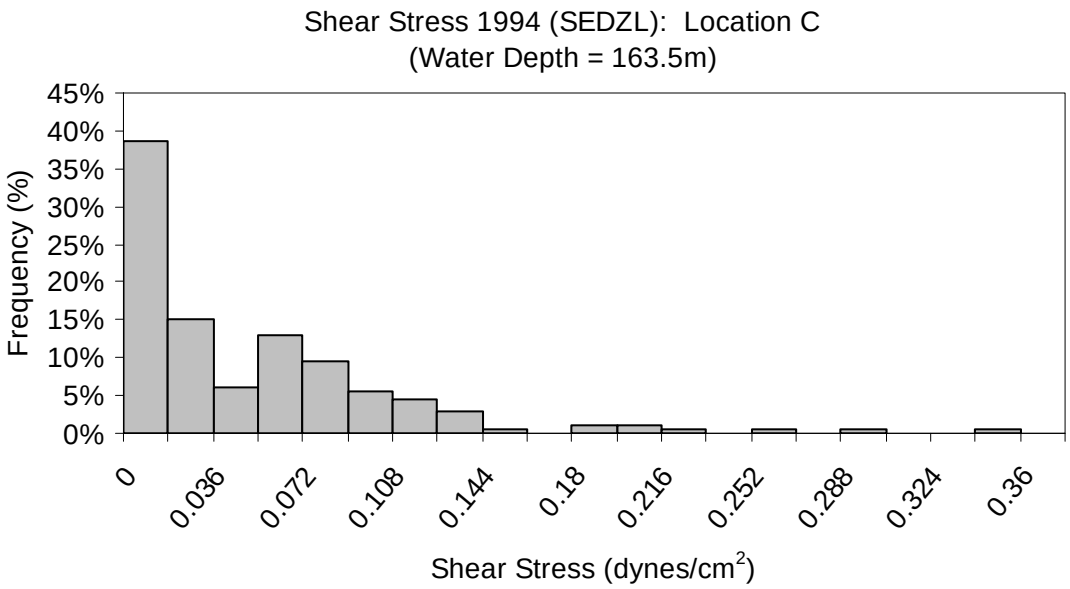


Figure 7.25. Distribution of shear stress for location C, 1994, k_N = 5.0 cm.

Theoretical and experimental work has been conducted to analyze the initiation of sediment movement (quartz particles) for a range of particle diameters and bulk densities (Roberts *et al*., 1998; Lick *et al*., 2004). The behavior of critical shear stress (τ_c) as a function of particle diameter with bulk density as a parameter was investigated. It was shown that τ_c reaches a minimum value around 0.5 dynes · cm^{-2} for the case of the lowest density of 1.65 kg · L^{-1}. For Lake Michigan, in particular, sediment erosion rates were determined at various sites (McNeil and Lick, 2002). These erosion rates were measured as a function of depth and shear stress. The lowest critical shear stress value recorded was 2 dynes · cm^{-2}. Given that for locations B and C shear stress is below 0.5 dynes · cm^{-2}, we can say that for those locations there are no sediment resuspension because shear stress is too low to initiate resuspension.

7.2 PCB Diffusion Process Between Sediment and Water

Diffusive exchange of polychlorinated biphenyls (PCBs) between the sediment and water columns is a process triggered by dissolved PCB concentration gradients between the sediment and the water column. By dissolved PCB concentration, it is meant the purely dissolved and the dissolved organic carbon (DOC)-bound PCBs fractions. In other words, we have two parallel diffusive processes taking place at the same time; one for purely dissolved PCBs and the other one for DOC-bound PCBs.

The discussion that follows only applies to the branches 3.2.15, 3.2.18, and 3.2.19 of LM3. The diffusion process is expressed as a mass flux; that is, mass of PCBs per unit time with a positive diffusive exchange flux representing a contribution of PCB to the water column and a negative diffusive exchange flux representing a contribution of PCB to the sediment bed. The formula that is used in LM3 to describe this sediment-water diffusion process is a purely empirical relationship based on laboratory measurements. The formula is the following:

$$DIFF_{flux} = EXCH_COEFF \bullet (EXCHC_{water} - EXCHC_{sed}) \bullet Area_{cell} \tag{7.20}$$

where:

$DIFF_{flux}$	=	diffusion flux (kg · s^{-1})
EXCH_COEFF	=	exchange coefficient (m · s^{-1})
$EXCHC_{water}$	=	exchange dissolved or DOC-bound PCB water concentration (kg · m^{-3})
$EXCHC_{sed}$	=	exchange dissolved or DOC-bound PCB sediment concentration (kg · m^{-3})
$Area_{cell}$	=	contact surface area between water and sediment 5 km x 5 km overlaying cells (m^2)

The exchange coefficient is given by:

$$EXCH_COEFF = EXCH \bullet D_{water,pcb} \tag{7.21}$$

where:

$$EXCH = \frac{\gamma \bullet \gamma}{TORTUOSITY \bullet LENGTH} \tag{7.22}$$

The term EXCH is actually the segment-specific portion of the exchange coefficient. γ is the porosity value for a given surficial sediment segment. Tortuosity is an indication of how the structure of a porous medium restricts the flow of gases or liquids through the material. In this sense, tortuosity is to fluid flow as the structure of a wire (cross-section, length, etc.) is to the flow of electricity. In our case, the porous medium is the sediment bed, and the fluid is the purely dissolved and DOC-bound PCBs in the water within the sediment bed. The value of tortuosity used for fine-grained sediments is 1.7 (Thibodeaux, 1979). The factor LENGTH is the characteristic length with a value of 0.0085 meters (Thibodeaux, 1979). $D_{water,pcb}$ is the diffusivity of a given PCB congener in water in units of $m^2 \cdot s^{-1}$ and is given by Equation 8.30 (see Section 8.2).

The terms $EXCHC_{water}$ and $EXCHC_{sed}$ are given by

$$EXCHC_{water} = \frac{C_{pcb,water}}{\gamma_{water}} \tag{7.23}$$

$$EXCHC_{sed} = \frac{C_{pcb,sed}}{\gamma_{sed}} \tag{7.24}$$

where:

$C_{pcb,water}$ = purely dissolved or DOC-bound PCB water concentration ($kg \cdot m^{-3}$)
$C_{pcb,sed}$ = purely dissolved or DOC-bound PCB sediment concentration ($kg \cdot m^{-3}$)
γ_{water} = porosity of bottom water segment which is assumed to be one
γ_{sed} = sediment segment-specific porosity

The term $Area_{cell}$ refers to the surface contact area between the bottom 5 km water segment and the surficial sediment segment that is right below the water segment. The grid geometry of the sediment cells is very similar to the ones of the water column such that they line up perfectly; that is, their horizontal cross-section area has the value of 5 km x 5 km which is 2.5 x 10^7 m^2 and is the same as that of the water column.

7.3 LM3 Sediment Bed

The discussion of the LM3 sediment bed construct that follows, in this and subsequent sections, only applies to the branches 3.2.15, 3.2.18, and 3.2.19. The LM3 sediment bed construct is primarily designed to look at PCB transport between the water column and sediment column and, consequently, is a relatively simple one. It has a number of simplifications that facilitate the simulation of a given PCB congener within a reasonable amount of computer time. The main simplifications are:

1. There is no actual computation of resuspension fluxes within the application LM3 itself. Instead, resuspension fluxes were previously calculated within the application known as SEDZL.

2. The initially specified sediment concentration values of the solids are constant throughout the simulation. Only one solid, refractory organic carbon (ROC), is included in the sediment bed as of this writing.

3. Sediment cell volumes are constant throughout the simulation.

4. No nutrients are included.

5. No diagenesis algorithm has been incorporated.

6. Only two carbon state variables: DOC and ROC.

7. PCB partitioning with respect to DOC and ROC is assumed to be the same as in the water column.

8. No diffusion of dissolved states is specified between sediment layers. The only diffusion process allowed is the PCB diffusion between the top sediment layer and the bottom water column layer (see Section 7.2).

9. Only vertical transport is allowed within the sediment bed. The vertical transport algorithm is the ULTIMATE QUICKEST (UQ) which is actually based on the vertical component of the UQ water column transport (see Chapter 5 and Appendix A).

7.3.1 Initial Conditions

Initial conditions for sediment concentrations need to be specified before performing any LM3 simulations. These initial conditions have to be estimated as accurately as possible using all field data available.

Section 11.1.1 has a discussion about the Common Data Format (CDF) rVariables corresponding to the initial sediment concentration values.

7.3.1.1 ROC Sediment Concentrations

Particulate organic carbon (POC) was measured in the top centimeter of the sediment samples. These data were then interpolated using the natural neighbor method (Watson, 1994) to populate each of the surficial 5 km grid cells in the sediment bed; there are 2,318 such grid cells. We assumed that all the POC is ROC, which is not true in the real world; but for the purpose of simulating PCBs in the sediment bed, this assumption was accepted. Sediment cores that were collected during the study period of 1994-1995 were used as a guidance to estimate ROC values for the deeper sediment layers; that is, the ones below the surficial layer. There were only four sediment cores for which particulate detrital carbon (PDC) sediment profiles were available. These cores were collected at the stations LM94-15, LM94-41, LM94-86, and LM94-112. The estimation of POC for the deeper layers was done by following the trends observed in these four cores.

7.3.1.2 Clay Sediment Concentrations

Clay concentration can be inferred from POC and grain size data, which were measured for the top centimeter of the sediment samples. However, as of this writing, no estimation of clay sediment concentrations has been done.

7.3.1.3 PCB Congener Sediment Concentrations

PCBs were measured in the top centimeter of the sediment samples. The data from these samples was then interpolated/extrapolated using the natural neighbor and inverse square distance methods. The natural neighbor method was applied for the open lake while the inverse square distance method was applied for Green Bay. We used the inverse square distance method for Green Bay because of lack of samples collected in that part of Lake Michigan (Xia, 2006).

7.3.1.4 DOC Sediment Concentrations

DOC is treated as a passive state variable in the sediment bed; that is, it does not respond to kinetics, resuspension, burial, diffusion, and settling. A constant concentration value of 17.5 mg $\cdot$ L^{-1} was specified for each of the sediment cells. DOC was used only when calculating the partitioning of PCBs with respect to carbon in the sediment bed.

7.3.2 Sediment Parameters

The sediment bed algorithm that we have in place requires knowledge of several parameters. These parameters are the particle densities, the PCB partition coefficients, and the porosities.

7.3.2.1 Particle Densities

Density of fine-grained sediment is about 2.45 $\pm$ 0.05 g $\cdot$ cm^{-3}, according to John Robbins (Robbins *et al*., 1999).

7.3.2.2 PCB Congener Partition Coefficients

We assumed that the PCB partition coefficients in the sediments were the same as in the water column; the POC and DOC partition coefficients are given in Table 8.1. PCB partition coefficients for clay are equal to zero; in other words, we do not partition any PCBs to clay; the theory is that PCBs bind mostly to organic particles.

7.3.2.3 Porosities

The sediment cell porosities were calculated using the ROC concentrations and the particle densities. The equation used to calculate porosity was:

$$\gamma_{sed} = 1 - \frac{C_{ROC,SED}}{\rho_s}$$

where

ρ_s = particle density = 2.45 g · cm^{-3} = 2450 kg · m^{-3}

Porosities are needed to calculate the diffusive exchange of PCBs between the sediment and water column (Section 7.2).

7.3.3 Boundary Conditions

Boundary conditions are needed for the current sediment bed construct. The boundary conditions are concentration values that are specified for both ROC and PCBs in a infinite mass reservoir located just below the deepest sediment layer. It is assumed that the ROC and PCB concentrations of this infinite mass reservoir are the same as the ones of the deepest sediment layer. This infinite mass reservoir is needed for our current sediment bed model to supply ROC mass to the deepest sediment layer when resuspension is taking place, in order to preserve the initial sediment stack concentration gradient of the solids, in our case ROC. However, when ROC mass is pumped into the sediment stack from below, a given PCB congener mass needs to be pumped into the stack from below as well because of the association of these chemicals with organic solids. When more particles are settling than resuspending, we then push solids down the sediment stack and into this infinite mass reservoir in order to preserve the initial solids concentration gradient.

The algorithm used for the boundary conditions is depicted in Figure 7.26.

7.3.3.1 ROC Sediment Boundary Condition

ROC boundary condition is set to the initial concentration of the deepest sediment layer for each sediment column (sediment stack).

7.3.3.2 PCB Congener Sediment Boundary Condition

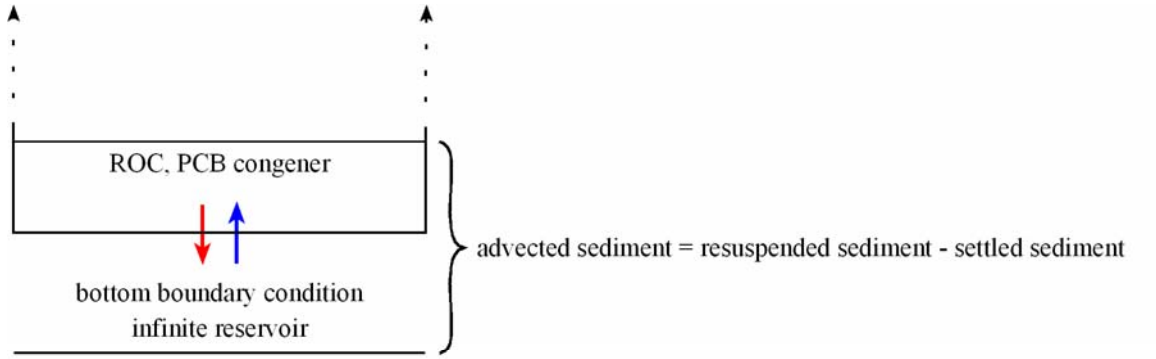
The PCB congener's boundary condition is set to the initial concentration of the deepest sediment layer for each sediment column.

7.3.3.3 Clay Sediment Boundary Condition

As of this writing, no clay sediment concentrations have been estimated and, consequently, no boundary condition has been set yet.

7.3.4 Sediment Transport Data

The LM3 sediment bed needed some basic transport data that was not calculated within the application. One such transport data was the resuspension velocities which were needed to determine how much sediment mass could be resuspended at a given time. The other sediment transport data that one would need are the bulk mixing coefficients, but for our present sediment model, they were not used nor were they estimated for each of the 5 km grid cells.



IF advected sediment < 0.0 THEN

net deposition case

$$\text{boundary deposition velocity} = \frac{\text{advected sediment}}{C_{\text{ROC, bottom layer}} * \text{Area}_{\text{cell}}}$$

$$\text{ROC}_{\text{bottom source}} = C_{\text{ROC, bottom layer}} * \text{Area}_{\text{cell}} * \text{boundary deposition velocity}$$

$$\text{PCB}_{\text{bottom source}} = C_{\text{PCB, bottom layer}} * \text{Area}_{\text{cell}} * \text{boundary deposition velocity}$$

ELSE IF advected sediment > 0.0 THEN

net erosion case

$$C_{\text{ROC, infinite reservoir}} = C_{\text{ROC, bottom layer}}$$

$$\text{boundary deposition velocity} = \frac{\text{advected sediment}}{C_{\text{ROC, infinite reservoir}} * \text{Area}_{\text{cell}}}$$

$$\text{ROC}_{\text{bottom source}} = C_{\text{ROC, infinite reservoir}} * \text{Area}_{\text{cell}} * \text{boundary deposition velocity}$$

$$C_{\text{PCB, infinite reservoir}} = C_{\text{PCB, bottom layer}}$$

$$\text{PCB}_{\text{bottom source}} = C_{\text{PCB, infinite reservoir}} * \text{Area}_{\text{cell}} * \text{boundary deposition velocity}$$

ELSE

no net particle flux case

$$\text{ROC}_{\text{bottom source}} = 0.0$$

$$\text{PCB}_{\text{bottom source}} = 0.0$$

ENDIF

Figure 7.26. Bottom boundary condition algorithm.

7.3.4.1 Resuspension Velocities

The resuspension velocities are obtained through a rather involved process because they are not part of the SEDZL output. The SEDZL output generates resuspension fluxes and not velocities, which means that a conversion from resuspension fluxes to resuspension velocities had to be used.

The units of the resuspension fluxes, as generated by SEDZL, are given in $g \cdot cm^{-2} \cdot s^{-1}$. These units are then converted to $kg \cdot m^{-2} \cdot s^{-1}$. One then needs to divide these fluxes by surficial sediment concentrations to get resuspension velocities in units of $m \cdot s^{-1}$.

However, before performing the conversion from fluxes to velocities, we need to process the fine sediment concentrations, which are interpolated over the SEDZL grid by rectifying the SEDZL grid values to fit the LM3 grid. The SEDZL grid is slightly different from that of LM3. SEDZL has 2,336 surface cells versus 2,318 surface cells in LM3. Moreover, some SEDZL surface grid cells that are considered land cells are considered water cells in LM3 and vice-versa.

SEDZL was originally developed as a tool to study sediment and contaminant transport only. It predates the development of LM3 by about 10 years. The grid employed by SEDZL was devised to approximate the complex geometry of Lake Michigan as close to reality as possible, in particular, the shoreline and boundaries of the lake. When the LM3 grid was devised, no attempts were made in matching it to the SEDZL grid because the original intent was to incorporate a sediment transport scheme, including shear stress calculations, into LM3. However, extensive computer resources were needed to run sediment transport (including shear stress) and eutrophication at the same time, which prevented LLRS from implementing this option. Thus the decision was made to use SEDZL to help in estimating resuspension velocities (*via* fluxes), which are then input into LM3. This decision meant, though, that a matching between the SEDZL and LM3 grids had to be made. Therefore, to handle this grid mismatch between SEDZL and LM3, the procedure below was devised. No statistical analysis is provided in this report to quantify the error incurred in rectifying the SEDZL output to fit the LM3 grid geometry. Nevertheless, the number of adjusted/manipulated grid cells is only 34; only eight LM3 grid cells, of the 2,318 that are over water, are assigned values. The remaining 26 were zeroed out because they are land grid cells in LM3 (see below). Therefore, it is safe to say that any error caused by mapping the SEDZL grid to the LM3 grid is negligible.

Before we proceed, some clarification of the notation used below needs a brief explanation. The array name FINE_SED_CONC stands for fine sediment concentration. The numbers in pairs between parentheses represent a given cell of the Lake Michigan Mass Balance Project (LMMBP) grid. The grid is a rectangular one which consists of 53 cells running in a west-east direction and 102 cells running in a south-north direction.

1. Evaluate the cells that were set to zero in SEDZL but that are over water in LM3 as:

```
FINE_SED_CONC(39,73)  =  FINE_SED_CONC(40,74)
FINE_SED_CONC(39,74)  =  (FINE_SED_CONC(40,74)  +  FINE_SED_CONC(40,75))  /  2.0
FINE_SED_CONC(41,72)  =  (FINE_SED_CONC(41,73)  +  FINE_SED_CONC(42,73))  /  2.0
FINE_SED_CONC(48,87)  =  (FINE_SED_CONC(47,87)  +  FINE_SED_CONC(47,88))  /  2.0
FINE_SED_CONC(48,86)  =  (FINE_SED_CONC(47,86)  +  FINE_SED_CONC(47,87))  /  2.0
```

```
FINE_SED_CONC(49,86)   =   (FINE_SED_CONC(47,86)   +   FINE_SED_CONC(47,87))  /  2.0
FINE_SED_CONC(50,86)   =   (FINE_SED_CONC(47,86)   +   FINE_SED_CONC(47,87))  /  2.0
FINE_SED_CONC(24,96)   =   (FINE_SED_CONC(24,95)   +   FINE_SED_CONC(23,95))  /  2.0
```

2. Zero out cells that are water cells in SEDZL but land cells in LM3.

```
FINE_SED_CONC(4,67)    =   0.0
FINE_SED_CONC(4,68)    =   0.0
FINE_SED_CONC(40,73)   =   0.0
FINE_SED_CONC(41,74)   =   0.0
FINE_SED_CONC(41,75)   =   0.0
FINE_SED_CONC(41,76)   =   0.0
FINE_SED_CONC(31,77)   =   0.0
FINE_SED_CONC(33,79)   =   0.0
FINE_SED_CONC(33,80)   =   0.0
FINE_SED_CONC(12,81)   =   0.0
FINE_SED_CONC(19,84)   =   0.0
FINE_SED_CONC(19,85)   =   0.0
FINE_SED_CONC(36,86)   =   0.0
FINE_SED_CONC(40,90)   =   0.0
FINE_SED_CONC(41,91)   =   0.0
FINE_SED_CONC(41,92)   =   0.0
FINE_SED_CONC(41,93)   =   0.0
FINE_SED_CONC(40,93)   =   0.0
FINE_SED_CONC(38,93)   =   0.0
FINE_SED_CONC(43,94)   =   0.0
FINE_SED_CONC(41,95)   =   0.0
FINE_SED_CONC(23,92)   =   0.0
FINE_SED_CONC(24,93)   =   0.0
FINE_SED_CONC(19,93)   =   0.0
FINE_SED_CONC(19,94)   =   0.0
FINE_SED_CONC(49,94)   =   0.0
```

3. Read in the ROC surficial sediment bed concentrations.

4. Assign a value to cells in LM3 that are over water but that are considered land cells in SEDZL.

```
RESUSPENSION_FLUX(39,73)   =   RESUSPENSION_FLUX(40,74)
RESUSPENSION_FLUX(39,74)   =   (RESUSPENSION_FLUX(40,74)
                           +   RESUSPENSION_FLUX(40,75))   /   2.0
RESUSPENSION_FLUX(41,72)   =   (RESUSPENSION_FLUX(41,73)
                           +   RESUSPENSION_FLUX(42,73))   /   2.0
RESUSPENSION_FLUX(48,87)   =   (RESUSPENSION_FLUX(47,87)
                           +   RESUSPENSION_FLUX(47,88))   /   2.0
RESUSPENSION_FLUX(48,86)   =   (RESUSPENSION_FLUX(47,86)
                           +   RESUSPENSION_FLUX(47,87))   /   2.0
RESUSPENSION_FLUX(49,86)   =   (RESUSPENSION_FLUX(47,86)
                           +   RESUSPENSION_FLUX(47,87))   /   2.0
RESUSPENSION_FLUX(50,86)   =   (RESUSPENSION_FLUX(47,86)
                           +   RESUSPENSION_FLUX(47,87))   /   2.0
RESUSPENSION_FLUX(24,96)   =   (RESUSPENSION_FLUX(24,95)
                           +   RESUSPENSION_FLUX(23,95))   /   2.0
```

5. Zero out cells that are water cells in SEDZL but land cells in LM3.

```
RESUSPENSION_FLUX(4,67)    =   0.0
RESUSPENSOIN_FLUX(4,68)    =   0.0
```

```
RESUSPENSION_FLUX(40,73)    =    0.0
RESUSPENSOIN_FLUX(41,74)    =    0.0
RESUSPENSION_FLUX(41,75)    =    0.0
RESUSPENSOIN_FLUX(41,76)    =    0.0
RESUSPENSION_FLUX(31,77)    =    0.0
RESUSPENSOIN_FLUX(33,79)    =    0.0
RESUSPENSION_FLUX(33,80)    =    0.0
RESUSPENSOIN_FLUX(12,81)    =    0.0
RESUSPENSION_FLUX(19,84)    =    0.0
RESUSPENSOIN_FLUX(19,85)    =    0.0
RESUSPENSION_FLUX(36,86)    =    0.0
RESUSPENSOIN_FLUX(40,90)    =    0.0
RESUSPENSION_FLUX(41,91)    =    0.0
RESUSPENSOIN_FLUX(41,92)    =    0.0
RESUSPENSION_FLUX(41,93)    =    0.0
RESUSPENSOIN_FLUX(40,93)    =    0.0
RESUSPENSION_FLUX(38,93)    =    0.0
RESUSPENSOIN_FLUX(43,94)    =    0.0
RESUSPENSION_FLUX(41,95)    =    0.0
RESUSPENSOIN_FLUX(23,92)    =    0.0
RESUSPENSION_FLUX(24,93)    =    0.0
RESUSPENSOIN_FLUX(19,93)    =    0.0
RESUSPENSION_FLUX(19,94)    =    0.0
RESUSPENSOIN_FLUX(49,94)    =    0.0
```

Once the mapping between the SEDZL and LM3 grids has been completed, one then proceeds to calculate the resuspension velocities for each of the surficial sediment cells. The velocities are determined by dividing the resuspension flux of a given surficial sediment cell by the largest value among the fine-grained sediment and the ROC concentrations of that cell. Choosing the largest value of these two sediment concentrations minimizes the chance of obtaining unrealistically high resuspension velocities. This choice is to a large extent artificial but necessary given the limitations of the current sediment construct. The algorithm used is depicted in Figure 7.27.

7.3.4.2 Bulk Mixing Coefficients

No bulk mixing coefficients have been interpolated onto the 5 km grid for our current sediment bed model as of this writing.

IF (a given cell is over water) THEN

IF resuspension flux > 0.0 THEN

IF $C_{\text{fine sediment}} \geq C_{ROC}$ THEN

$$\text{resuspension velocity} = \frac{\text{resuspension flux}}{C_{\text{fine sediment}}}$$

ELSE

$$\text{resuspension velocity} = \frac{\text{resuspension flux}}{C_{ROC}}$$

END

ELSE

resuspension velocity = 0.0

END

END

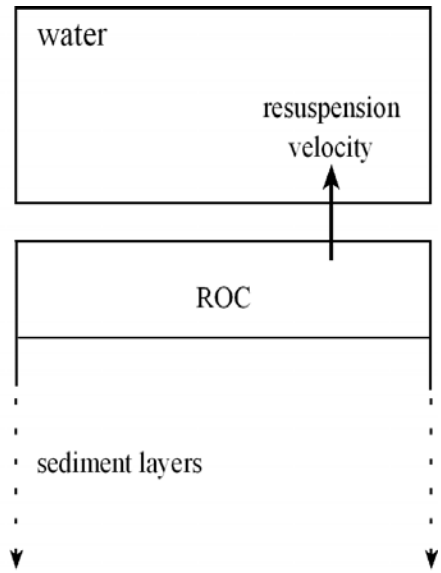


Figure 7.27. Resuspension velocity calculation algorithm.

CHAPTER 8

POLYCHLORINATED BIPHENYL KINETICS

Polychlorinated biphenyls (PCBs) are some of the toxic chemicals of interest to the Lake Michigan Mass Balance Project (LMMBP). Their persistence in the environment, in both water and sediments, coupled with the fact that they bioaccumulate in fish have been a major concern for the public and the government. One of the main concerns of PCBs has been the possibility of toxic effects on humans through the consumption of fish from Lake Michigan. Another major concern has been the effect PCBs might have had on fish reproduction, in particular, that of salmon and lake trout.

PCBs are a group of 209 organic chemicals that share certain chemical characteristics. Each of these chemicals is referred to as a PCB "congener." Even though each one of the PCB congeners has a specific chemical formula, the most common way to refer to each congener is by a number, e.g., PCB 1, PCB 2, and so on. For the purpose of this document, we will refer to each PCB congener by its number.

A total of 54 PCB congeners were modeled for the LMMBP. These accounted for approximately 70% of the total PCB mass in Lake Michigan (Zhang, 2006). For modeling purposes, 36 state variables were considered because some of the PCBs coeluted and could not be separated into their respective individual congeners during sample analysis. When referring to coeluting PCBs, we will be using the "+" sign together with their respective numbers, e.g., PCB 28+31.

8.1 PCB Partitioning

PCBs are hydrophobic which means that when found in the water column or sediments, they are more likely to be sorbed or bound to organic carbon particles. PCBs do occur in a purely dissolved state but their tendency is to bind themselves to organic carbon. This partitioning behavior between organic particles and water has some implications for the transport and fate of PCBs in the water and sediments. Consequently, a reasonable description of partitioning is very important.

PCBs are constantly redistributing themselves between a purely dissolved state and a particulate state. Thus if the concentration of organic particles changes, then the partitioning of PCBs will also change. For instance if the concentration of organic carbon increases, then the PCBs will sorb even more to the carbon particles, increasing their particulate concentration while at the same time reducing their purely dissolved state concentration.

It is assumed that a PCB congener that is bound to particulate organic solids and dissolved organic carbon (DOC) is always in equilibrium with its respective purely dissolved state. In other words, we have the following equilibrium processes taking place:

$$C'_{PCB-POC_j} \leftrightarrow C'_{PCB\text{-}DISS} + C'_{POC_j} \tag{8.1}$$

$$C'_{PCB-DOC} \leftrightarrow C'_{PCB-DISS} + C'_{DOC} \tag{8.2}$$

where:

$C'_{PCB-POC_j}$	=	concentration of PCB congener that is sorbed to the particulate organic carbon (POC) jth state
$C'_{PCB-DOC}$	=	concentration of PCB congener that is sorbed to DOC
$C'_{PCB-DISS}$	=	concentration of purely dissolved PCB congener
C'_{DOC}	=	concentration of DOC
C'_{POC_j}	=	concentration of POC jth state

For the purpose of the discussion in this section, all concentrations depicted with the prime superscript are calculated with respect to water volume like the ones shown above, while concentrations depicted without a prime superscript (see further below in this section) are calculated with respect to segment volume. The relationship between unprimed and primed concentrations is as follows:

$$C_{DOC} = C'_{DOC} \bullet \gamma \tag{8.3}$$

$$C_{POC_j} = C'_{POC_j} \bullet \gamma \tag{8.4}$$

$$C_{PCB-DISS} = C'_{PCB-DISS} \bullet \gamma \tag{8.5}$$

$$C_{PCB-DOC} = C'_{PCB-DOC} \bullet \gamma \tag{8.6}$$

$$C_{PCB-POC_j} = C'_{PCB-POC_j} \bullet \gamma \tag{8.7}$$

$$\gamma = \text{porosity} \equiv \frac{\text{volume of water in segment}}{\text{volume of segment}}$$

$$= \begin{cases} 1 \text{ in the water column} \\ \text{between 0 and 1 in the sediment column} \end{cases} \tag{8.8}$$

Note that for the water column, the porosity is one. This implies that the primed and unprimed concentrations are the same. LM3 concentrations are calculated internally with respect to segment volume for both water and sediment columns. The depiction of primed concentrations in this section is for derivational purposes only.

The phase concentrations of Equations 8.1 and 8.2 are governed by the equilibrium partition coefficients K_{POC_j} and K_{DOC}, where the subscripts POC_j and DOC are POC jth state and DOC, respectively. These partition coefficients are defined as:

$$K_{POC_j} = \frac{C'_{PCB-POC_j}}{C'_{POC_j}\left(C'_{PCB-DISS}\right)} \tag{8.9}$$

$$K_{DOC} = \frac{C'_{PCB-DOC}}{C'_{DOC}\left(C'_{PCB-DISS}\right)} \tag{8.10}$$

The number of POC states in the water column for LM3 version 3.2.15 is five. These particulate organic states are: diatoms, greens, labile organic carbon (LOC), refractor organic carbon (ROC), and herbivorous zooplankton. The number of POC states in the sediment bed is only one, which is represented by ROC. In theory one could argue that we have five POC partition coefficients in the water column plus another one in the sediment bed. However, the assumption in the model is that all the POC partition coefficients have exactly the same value, and that this value remains the same regardless of where the partitioning takes place, that is, in the water or sediment column. The DOC partition coefficient is also assumed to be the same in both the water and sediment column. Thus we are left with only two partition coefficient values for each of the PCB congeners: one POC partition coefficient value and one DOC partition coefficient value.

The complete list of PCB congener partition coefficients and their corresponding input parameter names are given in Table 8.1. This list only includes the PCB congener groups modeled within the LM3 application.

The fractions of a PCB congener for the purely dissolved, POC-bound, and DOC-bound phases are defined as:

$$f_{PCB-DISS} \equiv \frac{C'_{PCB-DISS} \bullet \gamma}{C_T} \tag{8.11}$$

$$f_{PCB-POC_j} \equiv \frac{C_{PCB-POC_j}}{C_T} \tag{8.12}$$

$$f_{PCB-DOC} \equiv \frac{C_{PCB-DOC}}{C_T} \tag{8.13}$$

Table 8.1. PCB Congener Groups Partition Coefficients

Key Name	Description	Value ($L \cdot (kg\ C)^{-1}$)
PART_COEFF_PCB_5+8_DOC	log K_{DOC} of PCB congener 5+8	4.46
PART_COEFF_PCB_5+8_DIA	log K_{DIA} of PCB congener 5+8	6.46
PART_COEFF_PCB_5+8_GRE	log K_{GRE} of PCB congener 5+8	6.46
PART_COEFF_PCB_5+8_ZOO	log K_{ZOO} of PCB congener 5+8	6.46
PART_COEFF_PCB_5+8_LOC	log K_{LOC} of PCB congener 5+8	6.46
PART_COEFF_PCB_5+8_ROC	log K_{ROC} of PCB congener 5+8	6.46
PART_COEFF_PCB_15+17_DOC	log K_{DOC} of PCB congener 15+17	4.20
PART_COEFF_PCB_15+17_DIA	log K_{DIA} of PCB congener 15+17	6.20
PART_COEFF_PCB_15+17_GRE	log K_{GRE} of PCB congener 15+17	6.20
PART_COEFF_PCB_15+17_ZOO	log K_{ZOO} of PCB congener 15+17	6.20
PART_COEFF_PCB_15+17_LOC	log K_{LOC} of PCB congener 15+17	6.20
PART_COEFF_PCB_15+17_ROC	log K_{ROC} of PCB congener 15+17	6.20
PART_COEFF_PCB_16+32_DOC	log K_{DOC} of PCB congener 16+32	3.74
PART_COEFF_PCB_16+32_DIA	log K_{DIA} of PCB congener 16+32	5.89
PART_COEFF_PCB_16+32_GRE	log K_{GRE} of PCB congener 16+32	5.89
PART_COEFF_PCB_16+32_ZOO	log K_{ZOO} of PCB congener 16+32	5.89
PART_COEFF_PCB_16+32_LOC	log K_{LOC} of PCB congener 16+32	5.89
PART_COEFF_PCB_16+32_ROC	log K_{ROC} of PCB congener 16+32	5.89
PART_COEFF_PCB_18_DOC	log K_{DOC} of PCB congener 18	4.00
PART_COEFF_PCB_18_DIA	log K_{DIA} of PCB congener 18	6.00
PART_COEFF_PCB_18_GRE	log K_{GRE} of PCB congener 18	6.00
PART_COEFF_PCB_18_ZOO	log K_{ZOO} of PCB congener 18	6.00
PART_COEFF_PCB_18_LOC	log K_{LOC} of PCB congener 18	6.00
PART_COEFF_PCB_18_ROC	log K_{ROC} of PCB congener 18	6.00
PART_COEFF_PCB_26_DOC	log K_{DOC} of PCB congener 26	4.01
PART_COEFF_PCB_26_DIA	log K_{DIA} of PCB congener 26	6.01
PART_COEFF_PCB_26_GRE	log K_{GRE} of PCB congener 26	6.01
PART_COEFF_PCB_26_ZOO	log K_{ZOO} of PCB congener 26	6.01
PART_COEFF_PCB_26_LOC	log K_{LOC} of PCB congener 26	6.01
PART_COEFF_PCB_26_ROC	log K_{ROC} of PCB congener 26	6.01
PART_COEFF_PCB_28+31_DOC	log K_{DOC} of PCB congener 28+31	4.13
PART_COEFF_PCB_28+31_DIA	log K_{DIA} of PCB congener 28+31	6.13
PART_COEFF_PCB_28+31_GRE	log K_{GRE} of PCB congener 28+31	6.13
PART_COEFF_PCB_28+31_ZOO	log K_{ZOO} of PCB congener 28+31	6.13
PART_COEFF_PCB_28+31_LOC	log K_{LOC} of PCB congener 28+31	6.13
PART_COEFF_PCB_28+31_ROC	log K_{ROC} of PCB congener 28+31	6.13
PART_COEFF_PCB_33_DOC	log K_{DOC} of PCB congener 33	3.74
PART_COEFF_PCB_33_DIA	log K_{DIA} of PCB congener 33	5.74
PART_COEFF_PCB_33_GRE	log K_{GRE} of PCB congener 33	5.74
PART_COEFF_PCB_33_ZOO	log K_{ZOO} of PCB congener 33	5.74
PART_COEFF_PCB_33_LOC	log K_{LOC} of PCB congener 33	5.74
PART_COEFF_PCB_33_ROC	log K_{ROC} of PCB congener 33	5.74
PART_COEFF_PCB_37+42_DOC	log K_{DOC} of PCB congener 37+42	4.18
PART_COEFF_PCB_37+42_DIA	log K_{DIA} of PCB congener 37+42	6.18
PART_COEFF_PCB_37+42_GRE	log K_{GRE} of PCB congener 37+42	6.18
PART_COEFF_PCB_37+42_ZOO	log K_{ZOO} of PCB congener 37+42	6.18
PART_COEFF_PCB_37+42_LOC	log K_{LOC} of PCB congener 37+42	6.18
PART_COEFF_PCB_37+42_ROC	log K_{ROC} of PCB congener 37+42	6.18

Table 8.1. PCB Congener Groups Partition Coefficients (Continued)

Key Name	Description	Value ($L \cdot (kg\ C)^{-1}$)
PART_COEFF_PCB_44_DOC	log K_{DOC} of PCB congener 44	4.04
PART_COEFF_PCB_44_DIA	log K_{DIA} of PCB congener 44	6.04
PART_COEFF_PCB_44_GRE	log K_{GRE} of PCB congener 44	6.04
PART_COEFF_PCB_44_ZOO	log K_{ZOO} of PCB congener 44	6.04
PART_COEFF_PCB_44_LOC	log K_{LOC} of PCB congener 44	6.04
PART_COEFF_PCB_44_ROC	log K_{ROC} of PCB congener 44	6.04
PART_COEFF_PCB_49_DOC	log K_{DOC} of PCB congener 49	4.28
PART_COEFF_PCB_49_DIA	log K_{DIA} of PCB congener 49	6.28
PART_COEFF_PCB_49_GRE	log K_{GRE} of PCB congener 49	6.28
PART_COEFF_PCB_49_ZOO	log K_{ZOO} of PCB congener 49	6.28
PART_COEFF_PCB_49_LOC	log K_{LOC} of PCB congener 49	6.28
PART_COEFF_PCB_49_ROC	log K_{ROC} of PCB congener 49	6.28
PART_COEFF_PCB_52_DOC	log K_{DOC} of PCB congener 52	4.16
PART_COEFF_PCB_52_DIA	log K_{DIA} of PCB congener 52	6.16
PART_COEFF_PCB_52_GRE	log K_{GRE} of PCB congener 52	6.16
PART_COEFF_PCB_52_ZOO	log K_{ZOO} of PCB congener 52	6.16
PART_COEFF_PCB_52_LOC	log K_{LOC} of PCB congener 52	6.16
PART_COEFF_PCB_52_ROC	log K_{ROC} of PCB congener 52	6.16
PART_COEFF_PCB_56+60_DOC	log K_{DOC} of PCB congener 56+60	4.59
PART_COEFF_PCB_56+60_DIA	log K_{DIA} of PCB congener 56+60	6.59
PART_COEFF_PCB_56+60_GRE	log K_{GRE} of PCB congener 56+60	6.59
PART_COEFF_PCB_56+60_ZOO	log K_{ZOO} of PCB congener 56+60	6.59
PART_COEFF_PCB_56+60_LOC	log K_{LOC} of PCB congener 56+60	6.59
PART_COEFF_PCB_56+60_ROC	log K_{ROC} of PCB congener 56+60	6.59
PART_COEFF_PCB_66_DOC	log K_{DOC} of PCB congener 66	4.70
PART_COEFF_PCB_66_DIA	log K_{DIA} of PCB congener 66	6.70
PART_COEFF_PCB_66_GRE	log K_{GRE} of PCB congener 66	6.70
PART_COEFF_PCB_66_ZOO	log K_{ZOO} of PCB congener 66	6.70
PART_COEFF_PCB_66_LOC	log K_{LOC} of PCB congener 66	6.70
PART_COEFF_PCB_66_ROC	log K_{ROC} of PCB congener 66	6.70
PART_COEFF_PCB_70+76_DOC	log K_{DOC} of PCB congener 70+76	4.49
PART_COEFF_PCB_70+76_DIA	log K_{DIA} of PCB congener 70+76	6.49
PART_COEFF_PCB_70+76_GRE	log K_{GRE} of PCB congener 70+76	6.49
PART_COEFF_PCB_70+76_ZOO	log K_{ZOO} of PCB congener 70+76	6.49
PART_COEFF_PCB_70+76_LOC	log K_{LOC} of PCB congener 70+76	6.49
PART_COEFF_PCB_70+76_ROC	log K_{ROC} of PCB congener 70+76	6.49
PART_COEFF_PCB_74_DOC	log K_{DOC} of PCB congener 74	4.54
PART_COEFF_PCB_74_DIA	log K_{DIA} of PCB congener 74	6.54
PART_COEFF_PCB_74_GRE	log K_{GRE} of PCB congener 74	6.54
PART_COEFF_PCB_74_ZOO	log K_{ZOO} of PCB congener 74	6.54
PART_COEFF_PCB_74_LOC	log K_{LOC} of PCB congener 74	6.54
PART_COEFF_PCB_74_ROC	log K_{ROC} of PCB congener 74	6.54
PART_COEFF_PCB_77+110_DOC	log K_{DOC} of PCB congener 77+110	4.58
PART_COEFF_PCB_77+110_DIA	log K_{DIA} of PCB congener 77+110	6.58
PART_COEFF_PCB_77+110_GRE	log K_{GRE} of PCB congener 77+110	6.58
PART_COEFF_PCB_77+110_ZOO	log K_{ZOO} of PCB congener 77+110	6.58
PART_COEFF_PCB_77+110_LOC	log K_{LOC} of PCB congener 77+110	6.58
PART_COEFF_PCB_77+110_ROC	log K_{ROC} of PCB congener 77+110	6.58

Table 8.1. PCB Congener Groups Partition Coefficients (Continued)

Key Name	Description	Value ($L \cdot (kg\ C)^{-1}$)
PART_COEFF_PCB_81_DOC	log K_{DOC} of PCB congener 81	4.53
PART_COEFF_PCB_81_DIA	log K_{DIA} of PCB congener 81	6.53
PART_COEFF_PCB_81_GRE	log K_{GRE} of PCB congener 81	6.53
PART_COEFF_PCB_81_ZOO	log K_{ZOO} of PCB congener 81	6.53
PART_COEFF_PCB_81_LOC	log K_{LOC} of PCB congener 81	6.53
PART_COEFF_PCB_81_ROC	log K_{ROC} of PCB congener 81	6.53
PART_COEFF_PCB_84+92_DOC	log K_{DOC} of PCB congener 84+92	4.19
PART_COEFF_PCB_84+92_DIA	log K_{DIA} of PCB congener 84+92	6.19
PART_COEFF_PCB_84+92_GRE	log K_{GRE} of PCB congener 84+92	6.19
PART_COEFF_PCB_84+92_ZOO	log K_{ZOO} of PCB congener 84+92	6.19
PART_COEFF_PCB_84+92_LOC	log K_{LOC} of PCB congener 84+92	6.19
PART_COEFF_PCB_84+92_ROC	log K_{ROC} of PCB congener 84+92	6.19
PART_COEFF_PCB_85_DOC	log K_{DOC} of PCB congener 85	4.76
PART_COEFF_PCB_85_DIA	log K_{DIA} of PCB congener 85	6.76
PART_COEFF_PCB_85_GRE	log K_{GRE} of PCB congener 85	6.76
PART_COEFF_PCB_85_ZOO	log K_{ZOO} of PCB congener 85	6.76
PART_COEFF_PCB_85_LOC	log K_{LOC} of PCB congener 85	6.76
PART_COEFF_PCB_85_ROC	log K_{ROC} of PCB congener 85	6.76
PART_COEFF_PCB_87_DOC	log K_{DOC} of PCB congener 87	4.23
PART_COEFF_PCB_87_DIA	log K_{DIA} of PCB congener 87	6.23
PART_COEFF_PCB_87_GRE	log K_{GRE} of PCB congener 87	6.23
PART_COEFF_PCB_87_ZOO	log K_{ZOO} of PCB congener 87	6.23
PART_COEFF_PCB_87_LOC	log K_{LOC} of PCB congener 87	6.23
PART_COEFF_PCB_87_ROC	log K_{ROC} of PCB congener 87	6.23
PART_COEFF_PCB_89_DOC	log K_{DOC} of PCB congener 89	4.07
PART_COEFF_PCB_89_DIA	log K_{DIA} of PCB congener 89	6.07
PART_COEFF_PCB_89_GRE	log K_{GRE} of PCB congener 89	6.07
PART_COEFF_PCB_89_ZOO	log K_{ZOO} of PCB congener 89	6.07
PART_COEFF_PCB_89_LOC	log K_{LOC} of PCB congener 89	6.07
PART_COEFF_PCB_89_ROC	log K_{ROC} of PCB congener 89	6.07
PART_COEFF_PCB_99_DOC	log K_{DOC} of PCB congener 99	4.46
PART_COEFF_PCB_99_DIA	log K_{DIA} of PCB congener 99	6.46
PART_COEFF_PCB_99_GRE	log K_{GRE} of PCB congener 99	6.46
PART_COEFF_PCB_99_ZOO	log K_{ZOO} of PCB congener 99	6.46
PART_COEFF_PCB_99_LOC	log K_{LOC} of PCB congener 99	6.46
PART_COEFF_PCB_99_ROC	log K_{ROC} of PCB congener 99	6.46
PART_COEFF_PCB_101_DOC	log K_{DOC} of PCB congener 101	4.48
PART_COEFF_PCB_101_DIA	log K_{DIA} of PCB congener 101	6.48
PART_COEFF_PCB_101_GRE	log K_{GRE} of PCB congener 101	6.48
PART_COEFF_PCB_101_ZOO	log K_{ZOO} of PCB congener 101	6.48
PART_COEFF_PCB_101_LOC	log K_{LOC} of PCB congener 101	6.48
PART_COEFF_PCB_101_ROC	log K_{ROC} of PCB congener 101	6.48
PART_COEFF_PCB_105+132+153_DOC	log K_{DOC} of PCB congener 105+132+153	5.04
PART_COEFF_PCB_105+132+153_DIA	log K_{DIA} of PCB congener 105+132+153	7.04
PART_COEFF_PCB_105+132+153_GRE	log K_{GRE} of PCB congener 105+132+153	7.04
PART_COEFF_PCB_105+132+153_ZOO	log K_{ZOO} of PCB congener 105+132+153	7.04
PART_COEFF_PCB_105+132+153_LOC	log K_{LOC} of PCB congener 105+132+153	7.04
PART_COEFF_PCB_105+132+153_ROC	log K_{ROC} of PCB congener 105+132+153	7.04

Table 8.1. PCB Congener Groups Partition Coefficients (Continued)

Key Name	Description	Value ($L \cdot (kg\ C)^{-1}$)
PART_COEFF_PCB_118_DOC	log K_{DOC} of PCB congener 118	4.65
PART_COEFF_PCB_118_DIA	log K_{DIA} of PCB congener 118	6.65
PART_COEFF_PCB_118_GRE	log K_{GRE} of PCB congener 118	6.65
PART_COEFF_PCB_118_ZOO	log K_{ZOO} of PCB congener 118	6.65
PART_COEFF_PCB_118_LOC	log K_{LOC} of PCB congener 118	6.65
PART_COEFF_PCB_118_ROC	log K_{ROC} of PCB congener 118	6.65
PART_COEFF_PCB_123+149_DOC	log K_{DOC} of PCB congener 123+149	4.73
PART_COEFF_PCB_123+149_DIA	log K_{DIA} of PCB congener 123+149	6.73
PART_COEFF_PCB_123+149_GRE	log K_{GRE} of PCB congener 123+149	6.73
PART_COEFF_PCB_123+149_ZOO	log K_{ZOO} of PCB congener 123+149	6.73
PART_COEFF_PCB_123+149_LOC	log K_{LOC} of PCB congener 123+149	6.73
PART_COEFF_PCB_123+149_ROC	log K_{ROC} of PCB congener 123+149	6.73
PART_COEFF_PCB_138+163_DOC	log K_{DOC} of PCB congener 138+163	4.68
PART_COEFF_PCB_138+163_DIA	log K_{DIA} of PCB congener 138+163	6.68
PART_COEFF_PCB_138+163_GRE	log K_{GRE} of PCB congener 138+163	6.68
PART_COEFF_PCB_138+163_ZOO	log K_{ZOO} of PCB congener 138+163	6.68
PART_COEFF_PCB_138+163_LOC	log K_{LOC} of PCB congener 138+163	6.68
PART_COEFF_PCB_138+163_ROC	log K_{ROC} of PCB congener 138+163	6.68
PART_COEFF_PCB_146_DOC	log K_{DOC} of PCB congener 146	4.86
PART_COEFF_PCB_146_DIA	log K_{DIA} of PCB congener 146	6.86
PART_COEFF_PCB_146_GRE	log K_{GRE} of PCB congener 146	6.86
PART_COEFF_PCB_146_ZOO	log K_{ZOO} of PCB congener 146	6.86
PART_COEFF_PCB_146_LOC	log K_{LOC} of PCB congener 146	6.86
PART_COEFF_PCB_146_ROC	log K_{ROC} of PCB congener 146	6.86
PART_COEFF_PCB_151_DOC	log K_{DOC} of PCB congener 151	4.75
PART_COEFF_PCB_151_DIA	log K_{DIA} of PCB congener 151	6.75
PART_COEFF_PCB_151_GRE	log K_{GRE} of PCB congener 151	6.75
PART_COEFF_PCB_151_ZOO	log K_{ZOO} of PCB congener 151	6.75
PART_COEFF_PCB_151_LOC	log K_{LOC} of PCB congener 151	6.75
PART_COEFF_PCB_151_ROC	log K_{ROC} of PCB congener 151	6.75
PART_COEFF_PCB_170+190_DOC	log K_{DOC} of PCB congener 170+190	5.22
PART_COEFF_PCB_170+190_DIA	log K_{DIA} of PCB congener 170+190	7.22
PART_COEFF_PCB_170+190_GRE	log K_{GRE} of PCB congener 170+190	7.22
PART_COEFF_PCB_170+190_ZOO	log K_{ZOO} of PCB congener 170+190	7.22
PART_COEFF_PCB_170+190_LOC	log K_{LOC} of PCB congener 170+190	7.22
PART_COEFF_PCB_170+190_ROC	log K_{ROC} of PCB congener 170+190	7.22
PART_COEFF_PCB_172+197_DOC	log K_{DOC} of PCB congener 172+197	5.09
PART_COEFF_PCB_172+197_DIA	log K_{DIA} of PCB congener 172+197	7.09
PART_COEFF_PCB_172+197_GRE	log K_{GRE} of PCB congener 172+197	7.09
PART_COEFF_PCB_172+197_ZOO	log K_{ZOO} of PCB congener 172+197	7.09
PART_COEFF_PCB_172+197_LOC	log K_{LOC} of PCB congener 172+197	7.09
PART_COEFF_PCB_172+197_ROC	log K_{ROC} of PCB congener 172+197	7.09
PART_COEFF_PCB_180_DOC	log K_{DOC} of PCB congener 180	5.05
PART_COEFF_PCB_180_DIA	log K_{DIA} of PCB congener 180	7.05
PART_COEFF_PCB_180_GRE	log K_{GRE} of PCB congener 180	7.05
PART_COEFF_PCB_180_ZOO	log K_{ZOO} of PCB congener 180	7.05
PART_COEFF_PCB_180_LOC	log K_{LOC} of PCB congener 180	7.05
PART_COEFF_PCB_180_ROC	log K_{ROC} of PCB congener 180	7.05

Table 8.1. PCB Congener Groups Partition Coefficients (Continued)

Key Name	Description	Value ($L \cdot (kg\ C)^{-1}$)
PART_COEFF_PCB_182+187_DOC	log K_{DOC} of PCB congener 182+187	4.58
PART_COEFF_PCB_182+187_DIA	log K_{DIA} of PCB congener 182+187	6.58
PART_COEFF_PCB_182+187_GRE	log K_{GRE} of PCB congener 182+187	6.58
PART_COEFF_PCB_182+187_ZOO	log K_{ZOO} of PCB congener 182+187	6.58
PART_COEFF_PCB_182+187_LOC	log K_{LOC} of PCB congener 182+187	6.58
PART_COEFF_PCB_182+187_ROC	log K_{ROC} of PCB congener 182+187	6.58
PART_COEFF_PCB_195+208_DOC	log K_{DOC} of PCB congener 195+208	4.84
PART_COEFF_PCB_195+208_DIA	log K_{DIA} of PCB congener 195+208	6.84
PART_COEFF_PCB_195+208_GRE	log K_{GRE} of PCB congener 195+208	6.84
PART_COEFF_PCB_195+208_ZOO	log K_{ZOO} of PCB congener 195+208	6.84
PART_COEFF_PCB_195+208_LOC	log K_{LOC} of PCB congener 195+208	6.84
PART_COEFF_PCB_195+208_ROC	log K_{ROC} of PCB congener 195+208	6.84
PART_COEFF_PCB_196+203_DOC	log K_{DOC} of PCB congener 196+203	5.08
PART_COEFF_PCB_196+203_DIA	log K_{DIA} of PCB congener 196+203	7.08
PART_COEFF_PCB_196+203_GRE	log K_{GRE} of PCB congener 196+203	7.08
PART_COEFF_PCB_196+203_ZOO	log K_{ZOO} of PCB congener 196+203	7.08
PART_COEFF_PCB_196+203_LOC	log K_{LOC} of PCB congener 196+203	7.08
PART_COEFF_PCB_196+203_ROC	log K_{ROC} of PCB congener 196+203	7.08
PART_COEFF_PCB_201_DOC	log K_{DOC} of PCB congener 201	4.65
PART_COEFF_PCB_201_DIA	log K_{DIA} of PCB congener 201	6.65
PART_COEFF_PCB_201_GRE	log K_{GRE} of PCB congener 201	6.65
PART_COEFF_PCB_201_ZOO	log K_{ZOO} of PCB congener 201	6.65
PART_COEFF_PCB_201_LOC	log K_{LOC} of PCB congener 201	6.65
PART_COEFF_PCB_201_ROC	log K_{ROC} of PCB congener 201	6.65

where:

$f_{PCB\text{-}DISS}$ = fraction of PCB congener in purely dissolved phase
$f_{PCB-POC_j}$ = fraction of PCB congener bound to POC jth state
$f_{PCB\text{-}DOC}$ = fraction of PCB congener bound to DOC
C_T = PCB congener total concentration

The PCB congener total concentration is given by the expression:

$$C_T = C'_{PCB-DISS} \bullet \gamma + C_{PCB-DOC} + \sum_j C_{PCB-POC_j} \tag{8.14}$$

or equivalently as,

$$C_T = C_{PCB-DISS} + C_{PCB-DOC} + \sum_j C_{PCB-POC_j} \tag{8.15}$$

The use of Equation 8.14 is for derivational purposes only. In practice, it is very difficult to measure the concentration of purely dissolved PCBs with respect to segment volume, particularly in the sediment bed. Typically the measurement is done with respect to the water volume of the sample collected, if done at all. For the LMMBP, no pore water analysis was carried out for any of the sediment bed samples, which makes Equation 8.14 particularly useful for the present discussion.

Using Equations 8.3, 8.4, 8.5, 8.6, 8.7, 8.9, and 8.10 we can rewrite Equation 8.14 as:

$$C_T = C_{PCB-DISS}\left(\gamma + K_{DOC}C_{DOC} + \sum_j K_{POC_j}C_{POC_j}\right) \tag{8.16}$$

We can now get a more explicit expression for the fractions of purely dissolved, POC-bound, and DOC-bound PCB congeners in terms of partition coefficients, DOC concentration, and POC concentration. We do this by substituting Equation 8.16 into Equations 8.11, 8.12, and 8.13 to get the following expressions:

$$f_{PCB-DISS} = \frac{\gamma}{\gamma + K_{DOC}C_{DOC} + \sum_j K_{POC_j}C_{POC_j}} \tag{8.17}$$

$$f_{PCB-POC_j} = \frac{K_{POC_j}C_{POC_j}}{\gamma + K_{DOC}C_{DOC} + \sum_j K_{POC_j}C_{POC_j}} \tag{8.18}$$

$$f_{PCB-DOC} = \frac{K_{DOC}C_{DOC}}{\gamma + K_{DOC}C_{DOC} + \sum_j K_{POC_j}C_{POC_j}} \tag{8.19}$$

The Equations 8.17, 8.18, and 8.19 are now used to calculate the PCB congener state concentrations

$$C_{PCB-DISS} = f_{PCB-DISS}C_T \tag{8.20}$$

$$C_{PCB-POC_j} = f_{PCB-POC_j}C_T \tag{8.21}$$

$$C_{PCB-DOC} = f_{PCB-DOC}C_T \tag{8.22}$$

The PCB congener total concentration is then recalculated using Equation 8.15. This seems, at first glance, like a circular calculation because Equations 8.20, 8.21, and 8.22 need the value of the total concentration. However, this is not the case because the PCB congener total concentration is specified as an initial condition value at the beginning of a LM3 simulation. In other words, C_T is specified the very first time in the simulation, say at time = 0, and then the

respective PCB congener phase concentrations are calculated using Equations 8.20, 8.21, and 8.22. It must be emphasized that initial PCB congener total concentrations are obtained from field data measurements; thus each congener C_T is known to the modeler at time = 0. Thereafter, C_T is always calculated using Equation 8.15 for every time step. It is important to emphasize that the PCB congener phase concentrations are modeled as separate state variables in the program which means that the transport routine gets applied to each one of the phases as independent entities. The fractions of each PCB congener phase together with their respective phase concentrations are then recalculated on every time step.

The complete partitioning algorithm used within LM3 can be outlined as follows:

1. Determine whether the partitioning is being done in the water or sediment column. This will set the number of PCB phases and the value of the segment porosity in the algorithm.

2. Determine whether this is the first time calling the algorithm subroutine or not. If so, then use the initial input total concentration value of the PCB congener being modeled; otherwise calculate C_T using Equation 8.15.

3. Calculate the denominator of Equations 8.17, 8.18, and 8.19 using the values of the partition coefficients and the carbon state concentrations. Note that this is the same denominator; the calculation needs to be done only once and then stored.

4. Calculate the PCB congener phase fractions using Equations 8.17, 8.18, and 8.19.

5. Calculate the PCB congener phase concentrations using Equations 8.20, 8.21, and 8.22.

8.2 PCB Volatilization

Volatilization is the gradient-driven movement of a chemical across the air-water interface. In other words, it is the process by which a chemical goes from a gas-phase (in the air) into a dissolved phase (in the water) and vice-versa with the direction of movement determined by the gradient between air and water concentrations. Volatilization is a mass transfer process for PCBs. This means that volatilization can either be a loss or a gain mechanism of PCBs for Lake Michigan. Note that volatilization affects only the PCB congeners in their purely dissolved phases in the water column and not their DOC and POC sorbed phases.

The purely dissolved PCB congener attempts to equilibrate with the gas-phase partial pressure. The equation that governs this volatilization process is:

$$\frac{\partial C_{PCB-DISS}}{\partial t} = -\frac{k_v}{D}\left(C_{PCB-DISS} - \frac{C_{PCB-AIR}}{H/RT}\right) \tag{8.23}$$

where:

$C_{PCB\text{-}DISS}$ = purely dissolved PCB congener water concentration ($kg \cdot m^{-3}$)
$C_{PCB\text{-}AIR}$ = PCB congener air concentration ($kg \cdot m^{-3}$)
t = time (s)
k_V = conductivity of PCB through water (rate constant) ($m \cdot s^{-1}$)

D = depth of surficial water segment (m)
H = Henry's law constant ($kJ \cdot mol^{-1}$)
R = ideal gas constant = $8.315 \cdot 10^{-3}$ $kJ \cdot mol^{-1} \cdot {}^{\circ}K^{-1}$
T = water temperature (°K)

Henry's law constant generally increases with increasing vapor pressure and decreases with increasing solubility of a compound. A unique numerical value of Henry's law constant is assigned to each PCB congener based on its individual chemical properties. The Henry's law constant formula used for PCBs in LM3 is the following:

$$H_0(T) \equiv \frac{H(T)}{RT} = \exp\left(-\frac{\Delta H_H}{RT} + \frac{\Delta S_H}{R}\right) \tag{8.24}$$

where:

$H_0(T)$ = dimensionless Henry's law constant
T = interfacial temperature (°K)
R = ideal gas constant = $8.315 \cdot 10^{-3}$ $kJ \cdot mol^{-1} \cdot {}^{\circ}K^{-1}$
ΔH_H = enthalpy of phase change ($kJ \cdot mol^{-1}$)
ΔS_H = entropy of phase change ($kJ \cdot mol^{-1} \cdot {}^{\circ}K^{-1}$)

This equation was obtained from Bamford *et al*., 2002. Each PCB congener group has its own enthalpy and entropy value. The complete list of PCB congener enthalpy and entropy values, together with their corresponding input parameter names, is given in Tables 8.2 and 8.3, respectively.

The conductivity (overall mass transfer) k_V is the reciprocal of the total resistance and is given by the following formula:

$$k_V = \frac{1}{R_L + R_G} = \frac{1}{\frac{1}{K_L} + \frac{1}{K_G \times H_0(T)}} \tag{8.25}$$

where:

R_L = liquid-phase resistance ($s \cdot m^{-1}$)
R_G = gas-phase resistance ($s \cdot m^{-1}$)
K_L = liquid-phase transfer coefficient ($m \cdot s^{-1}$)
K_G = gas-phase transfer coefficient ($m \cdot s^{-1}$)

Equation 8.25 is derived using a two-layer resistance model; however, the details are not included in this document. A more in-depth description of this formula can be found in the historical reference of Whitman (1923).

Table 8.2. PCB Congener Groups Enthalpy Values

Key Name	Description	Units ($kJ \cdot mol^{-1}$)
ENTHALPY_PCB_5+8	Enthalpy of PCB congener 5+8	46
ENTHALPY_PCB_15+17	Enthalpy of PCB congener 15+17	47
ENTHALPY_PCB_16+32	Enthalpy of PCB congener 16+32	45
ENTHALPY_PCB_18	Enthalpy of PCB congener 18	35
ENTHALPY_PCB_26	Enthalpy of PCB congener 26	41
ENTHALPY_PCB_28+31	Enthalpy of PCB congener 28+31	37
ENTHALPY_PCB_33	Enthalpy of PCB congener 33	42
ENTHALPY_PCB_37+42	Enthalpy of PCB congener 37+42	40
ENTHALPY_PCB_44	Enthalpy of PCB congener 44	26
ENTHALPY_PCB_49	Enthalpy of PCB congener 49	25
ENTHALPY_PCB_52	Enthalpy of PCB congener 52	31
ENTHALPY_PCB_56+60	Enthalpy of PCB congener 56+60	30
ENTHALPY_PCB_66	Enthalpy of PCB congener 66	29
ENTHALPY_PCB_70+76	Enthalpy of PCB congener 70+76	29
ENTHALPY_PCB_74	Enthalpy of PCB congener 74	25
ENTHALPY_PCB_77+110	Enthalpy of PCB congener 77+110	39
ENTHALPY_PCB_81	Enthalpy of PCB congener 81	33
ENTHALPY_PCB_84+92	Enthalpy of PCB congener 84+92	21
ENTHALPY_PCB_85	Enthalpy of PCB congener 85	26
ENTHALPY_PCB_87	Enthalpy of PCB congener 87	33
ENTHALPY_PCB_89	Enthalpy of PCB congener 89	21
ENTHALPY_PCB_99	Enthalpy of PCB congener 99	16
ENTHALPY_PCB_101	Enthalpy of PCB congener 101	30
ENTHALPY_PCB_105+132+153	Enthalpy of PCB congener 105+132+153	68
ENTHALPY_PCB_118	Enthalpy of PCB congener 118	50
ENTHALPY_PCB_123+149	Enthalpy of PCB congener 123+149	51
ENTHALPY_PCB_138+163	Enthalpy of PCB congener 138+163	84
ENTHALPY_PCB_146	Enthalpy of PCB congener 146	59
ENTHALPY_PCB_151	Enthalpy of PCB congener 151	37
ENTHALPY_PCB_170+190	Enthalpy of PCB congener 170+190	157
ENTHALPY_PCB_172+197	Enthalpy of PCB congener 172+197	147
ENTHALPY_PCB_180	Enthalpy of PCB congener 180	144
ENTHALPY_PCB_182+187	Enthalpy of PCB congener 182+187	97
ENTHALPY_PCB_195+208	Enthalpy of PCB congener 195+208	156
ENTHALPY_PCB_196+203	Enthalpy of PCB congener 196+203	167
ENTHALPY_PCB_201	Enthalpy of PCB congener 201	145

Table 8.3. PCB Congener Groups Entropy Values

Key Name	Description	Units ($kJ \cdot mol^{-1} \cdot °K^{-1}$)
ENTROPY_PCB_5+8	Entropy of PCB congener 5+8	0.12
ENTROPY_PCB_15+17	Entropy of PCB congener 15+17	0.12
ENTROPY_PCB_16+32	Entropy of PCB congener 16+32	0.11
ENTROPY_PCB_18	Entropy of PCB congener 18	0.08
ENTROPY_PCB_26	Entropy of PCB congener 26	0.10
ENTROPY_PCB_28+31	Entropy of PCB congener 28+31	0.09
ENTROPY_PCB_33	Entropy of PCB congener 33	0.11
ENTROPY_PCB_37+42	Entropy of PCB congener 37+42	0.10
ENTROPY_PCB_44	Entropy of PCB congener 44	0.05
ENTROPY_PCB_49	Entropy of PCB congener 49	0.05
ENTROPY_PCB_52	Entropy of PCB congener 52	0.07
ENTROPY_PCB_56+60	Entropy of PCB congener 56+60	0.07
ENTROPY_PCB_66	Entropy of PCB congener 66	0.06
ENTROPY_PCB_70+76	Entropy of PCB congener 70+76	0.06
ENTROPY_PCB_74	Entropy of PCB congener 74	0.05
ENTROPY_PCB_77+110	Entropy of PCB congener 77+110	0.09
ENTROPY_PCB_81	Entropy of PCB congener 81	0.07
ENTROPY_PCB_84+92	Entropy of PCB congener 84+92	0.04
ENTROPY_PCB_85	Entropy of PCB congener 85	0.05
ENTROPY_PCB_87	Entropy of PCB congener 87	0.07
ENTROPY_PCB_89	Entropy of PCB congener 89	0.04
ENTROPY_PCB_99	Entropy of PCB congener 99	0.02
ENTROPY_PCB_101	Entropy of PCB congener 101	0.07
ENTROPY_PCB_105+132+153	Entropy of PCB congener 105+132+153	0.19
ENTROPY_PCB_118	Entropy of PCB congener 118	0.13
ENTROPY_PCB_123+149	Entropy of PCB congener 123+149	0.14
ENTROPY_PCB_138+163	Entropy of PCB congener 138+163	0.25
ENTROPY_PCB_146	Entropy of PCB congener 146	0.17
ENTROPY_PCB_151	Entropy of PCB congener 151	0.10
ENTROPY_PCB_170+190	Entropy of PCB congener 170+190	0.49
ENTROPY_PCB_172+197	Entropy of PCB congener 172+197	0.46
ENTROPY_PCB_180	Entropy of PCB congener 180	0.45
ENTROPY_PCB_182+187	Entropy of PCB congener 182+187	0.30
ENTROPY_PCB_195+208	Entropy of PCB congener 195+208	0.40
ENTROPY_PCB_196+203	Entropy of PCB congener 196+203	0.52
ENTROPY_PCB_201	Entropy of PCB congener 201	0.46

There are two formulations that are used for K_L and K_G; that is, the liquid-phase mass transfer rate and the gas-phase mass transfer rate, respectively. The formulations used in the application LM3 were recommended by the LMMBP Atmospheric Workgroup. The liquid-phase transfer coefficient, K_L, is calculated using the Wanninkhof (Wanninkhof, 1992) formulation. The gas-phase transfer coefficient, K_G, is calculated using the Schwarzenbach (Schwarzenbach *et al*., 1993) formulation.

The Wanninkhof equation for K_L, with correction for PCB congener molecular diffusivity in reference to carbon dioxide (CO_2) molecular diffusivity across the air-water interface, is given by the following formula:

$$K_L = 0.45 \cdot \sqrt{\frac{D_{water,pcb}}{D_{water,CO_2}}} \cdot U_{10}^{1.64} \cdot \frac{1}{3.6 \cdot 10^5} \tag{8.26}$$

The Schwarzenbach equation for K_G, with correction of PCB congener molecular diffusivity in reference to water vapor molecular diffusivity across the air-water interface, is given by the following formula:

$$K_G = \left(0.2\ U_{10} + 0.3\right)\left(\frac{D_{air,pcb}}{D_{air,water}}\right)^{0.61} \cdot 10^{-2} \tag{8.27}$$

where:

U_{10} = wind speed measured at 10 m above water surface ($m \cdot s^{-1}$)
$D_{water,pcb}$ = diffusivity of PCB congener in water ($m^2 \cdot s^{-1}$)
$D_{water,CO2}$ = diffusivity of carbon dioxide in water ($m^2 \cdot s^{-1}$)
$D_{air,pcb}$ = diffusivity of PCB congener in air ($m^2 \cdot s^{-1}$)
$D_{air,water}$ = diffusivity of water in air ($m^2 \cdot s^{-1}$)

The number, $3.6 \cdot 10^5$, that appears in the denominator of Equation 8.26 is actually a unit conversion factor that converts the units of K_L from $cm \cdot h^{-1}$ to $m \cdot s^{-1}$. The factor of 10^{-2} that appears in the equation of K_G, Equation 8.27, converts the units from $cm \cdot s^{-1}$ to $m \cdot s^{-1}$. K_G, is being multiplied by $H_0(T)$ in Equation 8.25 to scale the gas-phase transfer coefficient by the air-water partition coefficient. Note that K_L and K_G are both temperature- and PCB congener-dependent. That is because the diffusivities are temperature dependent, and the PCB diffusivities are congener-specific. The wind speed U_{10} is segment-specific which means that for each of the 2,318 surficial water column segments, a unique value is assigned.

The formulas used in LM3 for the diffusivities are:

$$D_{water,CO_2} = 6.053 \cdot 10^{-8} \cdot 10^{-4} \cdot \frac{T}{\eta} \tag{8.28}$$

$$D_{air,water} = 4.829 \cdot 10^{-5} \cdot 10^{-4} \cdot T^{1.5} \tag{8.29}$$

$$D_{water,pcb} = \frac{2.2 \cdot 10^{-9}}{(MW_{PCB})^{2/3}} \tag{8.30}$$

$$D_{air,pcb} = \frac{1.9 \cdot 10^{-4}}{(MW_{PCB})^{2/3}} \tag{8.31}$$

where:

T	=	water temperature (°K)
η	=	kinematic viscosity of water
MW_{PCB}	=	molecular weight of a given PCB congener

The molecular weight of a PCB congener is the weight in grams of one mole of PCB congener molecules. One mole of a particle is equal to the Avogadro's number of the particles, that is, $6.022 \cdot 10^{23}$ particles. The units of molecular weight are $g \cdot mol^{-1}$. A list of molecular weights of the PCB congeners modeled within LM3 and their corresponding input variable names is given in Table 8.4.

The kinematic viscosity of water is given by:

$$\eta = \frac{\nu}{\rho_w} \tag{8.32}$$

where:

ν	=	dynamic viscosity of water ($g \cdot cm^{-1} \cdot s^{-1}$)
ρ_w	=	density of water = 1.0 $g \cdot cm^{-3}$

Note that we are keeping the density of water, ρ_w, constant even though it varies with temperature. However, the variation with respect to temperature is less than one percent of the value of 1.0 $g \cdot cm^{-3}$; consequently the numerical error incurred when calculating the kinematic viscosity of water, Equation 8.32, is very small. The dynamic viscosity of water is given by the following equation:

$$\log_{10} \nu = \begin{cases} \left(\dfrac{1301.0}{998.333 + 8.1855(T - 20) + 0.00585(T - 20.0)^2} - 3.30233 \right) & T \le 20^\circ C \\ \left(\dfrac{1.3272(20.0 - T) - 0.001053(T - 20.0)^2}{T + 105} \right) \cdot 1.002 & T > 20^\circ C \end{cases} \tag{8.33}$$

where

T	=	water temperature (°C)

Table 8.4. PCB Congener Groups Molecular Weight

Key Name	Description	Molecular Weight ($g \cdot mol^{-1}$)
MOLWT_PCB_5+8	Molecular weight of PCB congener 5+8	228.89
MOLWT_PCB_15+17	Molecular weight of PCB congener 15+17	240.47
MOLWT_PCB_16+32	Molecular weight of PCB congener 16+32	252.30
MOLWT_PCB_18	Molecular weight of PCB congener 18	243.32
MOLWT_PCB_26	Molecular weight of PCB congener 26	255.30
MOLWT_PCB_28+31	Molecular weight of PCB congener 28+31	260.54
MOLWT_PCB_33	Molecular weight of PCB congener 33	265.78
MOLWT_PCB_37+42	Molecular weight of PCB congener 37+42	274.50
MOLWT_PCB_44	Molecular weight of PCB congener 44	278.87
MOLWT_PCB_49	Molecular weight of PCB congener 49	282.97
MOLWT_PCB_52	Molecular weight of PCB congener 52	285.43
MOLWT_PCB_56+60	Molecular weight of PCB congener 56+60	290.35
MOLWT_PCB_66	Molecular weight of PCB congener 66	296.92
MOLWT_PCB_70+76	Molecular weight of PCB congener 70+76	302.66
MOLWT_PCB_74	Molecular weight of PCB congener 74	303.48
MOLWT_PCB_77+110	Molecular weight of PCB congener 77+110	318.44
MOLWT_PCB_81	Molecular weight of PCB congener 81	309.22
MOLWT_PCB_84+92	Molecular weight of PCB congener 84+92	314.46
MOLWT_PCB_85	Molecular weight of PCB congener 85	312.22
MOLWT_PCB_87	Molecular weight of PCB congener 87	313.71
MOLWT_PCB_89	Molecular weight of PCB congener 89	315.21
MOLWT_PCB_99	Molecular weight of PCB congener 99	322.70
MOLWT_PCB_101	Molecular weight of PCB congener 101	324.20
MOLWT_PCB_105+132+153	Molecular weight of PCB congener 105+132+153	346.65
MOLWT_PCB_118	Molecular weight of PCB congener 118	336.93
MOLWT_PCB_123+149	Molecular weight of PCB congener 123+149	351.19
MOLWT_PCB_151	Molecular weight of PCB congener 151	363.36
MOLWT_PCB_138+163	Molecular weight of PCB congener 138+163	362.95
MOLWT_PCB_146	Molecular weight of PCB congener 146	359.25
MOLWT_PCB_170+190	Molecular weight of PCB congener 170+190	393.91
MOLWT_PCB_172+197	Molecular weight of PCB congener 172+197	403.24
MOLWT_PCB_180	Molecular weight of PCB congener 180	393.91
MOLWT_PCB_182+187	Molecular weight of PCB congener 182+187	400.37
MOLWT_PCB_195+208	Molecular weight of PCB congener 195+208	449.89
MOLWT_PCB_196+203	Molecular weight of PCB congener 196+203	431.23
MOLWT_PCB_201	Molecular weight of PCB congener 201	435.54

The dynamic viscosity of water, ν, is expressed in centipoise, which is equal to 100 poises with one poise being equal to 1.0 $g \cdot cm^{-1} \cdot s^{-1}$. This means that when calculating the dynamic viscosity of water by applying $\log^{-1}_{10}$ to both sides of Equation 8.33, one has to multiply by 100 to get the right magnitude of ν. We can see from Equation 8.33 that the dynamic viscosity of water has two formulas depending on whether the water temperature is above or below 20°C. These formulas were taken from Weast (1989) and the website http://scienceworld.wolfram.com/physics/ DynamicViscosity.html.

Now that we have specified all the volatilization-related equations, we then proceed to outline the general algorithm used within LM3 to calculate PCB volatilization. The steps of the algorithm are:

1. Calculate the dimensionless Henry's constant $H_0(T)$ using Equation 8.24.

2. Calculate the dynamic viscosity of water using Equation 8.33.

3. Calculate the kinematic viscosity of water using Equation 8.32.

4 Calculate the diffusivity of CO_2 in water using Equation 8.28.

5. Calculate the diffusivity of a given PCB congener in water using Equation 8.30.

6. Calculate the liquid-phase transfer coefficient using Equation 8.26.

7. Calculate the diffusivity of water vapor in air using Equation 8.29.

8. Calculate the diffusivity of a given PCB congener in air using Equation 8.31.

9. Calculate the gas-phase transfer coefficient using Equation 8.27.

10. Calculate the overall mass transfer, k_v, using Equation 8.25.

11. Divide k_v by the segment depth D.

12. Calculate net volatilization rate per unit of water volume using Equation 8.23.

13. To obtain the mass derivative term in units of $kg \cdot s^{-1}$, multiply the value obtained from Equation 8.23 by the volume of the surficial water segment in question.

CHAPTER 9

ATRAZINE KINETICS

Atrazine is a herbicide widely used in the United States to control broadleaf and grassy weeds. This pesticide is a white crystalline, solid organic compound that is available in many forms as a dry flowable, flowable liquid, water dispersible granular liquid, and a wettable powder. Atrazine is a member of the triazine class of herbicides; its chemical formula is $C_8H_{14}ClN_5$. The atrazine structural formula is shown in Figure 9.1.

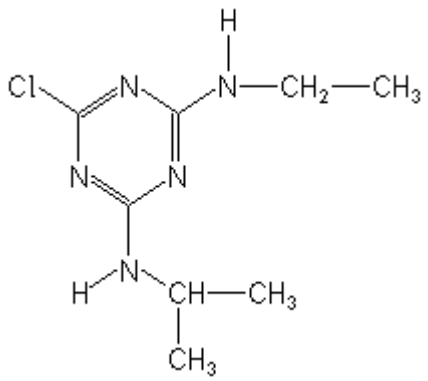


Figure 9.1. Structural formula for atrazine.

Atrazine is mainly applied to corn crops, but it is also used on sorghum, sugarcane, pineapple, Christmas tree farms, and other crops. Atrazine is used extensively on corn crops grown in the upper Midwest region of the United States, which includes Lake Michigan and its watershed. Atrazine is primarily applied directly to the soil during pre-planting or pre-emergence. It is also applied post-emergence by foliar application directly to target plants.

Atrazine is carried by water into the soil and other water bodies. Its relatively low adsorption and solubility in water makes the chemical highly mobile and susceptible to transport during runoff. Intense rain events can create large runoff loadings of atrazine into streams, rivers, and eventually lakes. Lake Michigan is one of such lakes that can receive large loadings of atrazine during heavy rain events.

Atrazine is moderately susceptible to aerobic degradation in soils, with a half-life of 60-150 days (Ribaudo and Bouzaher, 1994). Under anaerobic conditions, this degradation rate slows dramatically to a half-life of 660 days. Chemical hydrolysis, followed by degradation by soil microorganisms probably account for the most of the breakdown of atrazine in soil.

Atrazine is moderately soluble in water. It decays in water, with half-lives ranging from several days, as reported in studies on wetland systems and artificial streams, to over 300 days in larger lake systems (U.S. Environmental Protection Agency, 2001). The decay of atrazine in water occurs as a result of various processes such as chemical hydrolysis, biodegradation, and photolysis. Hydrolysis is rapid under acidic or basic conditions, but is slower at neutral pHs. Addition of humic material increases the rate of hydrolysis. Atrazine biodegradation can occur by different pathways (Zeng *et al*., 2004; Wackett *et al*., 2002; Ralebitso *et al*., 2002). Biodegradation is mediated by microorganisms, usually by some kind of bacteria. Photolysis does not appear to be a major loss mechanism in water because atrazine lacks absorption of light in the ultraviolet and visible region of the light spectrum. However, degradation of atrazine in water has been noticed when exposed to natural or artificial light. This has led researchers to believe that an indirect mechanism for photodegradation might be occurring. One such mechanism is the chemical reaction of nitrate with water and light. Nitrate is known to produce hydroxyl radical on illumination (Zepp *et al*., 1987). Hydroxyl radicals are known to be highly reactive with most organic compounds, including atrazine (Zepp *et al*., 1987; Pratap and Lemley, 1998; Balmer and Sulzberger, 1999; Acero *et al*., 2000; Tauber and von Sonntag, 2000).

Atrazine does not strongly attach to particles. Thus partitioning of atrazine to particles is not considered an important process; consequently, settling of particles in the water column has no significant effect on atrazine concentration levels in an aquatic system such as Lake Michigan. Atrazine does not bioaccumulate, but it has been shown to be toxic to aquatic animals such as frogs (Hayes *et al*., 2003). Atrazine volatilizes but not in significant amounts due to its relatively small Henry's law constant (see Section 9.2.1). See Rygwelski (2008) for a complete report of atrazine modeling results of the Lake Michigan Mass Balance Project (LMMBP).

In the sections that follow, a description of the atrazine kinetics modeled within LM3 is given. Section 9.1 discusses the approach used to model the decay of atrazine in the water column. Section 9.2 discusses the volatilization formulation used for atrazine. A discussion of the Henry's law constant used for atrazine is given in Section 9.2.1. This is followed by Sections 9.2.2 and 9.2.3, which discuss the O'Connor and Wanninkhof/Schwarzenbach volatilization scheme.

9.1 Atrazine Lumped First-Order Decay

As previously indicated, atrazine has been found to chemically break down in water under certain conditions. Laboratory experiments have been performed to test atrazine stability in water under various physical conditions such as light exposure and water temperature (Burkhard and Guth, 1976; Schabacker, 1991; Konstantinou *et al*., 2001; Rebelo *et al*., 2007). Additionally, biological conditions have been studied such as the effect of bacteria and other microorganisms on the decay of atrazine. However, it is very difficult to relate laboratory studies of atrazine decay to actual conditions in aquatic systems such as Lake Michigan. This is because laboratory experiments are conducted under artificial conditions that do not capture the full complexity of the biology, chemistry, and physics present in aquatic systems. To that effect,

it was decided to use a lumped first-order decay term to account for atrazine's decay in the water column of Lake Michigan (Rygwelski, 2008). It must be emphasized that LM3 does not simulate the decay products of atrazine nor does it keep track of them. The Large Lakes Research Station (LLRS) has not contemplated simulating atrazine decay's breakdown products as of this writing.

This lumped first-order decay term encompasses all the known and unknown biological chemical, and physical processes that contribute to the break down of atrazine in the water column. This approach conveniently handles our lack of knowledge of the precise nature of decay of atrazine in the water. This lumped first-order decay term is represented in LM3 as an input parameter. Thus the user has the option to adjust the parameter within observed ranges of decay.

A simple first-order decay process (ignoring transport and other kinetic processes) is described below:

$$C\left(t_0 + \Delta t\right) \approx C\left(t_0\right) + \Delta t \left.\frac{dC}{dt}\right|_{t=t_0} \tag{9.1}$$

where:

$$\left.\frac{dC}{dt}\right|_{t=t_0} = -k_A C\left(t_0\right) \tag{9.2}$$

and

$C(t)$ = atrazine concentration at time t
k_A = atrazine decay rate (s^{-1}) = DECAY_RATE_ATRAZINE in Table 9.1
t_0 = arbitrary initial time
Δt = time step

The user has to specify the decay rate, k_A, as an input value; otherwise, LM3 will use the default value shown in Table 9.1. This input value could be adjusted if desired; however, this was not done for LM3-Atrazine because the decay was previously estimated. This first-order decay process can be viewed as a loss mechanism for atrazine because it is essentially reducing the amount of the pesticide in the water column.

Table 9.1. Atrazine Parameters

Key Name	Description	Default Value	Units
DECAY_RATE_ATRAZINE	Lumped first-order decay rate of atrazine	0.0	s^{-1}
MOLTW_ATZ	Molecular weight of atrazine	215.69	$g \cdot mol^{-1}$

9.2 Atrazine Volatilization

Atrazine undergoes volatilization in Lake Michigan although to a lesser degree than other chemicals such as polychlorinated biphenyls (PCBs). Atrazine tends to remain in the dissolved phase in the water column with just a small amount of the chemical volatilizing out of the water and into the atmosphere.

The equation that governs the volatilization process of atrazine is:

$$\frac{\partial C_{ATZ-WATER}}{\partial t} = -\frac{k_v}{D}\left(C_{ATZ-WATER} - \frac{C_{ATZ-AIR}}{H/RT}\right) \tag{9.3}$$

where:

$C_{ATZ-WATER}$ = atrazine water concentration (kg · m^{-3})
$C_{ATZ-AIR}$ = atrazine air concentration (kg · m^{-3})
t = time (s)
k_v = conductivity of atrazine through water (rate constant) (m · s^{-1})
D = depth of surficial water segment (m)
H = Henry's law constant (kJ · mol^{-1})
R = ideal gas constant = 8.315 · 10^{-3} kJ · mol^{-1} · $°K^{-1}$
T = water temperature (°K)

The conductivity of atrazine through water, k_v, has the same mathematical expression as the PCBs (see Equation 8.25):

$$k_v = \frac{1}{\frac{1}{K_K} + \frac{1}{K_G \times H(T)/RT}} \tag{9.4}$$

where:

K_L = liquid-phase transfer coefficient (m · s^{-1})
K_G = gas-phase transfer coefficient (m · s^{-1})

The formula used for Henry's law constant is discussed in the section that follows, Section 9.2.1. There are two formulations that are used for K_L and K_G; that is, the liquid-phase mass transfer rate and the gas-phase mass transfer rate, respectively. These were incorporated in LM3. The formulations are known as the O'Connor formulation which is discussed in Section 9.2.2, and the Wanninkhof and Schwarzenbach formulation, which is discussed in Section 9.2.3. Only one formulation is used for a given simulation; it is up to the user to decide which formulation to use. There is no strong argument to favor one formulation over the other because, for atrazine, volatilization is not a significant loss/gain process in Lake Michigan.

9.2.1 Henry's Law Constant

The algorithm to calculate the temperature-dependent Henry's law coefficient of atrazine was derived from Scholtz *et al.* (1999) and Miller (1999). The form of the equation is the following:

$$\log_{10}\left(H_0\left(T\right)\right) = \log_{10}\left(H_0\left(T_{ref}\right)\right) - \Delta H\left(\frac{T^{-1} - T_{ref}^{-1}}{2.303\ R}\right) \tag{9.5}$$

where

H_0 = the dimensionless Henry's law coefficient at temperature T
T = water temperature in Kelvin = T (°C) + 273.15
T_{ref} = reference temperature equal to 298.15°K (25°C)
$H_0(T_{ref})$ = Henry's law coefficient at the reference temperature T_{ref} with a value of $8.1 \cdot 10^{-8}$.
ΔH = the enthalpy of volatilization taken at 50 $kJ \cdot mol^{-1}$
R = ideal gas constant whose value is $8.206 \cdot 10^{-5} \cdot atm \cdot m^3 \cdot mol^{-1} \cdot {}^\circ K^{-1}$

When substituting the numerical values above into the corresponding terms of Equation 9.5, the units of ΔH need to be converted from $kJ \cdot mol^{-1} \cdot$ to $atm \cdot m^3 \cdot mol^{-1}$ using the conversion factor $9.86923 \cdot 10^{-3}$. Equation 9.5 then becomes:

$$\log_{10}\left(H_0\left(T\right)\right) = \log_{10}\left(8.1 \cdot 10^{-8}\right) - 50.0 \cdot 9.86923 \cdot 10^{-3} \cdot \left(\frac{\left(T\left(°C\right) + 273.15\right)^{-1} - (298.15)^{-1}}{2.303 \cdot 8.206 \cdot 10^{-5}}\right) \tag{9.6}$$

$H_0(T)$ is then explicitly obtained by taking the $\log^{-1}_{10}$ on both sides of the equation above. Note that H(T) has already been normalized to be dimensionless. $H_0(T)$ is represented with the variable name KAW inside the LM3 source code.

9.2.2 O'Connor Formulation

The O'Connor formulation is one of two atrazine volatilization formulation choices provided in LM3. The other one is the Wanninkhof and Schwarzenbach formulation which is discussed in Section 9.2.3.

The atrazine O'Connor volatilization scheme is a wind-driven formulation (O'Connor, 1983a,b), which can be used in cases where mass transfer is controlled by the wind velocity, typical of lakes and reservoirs. The O'Connor formulation involves the calculation of several parameters. The calculation of the parameters as carried out within the LM3 source code is detailed below.

9.2.2.1 Kinematic Viscosity and Density of Air

The kinematic viscosity of air and the density of air were calculated using empirical formulas that are temperature-dependent.

$$\eta_{air} = 0.1 \cdot \left(1.32 + 0.009 \cdot T\left(°C\right)\right) \tag{9.7}$$

$$\rho_{air} = 1.293 \cdot 10^{-3} / \left(1.0 + 0.00367 \cdot T\left(^{\circ}C\right)\right) \tag{9.8}$$

where:

η_{air} = kinematic viscosity of air ($cm^2 \cdot s^{-1}$)
ρ_{air} = density of air ($g \cdot cm^{-3}$)
T(°C) = surficial water temperature (°C)

Note that the temperature, T(°C), used in the equations above was not the air temperature but the water temperature of the surficial layer. Strictly speaking it is not correct to use water temperature as a substitute of air temperature but in doing so we are introducing just a small error. We used water temperature due to lack of air temperature data at the air-water interface.

9.2.2.2 Diffusivity of Atrazine in Air and in Water

The diffusivities of atrazine in air and water were needed because they were used in the calculations of the liquid and gas exchange coefficients.

$$D_{air,atz} = \frac{1.9 \cdot 10^{-1}}{(MW_{atz})^{2/3}} \tag{9.9}$$

$$D_{water,atz} = \frac{22 \cdot 10^{-9}}{(MW_{atz})^{2/3}} \tag{9.10}$$

where:

MW_{atz} = molecular weight of atrazine = 215.69 $g \cdot mol^{-1}$ = MOLWT_ATZ in Table 9.1

9.2.2.3 Shear Velocity Iteration Scheme

Another parameter that needed to be determined was the shear velocity, U*, which was calculated iteratively. It was tested for convergence at every step of the iteration. The steps are:

1. Set shear velocity convergence test variable.

$$U^*_{old} = \sqrt{C_d} \cdot V_{wind} \cdot 100 \tag{9.11}$$

where

C_d = dimensionless drag coefficient = 0.0011 (O'Connor, 1983b)
V_{wind} = wind velocity at 10 m above the water surface ($m \cdot s^{-1}$)
U^*_{old} = initial iterative shear velocity value ($cm \cdot s^{-1}$)

The number, 100, in Equation 9.11 is a conversion factor; it converts the units of U^*_{old} from $m \cdot s^{-1}$ to $cm \cdot s^{-1}$.

2. Calculate the shear velocity.

$$U^* = \frac{\kappa \bullet V_{wind} \bullet 100}{\ln\left(1000 \bullet \left(\frac{1}{Z_e} + \frac{\lambda_1 \bullet U^*_{old} \bullet \exp\left(-\frac{U^*_{old}}{U^*_{Trans}}\right)}{\eta_{air}}\right)\right)} \tag{9.12}$$

where

κ	=	von Karman constant = 0.40 (O'Connor, 1983b)
λ_1	=	dimensionless factor = 3.0 (O'Connor, 1983b)
η_{air}	=	kinematic viscosity of air, given by Equation 9.5
U^*_{Trans}	=	transitional shear velocity = 10 $cm \cdot s^{-1}$ (O'Connor, 1983b)
V_{wind}	=	wind velocity at 10 m above the water surface
Z_e	=	limiting roughness length = 0.35 cm (O'Connor, 1983b)

The factor of 100 converts the U* units from $m \cdot s^{-1}$ to $cm \cdot s^{-1}$.

3. Take the absolute value of the difference between the new and old shear velocity.

$$U_{CONV} = \left|U^* - U^*_{old}\right| \tag{9.13}$$

4. Reset the old shear velocity with the new one.

$$U^*_{old} = U^* \tag{9.14}$$

5. Test for convergence: If $U_{CONV} > 0.5$ then repeat Steps 2 to 4.

Convergence is achieved when $U_{conv} \leq 0.5$. The cut-off value of 0.5 was chosen based on tests performed to determine the number of iterations typically needed to calculate U*. It was determined that after four or five iterations U_{conv} is less than 1% of the U* iterated value.

9.2.2.4 Contributing Coefficients $Z_0(U^*)$, $\Gamma(U^*)$, K_δ, and K_τ to the Liquid-Phase Transfer Coefficient K_L

The computation of the liquid-phase mass transfer rate K_L depends on the calculation of four coefficients: the roughness length Z_0(UI), the dimensionless sublayer thickness Γ(U*), the transfer coefficient for the diffusional layer K_δ, and the surface renewal transfer coefficient K_τ. Their mathematical expressions are:

$$Z_0(U^*) = \left(\frac{1}{Z_e} + \frac{\lambda_1 \bullet U^* \bullet \exp\left(-\frac{U^*}{U^*_{Trans}} \right)}{\eta_{air}} \right)^{-1} \tag{9.15}$$

where

$Z_0(U^*)$ = roughness height (cm)
Z_e = limiting roughness length = 0.35 cm (O'Connor, 1983b)
λ_1 = dimensionless factor = 3.0 (O'Connor, 1983b)
U^* = determined iteratively using Equation 9.12
U^*_{Trans} = transitional shear velocity = 10 cm · s^{-1} (O'Connor, 1983b)

$$\Gamma(U^*) = \begin{cases} \Gamma_0 \bullet \frac{U^*}{U^*_c} \bullet \exp\left(1.0 - \frac{U^*}{U^*_c}\right) & U^* \geq U^*_c \\ \Gamma_0 & U^* < U^*_c \end{cases} \tag{9.16}$$

where

$\Gamma(U^*)$ = dimensionless sublayer thickness
Γ_0 = critical sublayer thickness = 5 (O'Connor, 1983b)
U^*_c = critical shear velocity = 11.0 cm · s^{-1} (O'Connor, 1983b)

$$K_\delta = \left(\frac{D_{water,atz} \bullet 10^4}{\eta_w} \right)^{2/3} \bullet \kappa^{1/3} \bullet \sqrt{\frac{\rho_{air}}{\rho_w}} \bullet U^* \bullet \frac{1}{\Gamma(U^*)} \tag{9.17}$$

$$K_\tau = \sqrt{\frac{\left(D_{water,atz} \bullet 10^4\right) \bullet U^*}{\kappa \bullet Z_0(U^*)} \bullet \left(\frac{\rho_{air} \bullet \eta_{air}}{\rho_w \bullet \eta_w} \right)} \tag{9.18}$$

where

K_δ = transfer coefficient for the diffusional layer (cm · s^{-1})
K_τ = surface renewal transfer coefficient (cm · s^{-1})
$D_{water,atz}$ = given by Equation 9.10
κ = von Karman constant = 0.40 (O'Connor, 1983b)
$\Gamma(U^*)$ = given by Equation 9.16
$Z_0(U^*)$ = given by Equation 9.15
ρ_{air} = density of air, given by Equation 9.8

η_{air} = kinematic viscosity of air, given by Equation 9.7
ρ_w = density of water = 1.0 g · cm^{-3}
η_w = kinematic viscosity of water, given by Equation 8.32

9.2.2.5 Liquid-Phase Transfer Coefficient

The expression for the liquid-phase transfer coefficient, K_L, is:

$$K_L = \frac{10^{-2}}{\frac{1}{K_\delta} + \frac{1}{K_\tau}} \tag{9.19}$$

where:

K_L = liquid-phase transfer coefficient (m · s^{-1})

K_δ and K_τ are given by Equations 9.17 and 9.18, respectively.

The factor 10^{-2} converts the units of K_L from cm · s^{-1} to m · s^{-1}.

9.2.2.6 Gas-Phase Transfer Coefficient

The gas-phase transfer coefficient, K_G, had a simpler calculation scheme than the liquid exchange coefficient.

1. Calculate the shear velocity as a function of wind speed.

$$U^* = \sqrt{C_d} \bullet V_{wind} \tag{9.20}$$

where

C_d = dimensionless drag coefficient = 0.0011
V_{wind} = wind velocity at 10 m above the water surface (m · s^{-1})

C_d is not a true constant but varies as a function of wind speed; however, the value of 0.0011 is representative of wind speeds between one m · s^{-1} and 12 m · s^{-1} (O'Connor, 1983b). Note that Equation 9.20 represents the same shear velocity that was calculated with Equation 9.11 except that the units are different; Equation 9.11 is in units of cm · s^{-1} (thus the factor of 100) while Equation 9.20 is in units of m · s^{-1}.

2. Multiply $D_{air,atz}$ by 1.0 · 10^4 to convert from m^2 · s^{-1} to cm^2 · s^{-1}. Then calculate the ratio of the kinematic viscosity of air to the diffusivity of atrazine in air, which is commonly known as the Schmidt Number for air.

$$SC_a = \frac{\eta_{air}}{D_{air,atz} \bullet 10^4} \tag{9.21}$$

3. Use the results from 1 and 2 to calculate the gas exchange coefficient, K_G.

$$K_G = U^* \bullet \frac{\kappa^{1/3}}{\lambda} \bullet \left(\frac{1}{SC_a}\right)^{2/3} \tag{9.22}$$

where:

κ	=	von Karman constant = 0.4 (O'Connor, 1983b)
λ	=	dimensionless viscous sublayer thickness = 4 (O'Connor, 1983b)
U^*	=	given by Equation 9.20
SC_a	=	given by Equation 9.21

9.2.2.7 Overall Mass Transfer Coefficient

The gas-phase transfer coefficient was normalized by multiplying it by the air-water partition coefficient or Henry's law constant.

$$K_{GH} = K_G \bullet H(T) \tag{9.23}$$

where:

K_{GH} = scaled gas-phase transfer coefficient.

$H_0(T)$ is obtained from Equation 9.6.

The overall mass transfer is given by the expression below.

$$K_V = \frac{K_L \bullet K_{GH}}{K_L + K_{GH}} \tag{9.24}$$

where:

K_V = overall mass transfer coefficient

K_V is normalized with respect to segment depth.

$$K_{LA} = \frac{K_V}{BL_{seg}} \tag{9.25}$$

where

BL_{seg} = thickness of surficial water column segment in m

9.2.2.8 Volatilization Rate

$$VLT = -K_{LA} \bullet \left(C_{water,atz} - \frac{C_{air,atz}}{H(T)} \right) \tag{9.26}$$

where:

VLT = volatilization rate ($kg \cdot s^{-1} \cdot m^{-3}$)

K_{LA} is given by Equation 9.25

9.2.2.9 Steps of the O'Connor Algorithm

Now that we have presented all the volatilization-related equations corresponding to the O'Connor algorithm, we proceed to outline the calculation used within LM3. The steps of the algorithm are:

1. Calculate the dimensionless Henry's constant H(T) using Equation 9.6.

2. Calculate the kinematic viscosity of air η_{air} using Equation 9.7.

3. Calculate the air density ρ_{air} using Equation 9.8.

4. Calculate the diffusivity of atrazine in air $D_{air,atz}$ using Equation 9.9.

5. Calculate the diffusivity of atrazine in water $D_{water,atz}$ using Equation 9.10.

6. Calculate the shear velocity U* using the iterative scheme outlined in Equations 9.11 through 9.14.

7. Calculate the roughness height $Z_0(U^*)$ using Equation 9.15.

8. Calculate the dimensionless sublayer thickness $\Gamma(U^*)$ using Equation 9.16.

9. Calculate the transfer coefficient for the diffusional layer K_δ using Equation 9.17.

10. Calculate the surface renewal transfer coefficient K_τ using Equation 9.18.

11. Calculate the liquid-phase transfer coefficient K_L using Equation 9.19.

12. Calculate the shear velocity U* using Equation 9.20.

13. Calculate the Schmidt number for air SC_a using Equation 9.21.

14. Calculate the gas exchange coefficient K_G using Equation 9.22.

15. Calculate the scaled gas-phase transfer coefficient K_{GH} using Equation 9.23.

16. Calculate the overall mass transfer coefficient K_V using Equation 9.24.

17. Calculate the water-depth normalized overall mass transfer coefficient K_{LA} using Equation 9.25.

18. Calculate the volatilization rate VLT using Equation 9.26.

9.2.3 Wanninkhof and Schwarzenbach Formulation

The Wanninkhof and Schwarzenbach formulation is the other volatilization algorithm available for atrazine within the LM3 source code. It is a more recent formulation than O'Connor's: it was developed in the early 1990s, while the O'Connor formulation dates back to the early 1980s.

The Wanninkhof and Schwarzenbach formulation only pertains to the way the liquid and gas transfer coefficients, K_L and K_G, are calculated (see Equation 8.25). The calculation of Henry's law constant for atrazine is still given by Equation 9.5.

The liquid-phase transfer coefficient, K_L, is calculated using the Wanninkhof (1992) formulation. The gas-phase transfer coefficient, K_G, is calculated using the Schwarzenbach *et al*. (1993) formulation.

The Wanninkhof equation for K_L, with correction for atrazine molecular diffusivity in reference to carbon dioxide (CO_2) molecular diffusivity across the air-water interface, is given by the following formula:

$$K_L = 0.45 \cdot \sqrt{\frac{D_{water,atz}}{D_{water,CO_2}}} \cdot U_{10}^{1.64} \cdot \frac{1}{3.6 \cdot 10^5} \tag{9.27}$$

The Schwarzenbach equation for K_G, with correction of atrazine molecular diffusivity in reference to water vapor molecular diffusivity across the air-water interface, is given by the following formula:

$$K_G = \left(0.2\, U_{10} + 0.3\right) \cdot \left(\frac{D_{air,atz}}{D_{air,water}}\right)^{0.61} \cdot 10^{-2} \tag{9.28}$$

where:

U_{10} = wind speed measured at 10 m above water surface ($m \cdot s^{-1}$)
$D_{water,atz}$ = diffusivity of atrazine in water ($m^2 \cdot s^{-1}$); it is given by Equation 9.10
$D_{water,CO2}$ = diffusivity of carbon dioxide in water ($m^2 \cdot s^{-1}$); it is given by Equation 8.28
$D_{air,atz}$ = diffusivity of atrazine in air ($m^2 \cdot s^{-1}$); it is given by Equation 9.9
$D_{air,water}$ = diffusivity of water in air ($m^2 \cdot s^{-1}$); it is given by Equation 8.29

Equations 9.27 and 9.28 are the atrazine equivalent of the PCB Equations 8.26 and 8.27 (see Section 8.2).

Now that we have specified all the volatilization-related equations, we then proceed to outline the general algorithm used within LM3 to calculate atrazine volatilization using the Wanninkhof and Schwarzenbach formulation. The steps of the algorithm are:

1. Calculate the dimensionless Henry's law constant $H_0(T)$ using Equation 9.6.

2. Calculate the dynamic viscosity of water using Equation 8.33.

3. Calculate the kinematic viscosity of water using Equation 8.32.

4. Calculate the diffusivity of CO_2 in water using Equation 8.28.

5. Calculate the diffusivity of atrazine in water using Equation 9.10.

6. Calculate the liquid-phase transfer coefficient using Equation 9.27.

7. Calculate the diffusivity of water vapor in air using Equation 8.29.

8. Calculate the diffusivity of atrazine in air using Equation 9.9.

9. Calculate the gas-phase transfer coefficient using Equation 9.28.

10. Calculate the overall mass transfer, k_v, using Equation 9.4.

11. Divide k_v by the segment depth D.

12. Calculate net volatilization rate per unit of water volume using Equation 9.3.

13. To obtain the mass derivative term in units of $kg \cdot s^{-1}$, multiply the value obtained from Equation 9.3 by the volume of the surficial water segment in question.

CHAPTER 10

PROGRAM CODE, ORGANIZATION, AND MANAGEMENT

Source code for the LM3 water quality model involves tens of thousands of lines. Organization and management of the program has been performed using top-down algorithm procedures together with an independent source code management tool to track revision information throughout the development of LM3. The source code management tool used for LM3 is known as the Revision Control System (RCS) (see Appendix D).

10.1 The LM3 Algorithm

Wherever possible, the LM3 algorithm was broken into initialization, batch, and completion phases. Separate calculation tasks were assigned to individual subroutines and functions where appropriate. The main program, subroutines, and functions were each assigned unique source files. The program source code was written in FORTRAN 90. Compilation and loading/linking tasks were automated using the Linux utility application known as *make*. These tasks were specified as a set of instructions within a file named Makefile. The file Makefile is used by *make* to generate compilation and loading/linking commands which, in turn, are executed by the operating system. Makefile was included in the distribution of the source code. Minor platform dependence was encountered in some of the LM3 supporting code and these instances were dealt with by the C Preprocessor (cpp) pre-processing the source code before compilation[1]. A detailed discussion of the LM3 source code will not be provided here. Those intending to make modifications to the program should study the program source code and gain a thorough understanding of its details before proceeding. However, the overall program structure and the organization of the contributing elements are reasonably straightforward and are now summarized.

The basic flow of control for LM3 is summarized in Figures 10.1 through 10.5. All references to the LM3 subroutines and functions are included in these diagrams. Library functions and their references are excluded from these flow diagrams except the functions FIINIT and FIGO shown on Figure 10.2. Figure 10.1 shows the algorithm flow of control for the main program which consists of three subroutine calls to perform program initialization (LM3INIT), the main or batch element of LM3 execution (LM3BAT), and handling of the termination phase of the program (LM3FIN).

---

[1]See the LM3 and UT library sources for details on this issue.

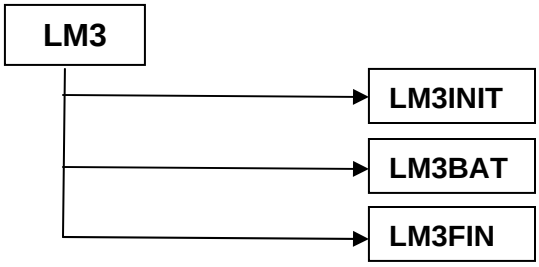


Figure 10.1. Topmost flow of control for the LM3 program.

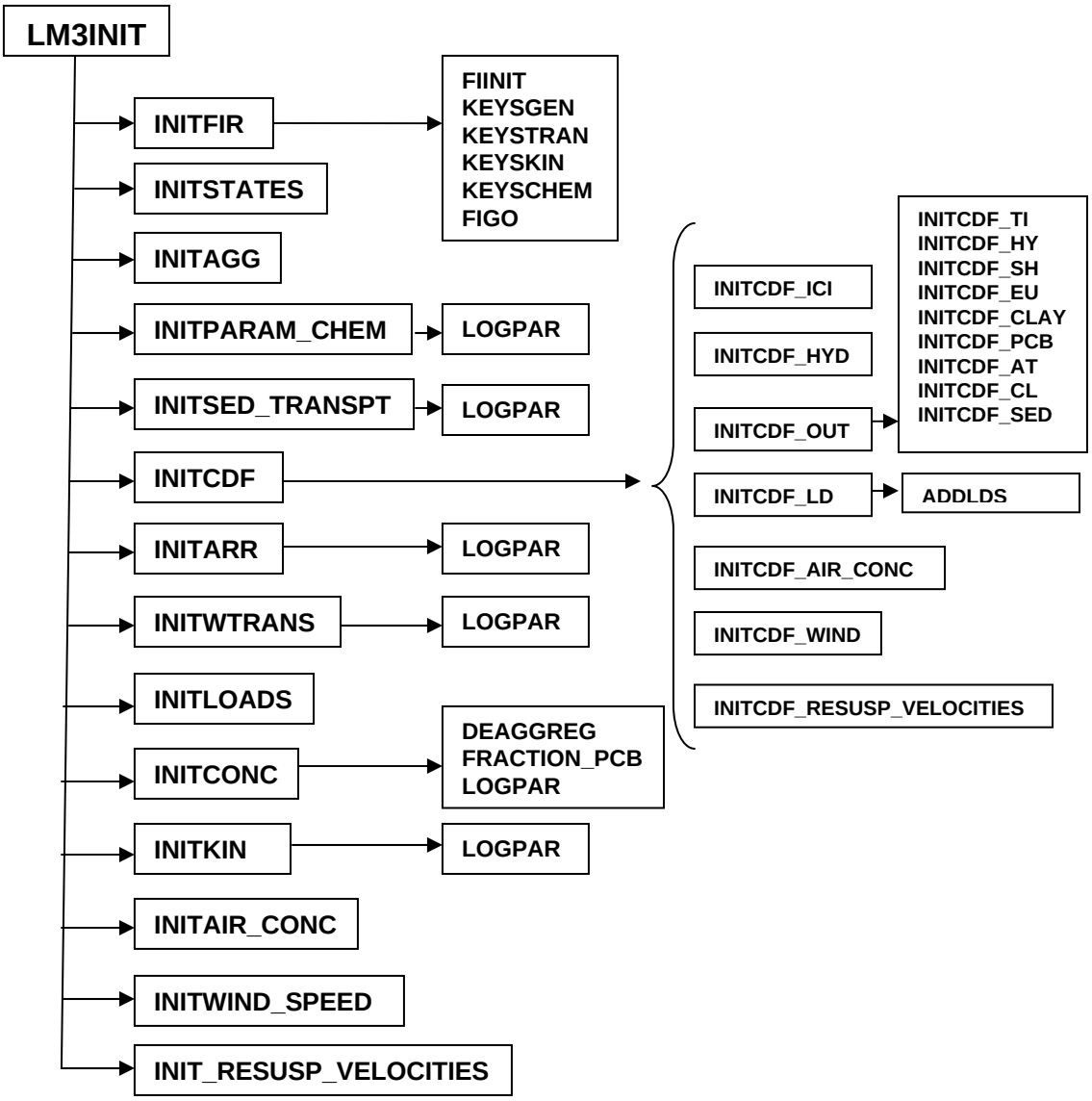


Figure 10.2. Flow of control for LM3 initialization.

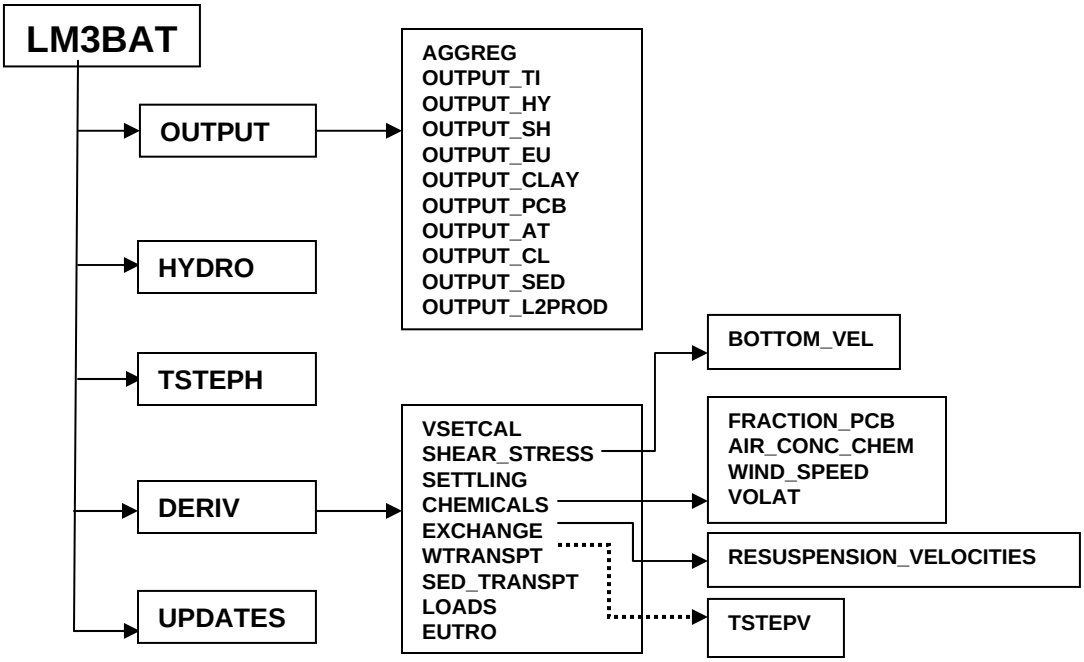


Figure 10.3. Flow of control for the LM3 main or batch level calculation.

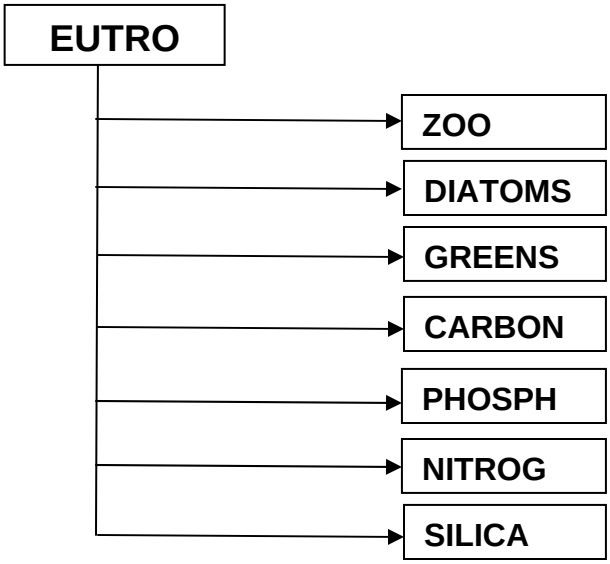


Figure 10.4. Flow of control for the LM3 eutrophication calculation.

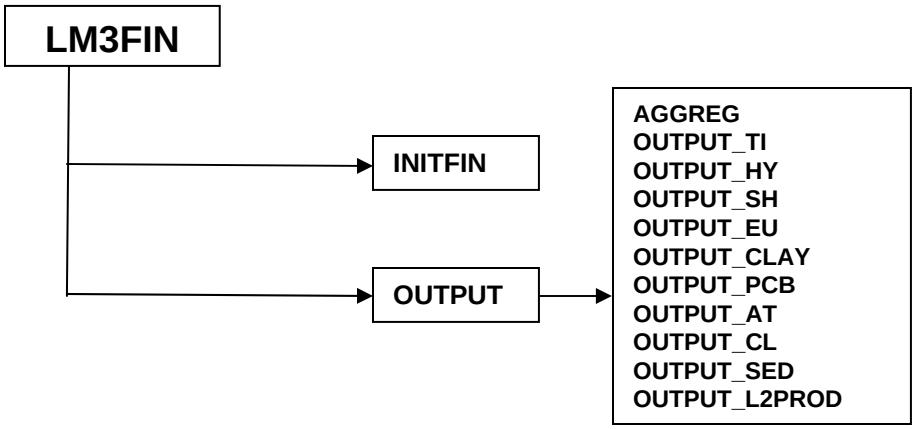


Figure 10.5. Flow of control for the LM3 finishing level calculation.

The initialization phase of the program is broken into the components described in Figure 10.2. The routine INITFIR is called to initialize FIREAD parameter handling for the program.

The UT (Settles, 1998) library interface for FIREAD is initialized (FIINIT). Parameter (KEYSTRAN), eutrophication kinetics (KEYSKIN), and chemicals kinetics (KEYSCHEM) are defined; the input deck is then read and FIREAD key values are assigned to the respective FORTRAN 90 variables (FIGO). Program initialization continues with definition and assignment of lookup tables for the different simulated states (INITSTATES).

Initialization of high-resolution (level 3) to low-resolution (level 2) geometry information for aggregation and de-aggregation purposes is performed (INITAGG); see Zhang (2006) for a detailed description of the level 2 geometry, Section 1.2 for a brief one, and Section 5.1 for a description of the high-resolution grid. The arrays needed for the volatilization calculation of polychlorinated biphenyls (PCBs) and atrazine are then initialized (INITPARAM_CHEM). The program proceeds to read mapping, geometry data, and time-invariant hydrodynamics. This is followed by the initialization of arrays needed for the sediment bed transport subroutine (INITSED_TRANSPT).

The next step is the definition of the Common Data Format (CDF) structures for the input and output data streams (INITCDF). This includes calls to subroutines supporting the initialization of the input initial conditions (INITCDF_ICI) and input hydrodynamics (INITCDF_HYD). Output CDF structures are defined (INITCDF_OUT), including output time records (INITCDF_TI), output hydrodynamics values (INITCDF_HY) [not to be confused with the input hydrodynamics definitions subroutine (INITCDF_HYD)], output shear calculation values (INITCDF_SH), output eutrophication state values (INITCDF_EU), output atrazine state variables (INITCDF_AT), output clay state values (INITCDF_CLAY), output chloride state values (INITCDF_CL), and output sediment transport values (INITCDF_SED). The subroutine supporting initialization of input load values is then called (INITCDF_LD) followed by the subroutine supporting the initialization of PCBs and atrazine vapor concentrations (INITCDF_AIR_CONC). The initialization of input wind speed values is in turn performed (INIT_WIND). Finally, the initialization of resuspension velocities is carried out (INITCDF_RESUSP_VELOCITIES).

Program initialization continues with the dynamic memory allocations of global arrays (INITARR). The transport algorithm is then initialized (INITWTRANS) which defined the simulation geometry, assignment of flow faces, vertical segmentation, and the setup of time-independent parameters used by the ULTIMATE QUICKEST (UQ) algorithm. The load calculation is then initialized (INITLOADS), and initial concentration values are read in (INITCONC) and the values reassigned from the coarse level 2 to the high-resolution grid (DEAGGREG). The fraction of PCBs in dissolved, dissolved organic carbon (DOC), and carbon sorbed components is calculated (FRACTION_PCB). Changes to maximum allowed concentration values for all modeled states are logged (LOGPAR). Kinetics calculations are initialized (INITKIN) including parameter logging (LOGPAR) and dynamic allocation of arrays.

Further program initialization continues with the input of the initial chemical vapor concentration values (INITAIR_CONC). The initial wind speed values are then read in (INITWIND_SPEED). Finally, the initial resuspension velocities values are read in (INIT_RESUSP_VELOCITIES).

After the program initialization is completed, the main or batch processing phase of LM3 is invoked by calling the subroutine LM3BAT. This subroutine has a time marching loop which iterates over the sequential time values of the simulation period. Program output corresponding to the previous iteration (or the program initialization) is generated (OUTPUT). The high-resolution model output is aggregated onto the coarse level 2 grid (AGGREG). Time values are then written (OUTPUT_TI), selected hydrodynamics values are written (OUTPUT_HY), elements of the shear calculation are output (OUTPUT_SH), and eutrophication state information is written (OUTPUT_EU). Clay state information is output (OUTPUT_CLAY), PCB congeners state information is output (OUTPUT_PCB), atrazine state information is output (OUTPUT_AT), chloride state information is output (OUTPUT_CL), and information used by the sediment calculation is passed to the output CDF stream (OUTPUT_SED). The program then stores (but does not actually write) primary productivity values from the eutrophication calculation that has been aggregated onto the level 2 calculation grid (OUTPUT_L2PROD).

After program output is written, hydrodynamics flow and dispersion fields are updated (HYDRO). The program time step is then calculated (TSTEPH). The bulk of the LM3 calculation is consumed by the calculation of the time derivatives of state variable masses, represented by the array DTM throughout the program (DERIV). There are several elements that are part of this calculation. The first element is the calculation of particle settling velocities (VSETCAL) which at the present time is not calculated but is set to the input values specified by the user. The second element is the calculation of bottom shear stress (SHEAR_STRESS) which then requires the calculation of the bottom orbital velocity (BOTTOM_VEL). The third element is the calculation of the water column bottom layer settling flux (SETTLING). The fourth element is the calculation of kinetics and partition processes associated with PCBs and atrazine (CHEMICALS). There are several sub-elements within this fourth element which are the calculation of fractions of PCB phases in the water and sediment column (FRACTION_PCB), the update of PCB and/or atrazine vapor concentration values (AIR_CONC_CHEM), the update of wind speed values (WIND_SPEED), the calculation of PCBs and/or atrazine volatilization (VOLAT), and the inline calculation of atrazine's lumped first-order decay rate (CHEMICALS). The fifth element is the calculation of settling and resuspension fluxes between the water column and the sediment bed (EXCHANGE). The calculation of resuspension fluxes requires the update of resuspension velocities (RESUSPENSION_VELOCITIES). The sixth element is

the calculation of the UQ water column transport (WTRANSPT)[1]. The subroutine WTRANSPT includes a call to another subroutine named TSTEPV which is shown in Figure 10.3 by a dashed line. The dashed line means that TSTEPV is not being used as of the writing of this document. This subroutine is supposed to calculate a separate time step, for vertical transport, but the algorithm used for this calculation is not fully operational at the present time.

The seventh element is the calculation of the UQ sediment bed transport (SED_TRANSPT). The eighth element is the calculation of load contributions (LOADS) to the mass derivative. The last element is the calculation of autochthonous solids production (EUTRO); that is, the calculation of eutrophication kinetics. Elements of this calculation are discussed below. Concentration mass, volume, and the simulation time counter are then updated (UPDATES), at which time concentration tests are performed to make certain that pre-defined maximum or negative values of state concentrations have not been violated.

The autochthonous solids production or eutrophication calculation (EUTRO) is called as a part of the time derivative calculation (DERIV) and simulated plankton growth in the lake and its decay into detrital carbon states. Light attenuation and light limitation variables are calculated inline, after which calculations for zooplankton (ZOO), diatoms (DIATOMS), non-diatom phytoplankton (GREENS), carbon (CARBON), phosphorus (PHOSPH), nitrogen (NITROG), and silica kinetics (SILICA) are performed. The subroutine calls are summarized in Figure 10.4.

After the time marching integration loop is completed, the termination phase of the program (LM3FIN) is called. The simple flow of control is shown in Figure 10.5. Actions taken to complete the time integration are performed (INTFIN); this routine is currently empty and is only present as a place maker in the algorithm. Output is written representing the state of the simulation at its completion (OUTPUT). This includes all or a subset of the following: simulation time, selected hydrodynamics values, shear stress, eutrophication, clay, PCBs, atrazine, chloride, and sediment bed concentrations. The main output file is then closed and primary productivity values are written to another, previously designated, output file.

10.1.1 LM3 Individual Program Units

A detailed, individual description of the LM3 subroutines, modules, generic subroutines, and include files follows.

10.1.1.1 Main Program

LM3
The LM3 source file is the main control file of the application. It operates the calling sequence for the input/initialization, execution/output, and closing tasks of the application. It calls the subroutines LM3INIT, LM3BAT, and LM3FIN.

---

[1]The wtranspt.F code and support source are products of the Waterways Experiment Station (WES) group – R. Chapman, T. Cole, and T. Gerald. The algorithm employed is designated to provide an accurate coupling to the Princeton Ocean Model (POM) hydrodynamics calculation and to minimize numerical dispersion effects. See Chapter 5 for a detailed derivation and discussion of the method used.

10.1.1.2 Initialization and Input Subroutines

The subroutines that follow are the ones that initialize the program and read in the input data before any simulation is performed.

LM3INIT
This is the subroutine that controls the calling sequence to all the subroutines that initialize LM3 (see Figure 10.2). It calls the subroutines INITIR, INITSTATES, INITAGG, INITPARAM_PCB, INITSED_TRANSPT, INITCDF, INITARR, INITWTRANSP, INITLOADS, INITCONC, INITKIN, INITAIR_CONC, INITWIND_SPEED, and INIT_RESUSP_VELOCITIES.

INITFIR
This subroutine controls the calling sequence to subroutines that initialize LM3 parameters using the UT library. It calls the subroutines FIINIT, KEYSGEN, KEYSTRAN, KEYSKIN, KEYSCHEM, and FIGO. FIINIT and FIGO are part of the UT library.

KEYSGEN
This subroutine initializes LM3 FIREAD keys for general control tasks.

KEYSTRAN
This subroutine initializes LM3 FIREAD keys for water and sediment column transport parameters.

KEYSKIN
This subroutine initializes LM3 FIREAD keys for eutrophication kinetic parameters.

KEYSCHEM
This subroutine initializes LM3 FIREAD keys for atrazine and PCBs kinetic parameters.

INITSTATES
This subroutine initializes state variable indices, initializes and allocates memory to arrays containing the indices.

INITAGG
This subroutine initializes coarse (level 2) to high-resolution (level 3) geometry aggregation and de-aggregation information.

INITPARAM_CHEM
This subroutine initializes and allocates memory to arrays related to the volatilization calculation of PCBs and atrazine.

INITSED_TRANSPT
This subroutine initializes and allocates memory to arrays needed for sediment bed transport. It also reads mapping, geometry data, and time-invariant hydrodynamics.

INITCDF
This subroutine controls the calling sequence to subroutines that initialize the handling of input and output CDF files. It calls the subroutines INITCDF_ICI, INITCDF_HYD, INITCDF_OUT, INITCDF_LD, INITCDF_AIR_CONC, INITCDF_WIND, and INITCDF_RESUSP_VELOCITIES.

INITCDF_ICI
This subroutine initializes the handling of CDF files containing initial concentrations of state variables for water and sediment columns.

INITCDF_HYD
This subroutine initializes the handling of CDF files containing the hydrodynamics. It also determines the number of grid cells, the number of surficial grid cells, the number of horizontal flows, and the number of vertical flows in the model.

INITCDF_OUT
This subroutine initializes the handling of CDF files that will contain the model output. It controls the calling sequence to subroutines that initialize the CDF output of state variables, hydrodynamics, and simulation time. It calls the subroutines INITCDF_TI, INITCDF_HY, INITCDF_SH, INITCDF_EU, INITCDF_CLAY, INITCDF_PCB, INITCDF_AT, INITCDF_CL, and INITCDF_SED.

INITCDF_TI
This subroutine initializes the CDF output for timestamps; it creates the CDF metadata and CDF variables needed for all timestamps output.

INITCDF_HY
This subroutine initializes the CDF output for hydrodynamics; it creates the CDF metadata and CDF variables needed for the hydrodynamics output.

INITCDF_SH
This subroutine initializes the CDF output for shear stress related variables; it creates the CDF metadata and CDF variables needed for shear stress and wave information output.

INITCDF_EU
This subroutine initializes the CDF output for eutrophication; it creates the CDF metadata and CDF variables needed for the eutrophication output.

INITCDF_CLAY
This subroutine initializes the CDF output for clay; it creates the CDF metadata and CDF variables needed for the clay output.

INITCDF_PCB
This subroutine initializes the CDF output for PCB congeners; it creates the CDF metadata and CDF variables needed for the PCB congeners output.

INITCDF_AT
This subroutine initializes the CDF output for atrazine; it creates the CDF metadata and CDF variables needed for the atrazine output.

INITCDF_CL
This subroutine initializes the CDF output for chloride; it creates the CDF metadata and CDF variables needed for the chloride output.

INITCDF_SED
This subroutine initializes the CDF output for the state variables that are simulated in the sediment bed; it creates the CDF metadata and CDF variables needed for the sediment-related variables output.

INITCDF_LD
This subroutine initializes the handling of the CDF load files. It opens the load locations file reading in the 5 km grid cell numbers the loads are discharged to. It calls the subroutine ADDLDS, for each modeled state variable, to add available loads to an internal list of loads.

ADDLDS
This subroutine adds the load locations that are available for a given state variable to a list (a FORTRAN structure). It loops over the maximum number of load locations, selecting only the locations that have a non-zero load and adding these locations to an internal list of application uses when discharging loads into a given grid cell.

INITCDF_AIR_CONC
This subroutine initializes the handling of the CDF file containing vapor (air) concentrations of PCBs and atrazine. It opens the CDF file; it then retrieves the CDF identification numbers corresponding to update times and vapor concentrations. This subroutine is used only when simulating volatilization.

INITCDF_WIND
This subroutine initializes the handling of the CDF file containing wind speeds. It opens the CDF file; it then retrieves the CDF identification numbers corresponding to update times and wind speeds. This subroutine is used only when simulating volatilization.

INITCDF_RESUSP_VELOCITIES
This subroutine initializes the handling of the CDF file containing resuspension velocities. It opens the CDF file; it then retrieves the CDF identification numbers corresponding to update times and resuspension velocities. This subroutine is used only when simulating sediment bed transport.

INITARR
This subroutine initializes and allocates memory to global arrays.

INITWTRANS
This subroutine initializes and allocates memory to arrays related to the water column transport calculation. It also initializes geometry and mapping information, reads in time-invariant hydrodynamics, and assigns user-specified settling velocities to corresponding arrays.

INITLOADS
This subroutine initializes the handling of loads in LM3. It reads in the initial loads and the times of the next update for the loads.

INITCONC
This subroutine reads in the initial state variable concentrations for the water and sediment columns. It scales initial concentrations according to user-defined scaling factors. It prints to standard output user-defined maximum allowed concentrations and stores those values in

arrays. It calculates porosities in the sediment bed; it calls the subroutine FRACTION_PCB to partition the initial PCB congener total concentration among carbon-based solids; it calls the subroutine DEAGGREG to de-aggregate initial conditions specified on the low-resolution (see Section 1.2) segmentation to the high-resolution grid (see Section 5.1); it calculates the initial total mass in the lake for each state variable.

INITKIN
This subroutine initializes and allocates memory to arrays related to the eutrophication kinetics. It also prints to standard output user-input eutrophication kinetics parameters.

INITAIR_CONC
This subroutine initializes the handling of vapor-phase chemical concentrations. It reads in the initial vapor-phase concentrations and the times of the next update for these values. This subroutine is used only when simulating volatilization of PCBs or atrazine.

INITWIND_SPEED
This subroutine initializes the handling of wind speed. It reads in the initial wind speed values and the times of the next update for these values. This subroutine is used only when simulating volatilization of PCBs or atrazine.

INIT_RESUSP_VELOCITIES
This subroutine initializes the handling of resuspension velocities. It reads in the initial resuspension velocities and the times of the next update for these values. This subroutine is used only when simulating sediment bed transport.

10.1.1.3 Simulation Execution and Output Subroutines

LM3BAT
This is the subroutine that controls the calling sequence to all the subroutines that carry out the model simulation and writing of the model output (see Figure 10.3). It calls the subroutines OUTPUT, HYDRO, TSTEPH, DERIV, and UPDATES.

OUTPUT
This subroutine controls the calling sequence to the subroutines that handle the writing of model results to CDF output files. It calls the subroutines OUTPUT_TI, OUTPUT_HY, OUTPUT_SH, OUTPUT_EU, OUTPUT_CLAY, OUTPUT_PCB, OUTPUT_AT, OUTPUT_CL, OUTPUT_SED, and OUTPUT_L2PROD.

OUTPUT_TI
This subroutine writes three timestamps to output: Julian days, calendar date, and simulation time in seconds.

OUTPUT_HY
This subroutine writes hydrodynamics information to output: cell volumes, temperature, currents, and dispersion coefficients.

OUTPUT_SH
This subroutine writes parameters related to the shear stress calculation to output: shear stress, wave height, wave period, and wave direction.

OUTPUT_EU
This subroutine writes eutrophication simulation results to output. These results consist of concentrations, growth limitation factors (nutrient, light, and temperature), eutrophication state variable masses, cumulative loads, settled-out masses, and settling rates.

OUTPUT_CLAY
This subroutine writes clay simulation results to output. These results consist of concentrations, cumulative loads, masses, settled-out masses, and settling rates.

OUTPUT_PCB
This subroutine writes PCB water column simulation results to output. These results consist of concentrations (dissolved, particulate, and total), cumulative loads, total masses, settled-out masses, settling rates, and volatilization gains and losses.

OUTPUT_AT
This subroutine writes atrazine simulation results to output. These results consist of concentrations, cumulative loads, total masses, decayed masses, lost masses (through outflows), and volatilization gains and losses.

OUTPUT_CL
This subroutine writes chloride simulation results to output. These results consist of concentrations, cumulative loads, and total masses.

OUTPUT_SED
This subroutine writes sediment bed simulation results to output. These results consist of concentrations corresponding to PCBs, clay, and carbon-based solids (only refractory organic carbon (ROC) as of this writing.

OUTPUT_L2PROD
This subroutine writes phytoplankton productivity values to output. Values are aggregated on a coarser grid; these values are then exported for use in the application known as LM2-PCB.

HYDRO
This subroutine reads in the hydrodynamics input data. The data consists of vertically-integrated water volumes, vertically-integrated flow face areas, horizontal flows, vertical flows, horizontal diffusions, vertical diffusions, water temperature, and short-wave radiation. If shear stress calculations are performed, wave information is input as well. Other tasks performed by this subroutine are: calculation of the horizontal components of currents, calculation of average diffusion coefficients for grid cells, determination of flow direction, turning on/off diffusion and/or flows, calculation of grid cell volumes, and calculation of flow face areas. This subroutine also re-sets the hydrodynamic and loads pointers for simulations of more than two years.

TSTEPH
This subroutine calculates the program time step. The time step is derived from the Courant stability constraint on horizontal transport.

DERIV
This subroutine controls the calling sequence to the subroutines that calculate the mass derivatives, due to kinetic processes, of each modeled state variable. It calls the subroutines

VSETCAL, SHEAR_STRESS, SETTLING, CHEMICALS, EXCHANGE, WTRANSPT, SED_TRANSPT, LOADS, and EUTRO.

VSETCAL
This subroutine assigns user-defined settling velocities to an internal array (vset) that in turn is used by the transport subroutine (WTRANSPT). This subroutine also serves as a placeholder to include calculations to make settling velocities space-dependent and to include flocculation algorithms.

SHEAR_STRESS
This subroutine calculates bottom shear stresses. It calls the subroutine BOTTOM_VEL.

BOTTOM_VEL
This subroutine calculates the bottom orbital velocity due to wave action in the water column. The bottom orbital velocity is used for the calculation of shear stress.

SETTLING
This subroutine calculates the bottom layer settling flux of each settling state variable. Mass derivative of state variables gets updated only when performing no sediment transport.

CHEMICALS
This subroutine controls the calling sequence to subroutines performing partitioning of PCBs and volatilization of PCBS and/or atrazine. It calls the subroutines FRACTION_PCB, AIR_CONC_CHEM, WIND_SPEED, and VOLAT. This subroutine also calculates a lumped first-order decay rate for atrazine.

FRACTION_PCB
This subroutine computes the fraction of a given PCB congener in the dissolved phase, dissolved organic carbon-sorbed phase, and particulate organic carbon-sorbed phase.

AIR_CONC_CHEM
This subroutine reads in the vapor-phase concentrations of PCBs and/or atrazine and the times of the next updates.

WIND_SPEED
This subroutine reads in wind speeds for each of the surficial 5 km grid cells and the time of the next update.

VOLAT
This subroutine computes the volatilization contribution to the mass derivative of PCBs and/or atrazine. This subroutine is called only when simulating volatilization.

EXCHANGE
This subroutine exchanges resuspension and settling fluxes of sediments between the water column and the sediment column. It calculates the porosity of each sediment grid cell. It calls the subroutine RESUSPENSION_VELOCITIES to determine resuspension velocities.

RESUSPENSION_VELOCITIES
This subroutine reads in resuspension velocities from a CDF input file.

WTRANSPT
This subroutine calculates the mass derivative of each state variable due to transport processes in the water column. The transport calculation is based on the UQ algorithm. This subroutine calls another subroutine, TSTEPV, only when settling the Crank-Nicholson parameter to zero. TSTEPV is supposed to calculate a vertical time step; however, as of this writing, TSTEPV does not work properly and therefore, it is not used.

SED_TRANSPT
This subroutine calculates the mass derivative of each state variable modeled in the sediment bed, due to transport processes in the sediment bed column. The sediment bed transport calculation is based on the UQ algorithm.

LOADS
This subroutine calculates the load contributions to the mass derivative of each state variable.

EUTRO
This subroutine controls the calling sequence to the subroutines that perform eutrophication kinetics. It calls the subroutines ZOO, DIATOMS, GREENS, CARBON, PHOSPH, NITROG, and SILICA.

ZOO
This subroutine performs zooplankton kinetic calculations. It accounts for zooplankton growth from consumption of phytoplankton and zooplankton mortality.

DIATOMS
This subroutine performs diatom kinetic calculations. It accounts for metabolism and light, temperature, and nutrient limitations.

GREENS
This subroutine performs non-diatom algae kinetic calculations. It accounts for metabolism and light, temperature, and nutrient limitations.

CARBON
This subroutine performs kinetic calculations of carbon. It accounts for sources and sinks of carbon such as phytoplankton metabolism and predation, zooplankton mortality, and mineralization.

PHOSPH
This subroutine performs kinetic calculations of phosphorus. It accounts for sources and sinks of phosphorus.

NITROG
This subroutine performs kinetic calculations of nitrogen. It accounts for sources and sinks of nitrogen.

SILICA
This subroutine performs kinetic calculations of silica. It accounts for sources and sinks of silica.

UPDATES
This subroutine performs updates of export masses, state variable concentrations, segment volumes, and simulation time. It prints out error and warning messages for cases when concentrations are out of range. This subroutine checks for the occurrence of negative concentrations; if negative values are observed, an algorithm is invoked that will fix the problem.

10.1.1.4 Finalizing Subroutines

LM3FIN
This subroutine handles the finishing stages of the simulation. It calls the subroutines INTFIN and OUTPUT. It closes all CDF output files and it then prints the timestamp corresponding to the end of the program's execution.

INTFIN
This subroutine is empty as of this writing. It serves only as a placeholder for any future addition of time integration calculations.

OUTPUT
This subroutine is described in Section 10.1.1.3. It is called one more time to write the last record of the model output.

10.1.1.5 Modules

Modules are separately compiled program units that contain the definitions and initial values of the data that users wish to share among program subroutines and functions. Modules were first introduced in FORTRAN 90; they have become the standard way of sharing large amounts of data among program units. The list of modules that follow represents the ones that are currently found in LM3.

AGG
This module declares arrays and structures used to provide aggregation and de-aggregation data between the coarse grid and the high-resolution grid.

CDFBLK_AT
This module declares and initializes CDF variables and constants used to specify atrazine information.

CDFBLK_CLAY
This module declares and initializes CDF variables and constants used to specify clay information.

CDFBLK_CL
This module declares and initializes CDF variables and constants used to specify chloride information.

CDFBLK_EU
This module declares and initializes CDF variables and constants used to specify eutrophication information.

CDFBLK
This module declares and initializes global CDF variables and constants used by LM3.

CDFBLK_HY
This module declares and initializes CDF variables and constants used to specify hydrodynamic information.

CDFBLK_PCB
This module declares and initializes CDF variables and constants used to specify PCB information.

CDFBLK_RESUSP_VELOCITIES
This module declares and initializes CDF variables and constants used to specify resuspension velocities information.

CDFBLK_SED
This module declares and initializes CDF variables and constants used to specify sediment bed information.

CDFBLK_SH
This module declares and initializes CDF variables and constants used to specify shear stress information.

CDFBLK_TI
This module declares and initializes CDF variables and constants used to specify timestamp information.

CDFBLK_WIND
This module declares and initializes CDF variables and constants used to specify wind speed information.

CDFPARM
This module declares and initializes CDF parameters.

CLIM
This module declares and initializes concentration limit parameters.

CONC
This module declares arrays and structures used to store concentrations and update times.

CONST
This module declares and initializes parameters used for calculating shear stress.

DIMS
This module declares and initializes parameters used to specify dimensions for arrays. For example: the number of PCB phases in the water and sediment columns, the maximum length of subroutine names, and the number of sediment layers.

DMASS
This module declares the arrays that store mass derivatives and mass balances of state variables. There is no initialization of the arrays in this module because all of them are declared as ALLOCATABLE.

ENVIRON
This module declares and initializes variables for the molecular weight and the decay rate of atrazine. It also declares a few arrays related to chemical kinetic processes that are not currently simulated in LM3.

EUT
This module declares and initializes all the parameters associated with the eutrophication kinetics. It also declares, but does not initialize, all the arrays associated with the eutrophication kinetics. These arrays are used to store information such as light, temperature, and nutrient limitation factors; production and predation rates; and temperature corrections for hydrolysis and mineralization.

FIBLK
This module declares the array, USRSTR, used to store FIREAD key definitions. See Appendix B for a discussion of the FIREAD utility.

FLAGS
This module declares and initializes program flags that turn on/off the simulation of kinetics or transport processes; it also turns on/off the simulation of PCB congeners, atrazine, and chloride.

FNAMES
This module declares and initializes variables that store the names of CDF input files.

LDBLK
This module declares arrays and structures that are used to store loads, load cell locations, and load scaling factors. It also declares and initializes variables associated with multi-year simulations.

LUN
This module stores logical units used when opening ASCII input files.

PARINT
This module is a generic interface block for LOGPAR. LOGPAR is actually three FORTRAN subroutines: LOGPAR_C, LOGPAR_I, and LOGPAR_R. The trailing letters C, I, and R stand for character, integer, and real, respectively. The right subroutine is invoked based on the FORTRAN type of the parameter being passed to LOGPAR. LOGPAR is used only to print input parameters to standard output. It is called from various initializing subroutines (see Figure 10.2).

PCB
This module declares and initializes variables containing the molecular weights, entropies, enthalpies, and partition coefficients of all of the PCB congeners that can be simulated in LM3.

SCRATCH
This module declares arrays used to store scratch memory.

SED_TRANSP_BLK
This module declares and initializes variables and arrays used for the UQ sediment column transport scheme.

STATES
This module declares all of the indices used to uniquely identify state variables.

TIMES
This module declares and initializes variables dealing with simulation times.

WAVE_CURRENT
This module defines arrays and variables that store information used for the shear stress calculation. Some of the information is related to waves and currents.

WTMP
This module declares arrays used to store water temperature.

WTRBLK
This module declares and initializes variables and arrays used for the UQ water column transport scheme.

10.1.1.6 Generic Subroutines

LM3 makes use of the FORTRAN 90 feature of generic subroutines. A generic subroutine is a subroutine that can operate properly with many different types of input data. Generic subroutines are implemented using a generic interface block. In LM3, only one generic subroutine, LOGPAR, was implemented in the module PARINT (see previous section). LOGPAR consists of three specific subroutines: LOGPAR_C, LOGPAR_I, and LOGPAR_R; each one is described below. Every time that LOGPAR is called, the compiler checks the arguments supplied to it and invokes the specific subroutine for those arguments.

LOGPAR_C
This subroutine prints parameters of type CHARACTER to standard output.

LOGPAR_I
This subroutine prints parameters of type INTEGER to standard output.

LOGPAR_R
This subroutine prints parameters of type REAL to standard output.

10.1.1.7 INCLUDE Files

LM3 makes use of INCLUDE files. INCLUDE files are not strictly speaking a feature of FORTRAN, but rather of C. These types of files are handled by the cpp before any compilation is done. LM3 contains only one INCLUDE file; its name is platform.h and is described below.
PLATFORM.H

This file defines status flags for handling machine-dependent code and selective inclusion/exclusion of source code routines for debugging.

10.2 LM3 Source and Application Management

Development of the LM3 calculation took place over five years and included a number of program modifications by different users. The source code management tool RCS is employed to track revision information and to assist with program documentation during the course of development. An "official" copy of the program is kept under the administration of a source code manager whose user name is designated "wmelende" as of this writing. All members of the modeling group have read access to all LM3 model sources; however, only designated modelers have write access to "repositories" or source code directories where the model code actually resides. The LM3 model was developed under this environment. Model revisions are periodically updated, and new source modifications are added to the LM3 repository; at the United States Environmental Protection Agency (USEPA), Large Lakes Research Station (LLRS), this directory is found under the Linux path /mnt/model/llrs_model/dev/LM3/RCS. Under normal circumstances, revisions to the LM3 calculations are made by the developers and the corresponding program executables are then generated individually by the user in his home directory. In other words, once a particular version of LM3 has been revised by the developer, the user obtains a copy of the source code from RCS and compiles it to generate the corresponding executable. This executable resides in the respective user's home directory under the Linux system. It is the responsibility of the user to make sure that the latest source code revision as well as the appropriate corresponding executable of the LM3 branch are used. The user determines which LM3 branch to use depending on what kind of simulation is to be run.

The procedures for compiling and linking the LM3 program are largely automated. The Linux *make* procedure was modified to retrieve the source code from the LM3 source code repository, compile the program, and link/load the executable. If the LM3 source is already available to the user, the application can be built using either of the commands below.

```
make
make nodebug
```

This produces an optimized version of the program with the name lm3-<version_number>. A binary containing additional symbol table and traceback information can be created using the command:

```
make debug
```

Program binaries created using this command disable compiler optimizations and enable array boundary violation checking. While useful for debugging, these features greatly slow the execution of the program and should not be used unless debugging is desired.

If program source is not available to the user, it must first be recovered from the LM3 source repository. The Linux commands for creating the application binary under these circumstances are:

```
ln -s /mnt/model/llrs_model/dev/LM3/RCS
co -r<version> Makefile
```

```
make source
make
```

The command "make source" can be omitted. In this case, source files will be added to the user's default directory from the RCS repository as the LM3 application is being built. As in the previous case, a version of the LM3 binary appropriate for debugging can be generated by using the command "make debug" instead of simply "make." Sources for the LM3 Makefile are appended to Section 10.3.

10.3 LM3 Support Libraries

The LM3 calculation is not a stand-alone, but it makes reference to a number of C and FORTRAN functions and subroutines that are neither included with the source distribution nor a part of the Linux runtime environment. At this time, three such libraries are required to build the application: the LLRS UT library, a subset of the SNIPPETS software library, and the CDF library. Sources for the UT library can be obtained by request from LLRS. The SNIPPETS[1] and CDF[2] library sources are publicly available and can be downloaded from the Internet. The procedure for linking these packages to the LM3 application is demonstrated in the LM3 Linux Makefile shown below.

```
# Makefile for program {LM3}.
#
# $Id: Makefile, v 3.2.15.24 2006/04/27  19:21:49 wam Exp $
# $Locker:  $
# Created:  10/15/95  M. Settles
#
# Revised:  04/27/06  W. Melendez :     Expanded the comment line
#                                       "# DO NOT DELETE THIS LINE --
#                                       make depend depends on it"
# Revised:  03/24/05  W. Melendez :     Updated make dependencies.
# Revised:  01/28/05  W. Melendez :     Updated make dependencies.
# Revised:  08/18/04  W. Melendez :     Updated make dependencies.
# Revised:  07/14/04  W. Melendez :     Updated Fortran compiler pathname on Linux.
# Revised:  07/02/04  W. Melendez :     Updated make dependencies.
# Revised:  08/29/03  W. Melendez :     Renamed cdfblk_resusp_fluxes.F,
#                                       initcdf_resusp_fluxes.F,
#                                       init_resusp_fluxes.F, and
#                                       resuspension_fluxes.F to
#                                       cdfblk_resusp_velocities.F,
```

---

[1]The required sources jdnl_l.c and sunriset.c can be obtained from the SNIPPETS homepage at http://www.brokersys.com/snippets/.

[2]CDF is a product of the National Space Science Data Center (NSSDC) at National Aeronautics and Space Administration (NASA). The CDF libraries can be downloaded from their web page at http://nssdc.gsfc.nasa.gov/cdf/cdf_home.html.

```
#                                       initcdf_resusp_velocities.F,
#                                       init_resusp_velocities.F, and
#                                       resuspension_velocities.F.
# Revised:  08/19/03  W. Melendez :     Added four new subroutines:
#                                       cdfblk_resusp_fluxes.F,
#                                       initcdf_resusp_fluxes.F,
#                                       init_resusp_fluxes.F, and
#                                       resuspension_fluxes.F.
#                                       Updated make dependencies.
# Revised:  06/10/03  W. Melendez :     Updated make dependencies.
# Revised:  06/09/03  W. Melendez :     Removed sed_bed.o dependency.
# Revised:  06/09/03  W. Melendez :     Updated make dependencies.
# Revised:  06/03/03  W. Melendez :     Updated make dependencies.
# Revised:  06/03/03  W. Melendez :     Updated make dependencies.
# Revised:  04/21/03  W. Melendez :     Added --staticlink Lahey/Linux compiler option.
# Revised:  04/07/03  W. Melendez :     Removed -u flag from "make cppsource".
# Revised:  04/07/03  W. Melendez :     Updated dependencies.
# Revised:  03/12/03  W. Melendez :     Removed option --ng from the macro NODEBUGFLAGS_LINUX.
# Revised:  02/28/03  W. Melendez :     Updated make dependencies.
#                                       Added new subroutines.
#                                       Allow for the compilation of LM3 in both Unix and Linux.
# Revised:  12/26/02  W. Melendez :     Added new module --> pcb.F.
#                                       Added wind subroutines --> cdfblk_wind.F, initcdf_wind.F,
#                                       initwind_speed.F, and wind_speed.F.
#                                       Added PCB air concentration files --> initcdf_air_conc.F,
#                                       initair_conc.F, and air_conc_pcb.F.
#                                       Added routine fraction_pcb.F.
# Revised:  11/20/02  W. Melendez :     Added three new PCB related routines:  cdfblk_pcb.F,
#                                       initcdf_pcb.F, and initparam_chem.F.
# Revised:  11/14/02  W. Melendez :     cdfblk_sa.F --> cdfblk_clay.F,
#                                       initcdf_sa.F --> initcdf_clay.F,
#                                       output_sa.F --> output_clay.F.
# Revised:  06/17/02  W. Melendez :     Removed the routines burial.F, initsedbed.F,
#                                       res_flux.F, and sed_bed.F.
#                                       Added four new routines:  initsed_transpt.F,
#                                       sed_transpt.F, sed_transp_bik.F, and exchange.F.
# Revised:  05/21/02  W. Melendez :     --> V-3.2.15.
# Revised:  08/10/01  W. Melendez :     --> v-3.2.13.
# Revised:  02/12/01  W. Melendez :     Removed the -g option from the NODEBUGFLAGS macro
#                                       definition.
# Revised:  11/27/00  W. Melendez :     --> v-3.2.12.
# Revised:  10/25/00  W. Melendez :     Added source and object for cdfblk_ci, initcdf_ci(), and
#                                       output_ci().
#                                       Updated make dependencies.
# Revised:  06/29/00  M. Settles :      --> v-3.2.11.
# Revised:  06/09/00  M. Settles :      Added source and object for output_lm2prod().
# Revised:  06/02/00  W. Melendez :     Copy of v-3.2.8 for LM3 v-3.2.9.
# Revised:  05/13/00  M. Settles :      Update make dependencies.
# Revised:  05/11/00  M. Settles :      Update make dependencies.
# Revised:  12/29/99  M. Settles :      Change name from IPX --> LM3.
# Revised:  12/29/99  M. Settles :      Copy from Makefile for IPX v-3.2.8.
#
#-----------------------------------------------------------------------------------------------------------------------------------
#
```

```
# Utility library directories and names.

# Version number
VERSION = 3.2.15

# Specify the RCS branch to be used during checkout.
CO = co -q -r$(VERSION)

# The execution shell.
SHELL = /bin/sh

# Linux copy.
CP = /bin/cp

# Linux rename
MV = /bin/mv

# The C-preprocessor.
# Some machines (SunOS) put the pre-processor in strange places.
# Just find it and stick the directory in your path.
CPP = cpp -P -C -I.

# How to delete things.
RM = /bin/rm -f

#How to update the file access dates.
TOUCH = touch -c

# The current directory.
CURRENTDIR = -I

# OSF1 compiler flags.
DEBUG = O0 - g -check bounds - ladebug
NODEBUGFLAGS = -O -free $(CURRENTDIR)
DEBUGFLAGS = $(DEBUG) -DDODEBUG -free -align dcommons $(CURRENTDIR)

# Linux compiler flags.
DEBUB_LINUX = -O0 -g --chk
DEBUGFLAGS_LINUX = $(DEBUG_LINUX) --staticlink --nfix --ml cdeci $(CURRENTDIR)
NODEBUGFLAGS_LINUX = -O --staticlink --trap --nchk --nfix --ml cdeci $(CURRENTDIR)
# name of the complied program binaries.

# Program LM3 binary.
LM3BIN = lm3-$(VERSION)

#Header File.
HEADER = platform.h
# names of the source files.
SOURCE = addids.F agg.F aggreg.F air_conc_chem.F bottom_vel.F carbon.F cdfblk.F
cdfblk_at.F cdfblk_ci.F cdfblk_clay.F cdfblk_eu.F cdfblk_hy.F cdfblk_pcb.F cdfblk_resusp_velocities.F
cdfblk_sed.F cdfblk_sh.F cdfblk_ti.F cdfblk_wind.F cdfparm.F chemicals.F clim.F conc.F const.F
deaggreg.F deriv.F diatoms.F dims.F dmass.F environ.F eut.F eutro.F exchange.F fiblk.F ficmax.F
fido.F fifile.F fiscale.F fiuser.F flags.F fnames.F fraction_pcb.F greens.F hydro.F initagg.F initair_conc.F
initarr.F initcdf.F initcdf_air_conc.F initcdf_at.F initcdf_ci.F initcdf_clay.F initcdf_eu.F initcdf_hy.F
```

```
initcdf_hyd.F initcdf_ici.F initcdf_id.F initcdf_out.F initicdf_pcb.F initcdf_resusp_velocities.F
initcdf_sed.F initcdf_sh.F initcdf_ti.F initcdf_wind.F initconc.F initeutro.F initfir.F initkin.F initloads.F
initparam_chem.F init_resusp_velocities.F initsed_transpt.F initstates.F initwind_speed.F initwtrans.F
initfin.F lm3.F lm3bat.F lm3fin.F lm3init.F keyschem.F keysgen.F keyskin.F keystran.F loads.F ldblk.F
logpar_c.F logpar_i.F logpar_r.F lun.F nitrog.F output.F output_at.F output_ci.F output_clay.F
output_eu.F output_hy.F output_12prod.F output_pcb.F output_sed.F output_sh.F output_ti.F parint.F
pcb.F phosph.F resuspension_velocities.F scratch.F sed_transpt.F sed_transp_blk.F settling.F
shear_stress.F silica.F states.F times.F tsteph.F tstepv.F updates.F volat.F vsetcal.F wave_current.F
wind_speed.F wtmp.F wtranspt.F wtrblk.F zoo.F

# names of the object files.
OBJECTS = addids.o agg.o aggreg.o air_conc_chem.o bottom_vel.o carbon.o cdfblk.o
cdfblk_at.o cdfblk_ci.o cdfblk_clay.o cdfblk_eu.o cdfblk_hy.o cdfblk_pcb.o cdfblk_resusp_velocities.o
cdfblk_sed.o cdfblk_sh.o cdfblk_ti.o cdfblk_wind.o cdfparm.o chemicals.o clim.o conc.o const.o
deaggreg.o deriv.o diatoms.o dims.o dmass.o environ.o eut.o eutro.o exchange.o fiblk.o ficmax.o
fido.o fifile.o fiscale.o fiuser.o flags.o fnames.o fraction_pcb.o greens.o hydro.o initagg.o initair_conc.o
initarr.o initcdf.o initcdf_air_conc.o initcdf_at.o initcdf_ci.o initcdf_clay.o initcdf_eu.o initcdf_hy.o
initcdf_hyd.o initcdf_ici.o initcdf_id.o initcdf_out.o initicdf_pcb.o initcdf_resusp_velocities.o
initcdf_sed.o initcdf_sh.o initcdf_ti.o initcdf_wind.o initconc.o initeutro.o initfir.o initkin.o initloads.o
initparam_chem.o init_resusp_velocities.o initsed_transpt.o initstates.o initwind_speed.o initwtrans.o
initfin.o lm3.o lm3bat.o lm3fin.o lm3init.o keyschem.o keysgen.o keyskin.o keystran.o loads.o ldblk.o
logpar_c.o logpar_i.o logpar_r.o lun.o nitrog.o output.o output_at.o output_ci.o output_clay.o
output_eu.o output_hy.o output_12prod.o output_pcb.o output_sed.o output_sh.o output_ti.o parint.o
pcb.o phosph.o resuspension_velocities.o scratch.o sed_transpt.o sed_transp_blk.o settling.o
shear_stress.o silica.o states.o times.o tsteph.o tstepv.o updates.o volat.o vsetcal.o wave_current.o
wind_speed.o wtmp.o wtranspt.o wtrblk.o zoo.o

# names of modules
MODULES = agg.mod cdfblk.mod  cdfblk_at.mod  cdfblk_ci.mod  cdfblk_clay.mod  cdfblk_eu.mod
cdfblk_hy.mod  cdfblk_pcb.mod  cdfblk_resusp_velocities.mod  cdfblk_sed.mod  cdfblk_sh.mod
cdfblk_ti.mod  cdfblk_wind.mod  cdfparm.mod  clim.mod  conc.mod  const.mod  dims.mod  dmass.mod
environ.mod  eut.mod  fiblk.mod  flags.mod  fnames.mod  job.mod  ldblk.mod  lun.mod  parint.mod
pcb.mod  scratch.mod  sed_transp_blk.mod  states.mod  statistics.mod  structure.mod  times.mod
wave_current.mod  wtmp.mod  wtrblk.mod

# Make everything
all:
	@case 'uname' in   \
	"OSF1")  make FFLAGS=$(NODEBUGFLAGS)" $(LM3BIN)   \
		"CDFLIBDIR=-L/home/llrs_model/local/cdf/lib"   \
		CDFLIB=-lcdf"   \
		"SNIPPETSLIBDIR=-L/home/llrs_model/dev/SNIPPETS/lib"   \
		"SNIPPETSLIB=-lsnippets-1.1"   \
		"UTLIBDIR=-L/home/llrs_model/dev/UT/lib"   \
		"UTLIB=--lut-1.3.2"  \
		"MAKE=/bin/make"   \
		"MAKEDEPEND=/usr/bin/X11/makedepend"   \
		"FC=/bin/f90";   \
		;;   \
	"Linux")  make FFLAGS="$(NODEBUGFLAGS_LINUX)"$(LM3BIN)   \
		"CDFLIBDIR=-L/usr/local/lib/CDF/lib"   \
		"CDFLIB=-lcdf"   \
		"SNIPPETSLIBDIR=-L/usr/local/lib/SNIPPETS/lib"   \
		"SNIPPETSLIB=-lsnippets-1.1"   \
```

```
                "UTLIBDIR=-L/usr/local/lib/UT/lib"   \
                UTLIB=-lut-1.3.2"   \
                "MAKE=/bin/make"   \
                "MAKEDEPEND=/usr/bin/X11/makedepend"   \
                "FC=/usr/local/lf9562/bin/lf95";   \
                ;;   \
                esac
debug:
        @case 'uname' in   \
        "OSF1")  make FFLAGS="$(DEBUGFLAGS)" $(LM3BIN)
                "CDFLIBDIR=-L/home/llrs_model/local/cdf/lib"   \
                "CDFLIB=-lcdf"   \
                "SNIPPETSLIBDIR=-L/home/llrs_model/dev/SNIPPETS/lib"   \
                "SNIPPETSLIB=-lsnippets-1.1"   \
                "UTLIBDIR=-L/home/llrs_model/dev/UTlib"   \
                "UTLIB=-lu-1.3.2"   \
                "MAKE=/bin/make"   \
                "FC=/bin/f90";   \
                ;;   \
        "Linux)  make FFLAGS="$(DEBUGFLAGS_LINUX)" $(LM3BIN)   \
                "CDFLIBDIR=-L/usr/local/lib/CDF/lib"   \
                "CDFLIB=-lcdf"   \
                "SNIPPETSLIBDIR=-L/usr/local/lib/SNIPPETS/lib"   \
                "SNIPPETS=-lsnippets-1.1"   \
                "UTLIBDIR=-L/usr/local/lib/UT/lib"   \
                "UTLIB=-lut-1.3.2"   \
                "MAKE=/bin/make"   \
                "FC=/usr/local/lf9562/bin/lf95";   \
                ;;   \
                esac
modebug:
        make all

# Clean up
clean:; $(RM) $(LM3BIN) $(OBJECTS) $(MODULES)
# Update time stamps
touch:;         $(TOUCH) $(SOURCE) $(HEADER)

# Link the executable
$(LM3BIN)       $(OBJECTS)
                $(FC)  $(FFLAGS)  $(PROFILE)  -o  $@  $(OBJECTS)  $(UTLIBDIR)
                $(SNIPPETSLIBDIR)  $(UTLIB)  $(SNIPPETSLIB)  $(CDFLIBDIR)  $(CDFLIB)

depend:         $(SOURCE)  $(HEADER)
                $(MAKEDEPEND)  -f$(MAKEFILE)  $(CURRENTDIR)  $(SOURCE)  $(HEADER)
# Compile FORTRAN source
F.o:
                $(FC)  $(FFLAGS)  $(SPEED)  -free  -c  $<

# Source files.
source:
        @case 'uname' in   \
        "OSF1")  make cppsource   \
                ;;   \
```

```
        "Linux")  make cppsource   \
                ;;   \
                esac

cppsource:
                $(CO)  $(SOURCE)  $(HEADER)

.SUFFIXES:  .o  .mod  .F
.PRECIOUS:  $(LM3BIN)

# DO NOT DELETE THIS LINE - make depend depends on it
```

CHAPTER 11

DATA STRUCTURES

Numeric data processed by the LM3 calculation includes input parameter values, and a number of output variables which are typically non-scalar and can possess detailed time and space dependence. The Common Data Format (CDF) (Treinish and Gough, 1987; Treinish and Goucher, 1988; Treinish, 1991; Brown *et al.*, 1993) was developed by the National Space Science Data Center (NSSDC) as a scientific data management tool. Its application to the Lake Michigan Mass Balance Project (LMMBP) simulation work is to provide shared data format linking simulations and analysis tools, field data and analysis tools, and different programs comprising the analysis effect.

The CDF data representations (National Aeronautics and Space Administration, 1996a,b,c) employ two principle data types. The rVariable format, described in Figure 11.1, requires that all contributing variables have the same dimension for each record. Using this format, variables with different array sizes or shapes are accommodated by explicitly specifying, by Boolean flag, the variability of a particular dimension, i.e., whether or not a particular variable varied in one of its indices. The more recently introduced zVariable data representation, portrayed in Figure 11.2, provides for variables that were specified without these limitations. Details concerning code implementations of these data types and access to the data can be found in the CDF reference manuals (National Aeronautics and Space Administration, 1996a,b,c). Examples can be gleaned from the LM3 source files initcdf_*.F, and output_*; however, these are verbose and developers are referred first to the CDF documentation.

The rVariable format is currently used to represent initial conditions. The record degree of freedom corresponds to the segment number of the level 2 geometry in the water column case; while it corresponds to the sediment layer number in the sediment column case. Each simulated state is represented by a unique CDF rVariable. All remaining input and output variables are presumed to be both time- and space-dependent and are represented using the CDF zVariable scheme. Throughout, each zVariable record corresponded to a different simulation time value. Details concerning each of these formats including variable names, array shapes, and descriptions are provided in Sections 11.1 and 11.2.

11.1 Input Data Structures

Input data passed to LM3 include run-time parameter values, hydrodynamic fields, initial conditions, and loads. Run-time input parameters are typically scalar and are frequently modified by the user to select simulated states and to modify calibration parameters; these are passed to the program in the Linux standard input stream, stdin, and are processed by the

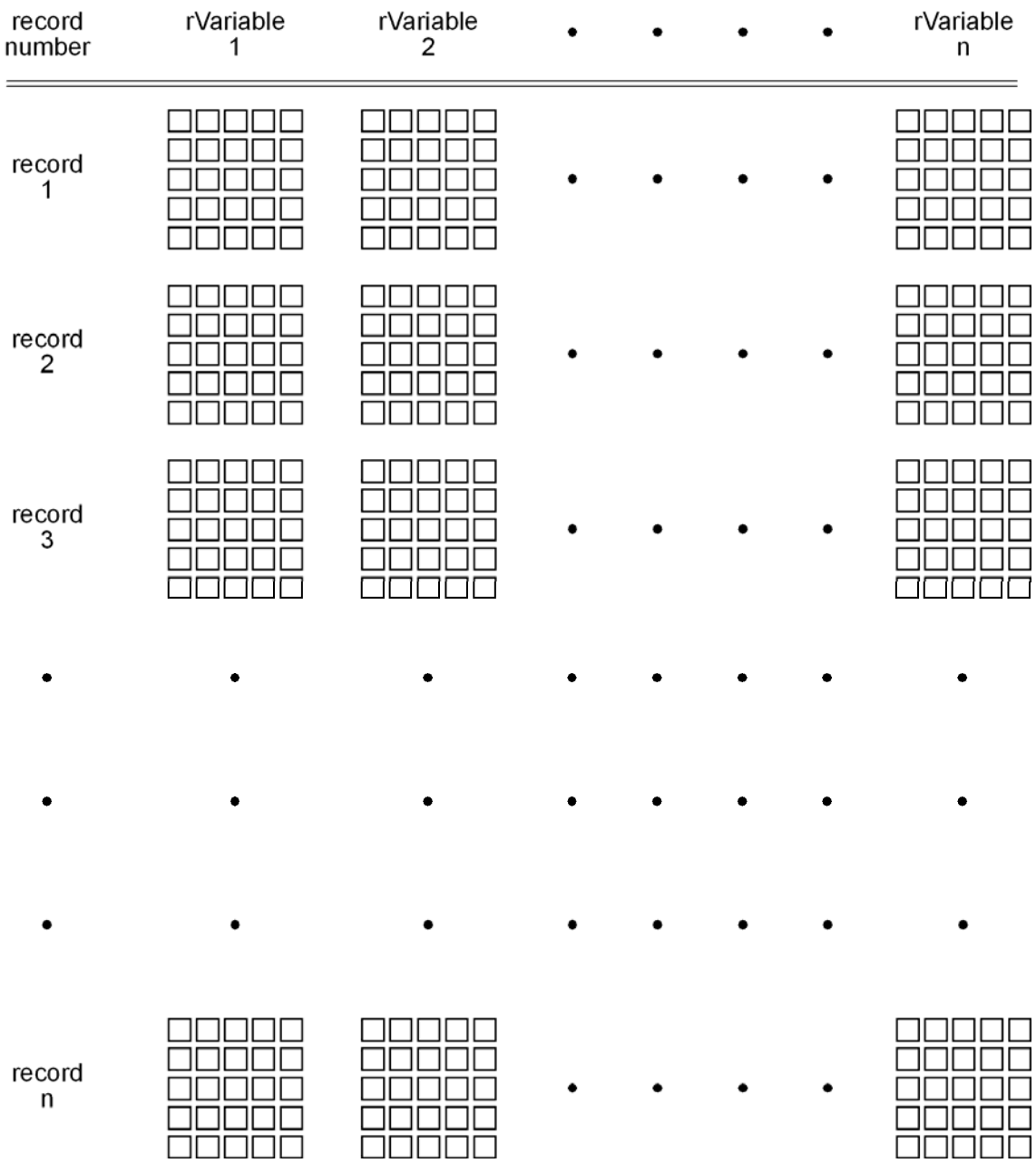


Figure 11.1. The CDF rVariable data representation scheme.

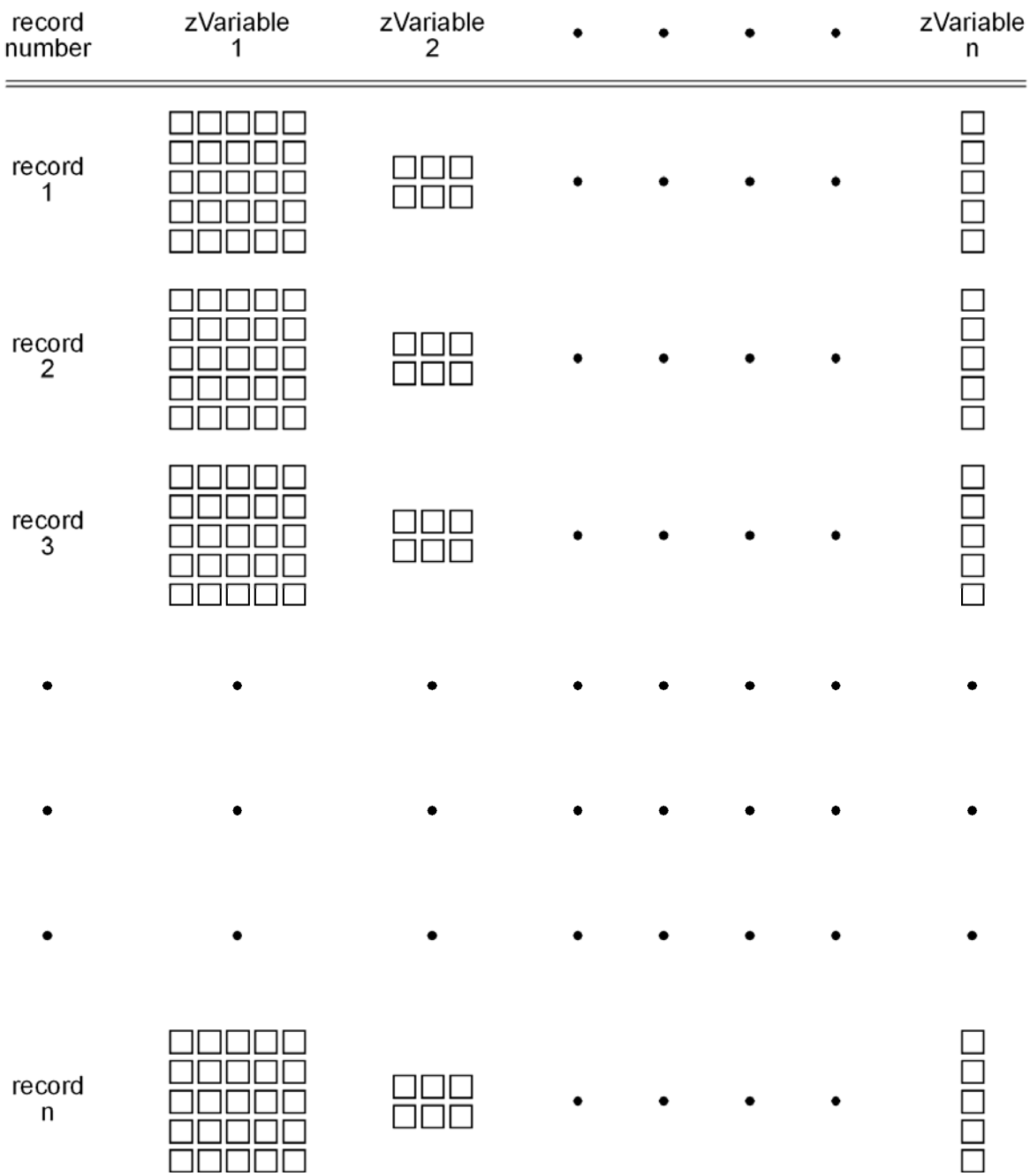


Figure 11.2 The CDF zVariable data representation scheme.

FIREAD utility as described in Chapter 3. The remaining input values include initial conditions, hydrodynamics, and load values.

11.1.1 Initial Conditions Data Structure

The initial conditions CDF defines are the initial concentrations for all modeled state variables, both used and unused, of the simulation. There are two types of initial condition files; one that specifies initial concentration values in the water column and one that specifies initial concentration values in the sediment bed. The initial conditions CDF is the only CDF used by LM3 that makes use of the rVariable format. Concentration values are specified using the meter-kilogram-second (mks) unit convention consistent with the rest of the LM3 calculation. Eutrophication solids are carbon normalized. The path for the initial input conditions file is specified using the FIREAD input key syntax.

*FILE ICI <filename>
*FILE ICI_SED <filename>

The initial conditions input file must be specified once and only once. The filename corresponding to the keyname ICI corresponds to the initial concentrations in the water column while the filename with the keyname ICI_SED corresponds to the initial concentrations in the sediment bed. When not performing sediment bed calculations, the latter does not have to be specified; otherwise both files have to be specified. Sediment bed calculations can be performed only when using the branches 3.2.15, 3.2.18, and 3.2.19 of LM3. A summary of the state variables represented in the water column initial conditions input file is provided in Tables 11.1, 11.2, and 11.3. The variables represented in the sediment column initial conditions input file are given in Tables 11.4 and 11.5. At present, the default initial condition file for the water column is /mnt/model/dsets/LM3/R0105/ici/ici.cdf, and the one for the sediment column is /mnt/model/dsets/LM3/R0105/ici/ici_sed.cdf on the Linux platform at United State Environmental Protection Agency (USEPA), Large Lakes Research Station (LLRS).

11.1.2 Hydrodynamics Data Structure

The hydrodynamics CDF provides all of the space- and time-dependent flow and dispersion information used in the LM3 transport scheme. The data contained in this CDF is not contained within a single file but uses a separate file for each zVariable. For technical reasons, the CDF libraries make use of a 32-bit addressing scheme for accessing the component data files. These files are limited to a size of $2^{(32-1)}$ bytes or roughly two gigabytes. The format of the hydrodynamics data was changed so that each of the three-dimensional variables (flow, dispersion, and temperature) is assigned a separate CDF zVariable corresponding to the different sigma layers of the geometry. Additional zVariables are assigned to the hydrodynamics CDF to contain light, heat flux, and wave information for use in the eutrophication and sediment transport calculations. The path for the hydrodynamics input file is specified using the FIREAD input key syntax.

*FILE HYDRO <filename>

The hydrodynamics input file must be specified once and only once. A summary of the variables represented in the hydrodynamics input file is provided in Table 11.6. At present, the

Table 11.1. Eutrophication Water Column Initial Conditions rVariables

r-Variable	Description	Units	Data Type	Dimension
DOC	Dissolved organic carbon concentration	$kg \cdot m^{-3}$	REAL	Scalar
DIA	Diatom carbon concentration	$kg \cdot m^{-3}$	REAL	Scalar
GRE	Non-diatom phytoplankton concentration	$kg \cdot m^{-3}$	REAL	Scalar
ZOO	Zooplankton concentration	$kg \cdot m^{-3}$	REAL	Scalar
LOC	Labile organic carbon concentration	$kg \cdot m^{-3}$	REAL	Scalar
ROC	Refractory organic carbon concentration	$kg \cdot m^{-3}$	REAL	Scalar
SRP	Soluble reactive phosphorus concentration	$kg \cdot m^{-3}$	REAL	Scalar
DOP	Dissolved organic phosphorus concentration	$kg \cdot m^{-3}$	REAL	Scalar
LOP	Labile organic phosphorus concentration	$kg \cdot m^{-3}$	REAL	Scalar
ROP	Refractory organic phosphorus concentration	$kg \cdot m^{-3}$	REAL	Scalar
NH_4	Ammonia concentration	$kg \cdot m^{-3}$	REAL	Scalar
NO_3	Nitrate concentration	$kg \cdot m^{-3}$	REAL	Scalar
DON	Dissolved organic nitrogen concentration	$kg \cdot m^{-3}$	REAL	Scalar
LON	Labile organic nitrogen concentration	$kg \cdot m^{-3}$	REAL	Scalar
RON	Refractory organic nitrogen concentration	$kg \cdot m^{-3}$	REAL	Scalar
SA	Available silica concentration	$kg \cdot m^{-3}$	REAL	Scalar
SU	Particulate biogenic silica concentration	$kg \cdot m^{-3}$	REAL	Scalar

Table 11.2. PCB Water Column Initial Conditions rVariables

rVariable	Description	Units	Data Type	Dimension
PCB_5+8	PCB congener 5+8 concentration	$kg \cdot m^{-3}$	REAL	Scalar
PCB_15+17	PCB congener 15+17 concentration	$kg \cdot m^{-3}$	REAL	Scalar
PCB_16+32	PCB congener 16+32 concentration	$kg \cdot m^{-3}$	REAL	Scalar
PCB_18	PCB congener 18 concentration	$kg \cdot m^{-3}$	REAL	Scalar
PCB_26	PCB congener 26 concentration	$kg \cdot m^{-3}$	REAL	Scalar
PCB_28+31	PCB congener 28+31 concentration	$kg \cdot m^{-3}$	REAL	Scalar
PCB_33	PCB congener 33 concentration	$kg \cdot m^{-3}$	REAL	Scalar
PCB_37+42	PCB congener 37+42 concentration	$kg \cdot m^{-3}$	REAL	Scalar
PCB_44	PCB congener 44 concentration	$kg \cdot m^{-3}$	REAL	Scalar
PCB_49	PCB congener 49 concentration	$kg \cdot m^{-3}$	REAL	Scalar
PCB_52	PCB congener 52 concentration	$kg \cdot m^{-3}$	REAL	Scalar
PCB_56+60	PCB congener 56+60 concentration	$kg \cdot m^{-3}$	REAL	Scalar
PCB_66	PCB congener 66 concentration	$kg \cdot m^{-3}$	REAL	Scalar
PCB_70+76	PCB congener 70+76 concentration	$kg \cdot m^{-3}$	REAL	Scalar
PCB_74	PCB congener 74 concentration	$kg \cdot m^{-3}$	REAL	Scalar
PCB_77+110	PCB congener 77+110 concentration	$kg \cdot m^{-3}$	REAL	Scalar
PCB_81	PCB congener 81 concentration	$kg \cdot m^{-3}$	REAL	Scalar
PCB_87	PCB congener 87 concentration	$kg \cdot m^{-3}$	REAL	Scalar
PCB_84+92	PCB congener 84+92 concentration	$kg \cdot m^{-3}$	REAL	Scalar
PCB_89	PCB congener 89 concentration	$kg \cdot m^{-3}$	REAL	Scalar
PCB_99	PCB congener 99 concentration	$kg \cdot m^{-3}$	REAL	Scalar
PCB_101	PCB congener 101 concentration	$kg \cdot m^{-3}$	REAL	Scalar
PCB_118	PCB congener 118 concentration	$kg \cdot m^{-3}$	REAL	Scalar
PCB_123+149	PCB congener 123+149 concentration	$kg \cdot m^{-3}$	REAL	Scalar

Table 11.2. PCB Water Column Initial Conditions rVariables (Continued)

rVariable	Description	Units	Data Type	Dimension
PCB_105+132+153	PCB congener 105+132+153 concentration	$kg \cdot m^{-3}$	REAL	Scalar
PCB_151	PCB congener 151 concentration	$kg \cdot m^{-3}$	REAL	Scalar
PCB_138+163	PCB congener 138+163 concentration	$kg \cdot m^{-3}$	REAL	Scalar
PCB_170+190	PCB congener 170+190 concentration	$kg \cdot m^{-3}$	REAL	Scalar
PCB_172+197	PCB congener 172+197 concentration	$kg \cdot m^{-3}$	REAL	Scalar
PCB_180	PCB congener 180 concentration	$kg \cdot m^{-3}$	REAL	Scalar
PCB_182+187	PCB congener 182+187 concentration	$kg \cdot m^{-3}$	REAL	Scalar
PCB_195+208	PCB congener 195+208 concentration	$kg \cdot m^{-3}$	REAL	Scalar
PCB_196+203	PCB congener 196+203 concentration	$kg \cdot m^{-3}$	REAL	Scalar
PCB_201	PCB congener 201 concentration	$kg \cdot m^{-3}$	REAL	Scalar
PCB_146	PCB congener 146 concentration	$kg \cdot m^{-3}$	REAL	Scalar
PCB_85	PCB congener 85 concentration	$kg \cdot m^{-3}$	REAL	Scalar

Table 11.3. Atrazine, Chloride, and Clay Water Column Initial Conditions rVariables

rVariable	Description	Units	Data Type	Dimension
ATZ	Atrazine concentration	$kg \cdot m^{-3}$	REAL	Scalar
CL	Chloride concentration	$kg \cdot m^{-3}$	REAL	Scalar
CLAY	Clay concentration	$kg \cdot m^{-3}$	REAL	Scalar

Table 11.4. Eutrophication Sediment Column Initial Conditions rVariables. Array Dimensions Refer to the Number of Surface Cells n_{sb}

rVariable	Description	Units	Data Type	Dimension
DOC	Dissolved organic carbon concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
LOC	Labile organic carbon concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
ROC	Refractory organic carbon concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
SRP	Soluble reactive phosphorus concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
DOP	Dissolved organic phosphorus concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
LOP	Labile organic phosphorus concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
ROP	Refractory organic phosphorus concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
NH_4	Ammonia concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
NO_3	Nitrate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
DON	Dissolved organic nitrogen concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
LON	Labile organic nitrogen concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
RON	Refractory organic nitrogen concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
SA	Available silica concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
SU	Particulate biogenic silica concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$

Table 11.5. PCB Sediment Column Initial Conditions rVariables. Array Dimensions Refer to the Number of Surface Cells n_{sb}

rVariable	Description	Units	Data Type	Dimension
PCB_5+8	PCB congener 5+8 concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
PCB_15+17	PCB congener 15+17 concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
PCB_16+32	PCB congener 16+32 concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
PCB_18	PCB congener 18 concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
PCB_26	PCB congener 26 concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
PCB_28+31	PCB congener 28+31 concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
PCB_33	PCB congener 33 concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
PCB_37+42	PCB congener 37+42 concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
PCB_44	PCB congener 44 concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
PCB_49	PCB congener 49 concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
PCB_52	PCB congener 52 concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
PCB_56+60	PCB congener 56+60 concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
PCB_66	PCB congener 66 concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
PCB_70+76	PCB congener 70+76 concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
PCB_74	PCB congener 74 concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
PCB_77+110	PCB congener 77+110 concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
PCB_81	PCB congener 81 concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
PCB_87	PCB congener 87 concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
PCB_84+92	PCB congener 84+92 concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
PCB_89	PCB congener 89 concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
PCB_99	PCB congener 99 concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
PCB_101	PCB congener 101 concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
PCB_118	PCB congener 118 concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
PCB_123+149	PCB congener 123+149 concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
PCB_105+132+153	PCB congener 105+132+153 concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
PCB_151	PCB congener 151 concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
PCB_138+163	PCB congener 138+163 concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
PCB_170+190	PCB congener 170+190 concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
PCB_172+197	PCB congener 172+197 concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
PCB_180	PCB congener 180 concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
PCB_182+187	PCB congener 182+187 concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
PCB_195+208	PCB congener 195+208 concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
PCB_196+203	PCB congener 196+203 concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
PCB_201	PCB congener 201 concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
PCB_146	PCB congener 146 concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
PCB_85	PCB congener 85 concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$
CLAY	Clay concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sb})$

Table 11.6. Input Hydrodynamics zVariables. Array Dimensions Refer to the Number of Layers n_ℓ, the Number of Surface Boxes n_{sb}, and the Number of Vertically Integrated Horizontal Flow Faces n_{hf2}

zVariable	Description	Units	Data Type	Dimension
TIME_JULIAN	Julian day number	Days	DOUBLE	Scalar
TIME_EPOCH	Calendar date and time	None	CDF_EPOCH	Scalar
TIME_RUNSEC	Simulation time	s	REAL	Scalar
DZ	Sigma layer height	None	REAL	$(1:n_\ell)$
BSAREA	Cell surface area	m^2	REAL	$(1:n_{sb})$
BLX	Cell length in x-coordinate	m	REAL	$(1:n_{sb})$
BLY	Cell length in y-coordinate	m	REAL	$(1:n_{sb})$
VOLUME	Cell volumes	m^3	REAL	$(1:n_{sb})$
FAREA	Vertically integrated flow face areas	m^2	REAL	$(1:n_{sb})$
$AVGQL_\ell$	Horizontal flow fields for layer ℓ	$m^3 \cdot s^{-1}$	REAL	$(1:n_{hf2})$
$AVGZQL_\ell$	Vertical flow fields for layer ℓ	$m^3 \cdot s^{-1}$	REAL	$(1:n_{hf2})$
$AVGDIFFL_\ell$	Horizontal dispersion fields for layer ℓ	$m^2 \cdot s^{-1}$	REAL	$(1:n_{hf2})$
$AVGZDIFFL_\ell$	Vertical dispersion fields for layer ℓ	$m^2 \cdot s^{-1}$	REAL	$(1:n_{sb})$
$TEMPL_\ell$	Water temperature for layer ℓ	°C	REAL	$(1:n_{sb})$
SWR	Short-wave radiation	$W \cdot m^{-2}$	REAL	$(1:n_{sb})$
LWR	Long-wave radiation	$W \cdot m^{-2}$	REAL	$(1:n_{sb})$
SHF	Sensible heat flux	$W \cdot m^{-2}$	REAL	$(1:n_{sb})$
LHF	Latent heat flux	$W \cdot m^{-2}$	REAL	$(1:n_{sb})$
HF	Total heat flux	$W \cdot m^{-2}$	REAL	$(1:n_{sb})$
WVD	Wave direction	R	REAL	$(1:n_{sb})$
WVH	Wave height	m	REAL	$(1:n_{sb})$
WVP	Wave period	s	REAL	$(1:n_{sb})$

recommended default hydrodynamics file is /mnt/model/dsets/LM3/ R0105/3h/hydro94.cdf on the Linux platform at USEPA-LLRS.

However, there are other hydrodynamics files that have been used by the modelers. One of them is the file /mnt/model/dsets/LM3/R0105/3h_modified/hydro94.cdf which contains the same data as the previous file except for the first 90 days of 1994. The original values for the first 90 days of 1994 were replaced by the values taken from the first 90 days of 1995. The values that were replaced include water temperature, solar radiation, diffusion coefficients, and currents.

Another hydrodynamics file that has been used by the modelers is /mnt/model/dsets/LM3/ R0104/3h/hydro94.cdf. This file does not include any inflows and outflows, unlike the aforementioned files and thus the lake is viewed as a closed system. This file is useful when the user wishes to perform mass balance checks on a given nutrient or some other state variable. A mass balance check is usually done before the user starts doing any extensive calibration on the model.

11.1.3 Loads Data Structure

The loads CDF files provide all of the space- and time-dependent mass loadings used in the LM3 calculation. The data used for providing loading information are contained within two different CDF structures and several different input files. The CDF cannot represent complex data structures. Pointers, linked lists, and similar random access tools are not available in CDF. It is a purely sequential access database. NSSCD plans to upgrade their product to include more differential and data representation.

11.1.3.1 Load Location Structure

The first loads CDF structure is a load locations list that associates pre-defined load location indices with the geometry used by the LM3 calculation. Each tributary, atmospheric, sediment, and shoreline erosion (bluff) load grouping is given a number and this index is mapped to a corresponding group of hydrodynamic cell numbers in the high-resolution calculation. Cell indices corresponding to the different load locations are assigned using CDF global attributes or gAttribute entries (National Aeronautics and Space Administration, 1996a,b,c).

There are two load locations files that have been created; one is used when running eutrophication only, and the other is used when running polychlorinated biphenyls (PCBs) with the eutrophication. The eutrophication calculation needs 45 load locations while the PCB calculation needs a total of 46 load locations. The path for the load locations input file is specified using the FIREAD input key syntax:

*FILE LDLOC <filename>

The load locations file must be specified once and only once, even if mass loads are not being applied. A summary of the gAttributes represented in the load locations input file is provided in Table 11.7. At present, the recommended load location files are /mnt/model/dsets/LM3/R0105/ldlocations.cdf and /mnt/model/dsets/LM3/R0105/ldlocations_pcb.cdf.

11.1.3.2 Load Mass Values Structure

The second loads CDF structure contains the load mass values themselves as well as the time at which those loads are to be applied in the simulation. Each tributary, atmospheric, sediment, and shoreline erosion (bluff) load grouping is represented by two CDF zVariables. The first has the generic name Tm_{ln} and provides the Julian day number at which the load is applied for load number *ln*. Each zVariable corresponds to the load location with the same index *ln* already discussed in Table 11.7. Each load time variable Tm_{ln} contains successive CDF records run specifying the different times at which the load number *ln* is to be applied.

The second CDF variable specifying the corresponding load values has the generic name Ld_{ln}. Each zVariable corresponds to the load location with the same index *ln* already discussed in Table 11.7. Each load mass variable Ld_{ln} specifies the different values of the load to be applied at the location corresponding to *ln* and to the time corresponding to the record number *rn*. A unique load CDF input file must be provided for each simulated state for which a load is to be

Table 11.7. Load Location gAttributes

gAttribute Name	Description
AUTHOR	CDF author name
MXLOC	Maximum number of unreserved loads
LOC_NM	List of descriptive load names
LOC_NC	Number of cells for each load
LOC_CELLS_01	Menominee River
LOC_CELLS_02	Fox River
LOC_CELLS_03	Sheboygan River
LOC_CELLS_04	Milwaukee River
LOC_CELLS_05	Grand Calumet River
LOC_CELLS_06	St. Joseph River
LOC_CELLS_07	Kalamazoo River
LOC_CELLS_08	Grand River
LOC_CELLS_09	Muskegon River
LOC_CELLS_10	Pere Marquette River
LOC_CELLS_11	Manistique River
LOC_CELLS_12	Cedar River
LOC_CELLS_13	Peshtigo River
LOC_CELLS_14	Oconto River
LOC_CELLS_15	Pensaukee River
LOC_CELLS_16	Duck River
LOC_CELLS_17	Kewaunee River
LOC_CELLS_18	East Twin River
LOC_CELLS_19	West Twin River
LOC_CELLS_20	Manitowoc River
LOC_CELLS_21	Root River
LOC_CELLS_22	Black (SH) River
LOC_CELLS_23	Black (HD) River
LOC_CELLS_24	Pigeon River
LOC_CELLS_25	White River
LOC_CELLS_26	Pentwater River
LOC_CELLS_27	Big Sable River
LOC_CELLS_28	Manistee River
LOC_CELLS_29	Betsie River
LOC_CELLS_30	Boardman River
LOC_CELLS_31	Jordan River
LOC_CELLS_32	Sturgeon River
LOC_CELLS_33	Whitefish River
LOC_CELLS_34	Rapid River
LOC_CELLS_35	Escanaba River
LOC_CELLS_36	Ford River
LOC_CELLS_37	Chicago SS/Canal
LOC_CELLS_38	Straits of Mackinac
LOC_CELLS_39	Atmospheric interface
LOC_CELLS_40	Sediment interface
LOC_CELLS_41	Unmonitored Kalamazoo River
LOC_CELLS_42	Unmonitored Pere Marquette River
LOC_CELLS_43	Milecoquins River
LOC_CELLS_44	Galien River
LOC_CELLS_45	Shoreline erosion load values
LOC_CELLS_46	Atmospheric interface

defined. The path for the load time and mass values input files is specified using the FIREAD input key syntax.

* FILE LOAD <state> <filename>

Only one load file may be specified for each state. States for which no load file is provided will be simulated without mass loads. A summary of the load file update time zVariables is provided in Table 11.8. A summary of the load file mass value zVariables is provided in Table 11.9. A summary of the FIREAD load state number specifications and recommended load file path is provided in Tables 11.10, 11.11, and 11.12.

11.2 Output Data Structures

Output data generated by the LM3 simulation included space- and time-dependent concentration and mass balance information for the different studies defined within the simulation. As with the output of the hydrodynamics program whose flow and dispersion fields LM3 uses, the output of the water quality model calculation can be of the order of gigabytes because dozens of state variables can be output on the high-resolution grid (see Section 5.1) As with much of the input data, the task of organizing program output into a format easily accessible to analysis and graphical plotting programs had been assigned for CDF, and that data format had been employed universally to the LM3 output of space- and time-dependent state information. State variable output is documented by LM3 within the CDF data structure; descriptions of states are provided and units for each variable are described. As with the program input, the C library XDR representation is used so that program output can be accessed across platforms. Unlike the LM3 input, all CDF output files are presumed to contain both space- and time-dependent degrees of freedom, and all output data employed the CDF zVariable representation scheme (National Aeronautics and Space Administration, 1996a, b, c).

The details of the output structure vary depending on the study being simulated. Common to the different simulation scenarios is the presence of timestamp information describing the simulation time corresponding to each CDF zVariable record. This is reviewed in Section 11.2.1. Program output for hydrodynamics, shear, eutrophication, sand, atrazine, and sediment transport are unique to each of those cases, and the CDF output appropriate to each is discussed in Sections 11.2.2 through 11.2.8.

Beyond the selection of state variables and the study being employed, the “kinds” of program output are limited to the choice of grid resolution. Though the simulation is always performed on the scale of the high-resolution grid, CDF output can be generated either on the scale of this grid or integrated onto the scale of the LMMBP coarser grid known as the level 2 grid.

High-resolution LM3 output is currently written to files with the names bdump.cdf and bdump.z*. Output using this format is extensive and can be used to make point comparisons between model predictions and field data from individual stations. The high-resolution model output is also appropriate for generating maps and animation sequences that distinguish nearshore and other small scale gradients. Coarse-resolution LM3 output is currently written to files with the names sdump.cdf and sdump.z*. Output using this format is considerably more compact and can be used to make point comparisons between field data averaged over the level 2 segmentation and level 2 model predictions. Both the high- and coarse-resolution CDF output files enjoy identical internal data structures. These are discussed in the sections below.

Table 11.8. Load File Update Time zVariables

gAttribute Name	Description	Units
TM_01	Update time for Menominee River load	Julian days
TM_02	Update time for Fox River load	Julian days
TM_03	Update time for Sheboygan River load	Julian days
TM_04	Update time for Milwaukee River load	Julian days
TM_05	Update time for Grand Calumet River load	Julian days
TM_06	Update time for St. Joseph River load	Julian days
TM_07	Update time for Kalamazoo River load	Julian days
TM_08	Update time for Grand River load	Julian days
TM_09	Update time for Muskegon River load	Julian days
TM_10	Update time for Pere Marquette River load	Julian days
TM_11	Update time for Manistique River load	Julian days
TM_12	Update time for Cedar River load	Julian days
TM_13	Update time for Peshtigo River load	Julian days
TM_14	Update time for Oconto River load	Julian days
TM_15	Update time for Pensaukee River load	Julian days
TM_16	Update time for Duck River load	Julian days
TM_17	Update time for Kewaunee River load	Julian days
TM_18	Update time for East Twin River load	Julian days
TM_19	Update time for West Twin River load	Julian days
TM_20	Update time for Manitowoc River load	Julian days
TM_21	Update time for Root River load	Julian days
TM_22	Update time for Black (SH) River load	Julian days
TM_23	Update time for Black (HD) River load	Julian days
TM_24	Update time for Pigeon River load	Julian days
TM_25	Update time for White River load	Julian days
TM_26	Update time for Pentwater River load	Julian days
TM_27	Update time for Big Sable River load	Julian days
TM_28	Update time for Manistee River load	Julian days
TM_29	Update time for Betsie River load	Julian days
TM_30	Update time for Boardman River load	Julian days
TM_31	Update time for Jordan River load	Julian days
TM_32	Update time for Sturgeon River load	Julian days
TM_33	Update time for Whitefish River load	Julian days
TM_34	Update time for Rapid River load	Julian days
TM_35	Update time for Escanaba River load	Julian days
TM_36	Update time for Ford River load	Julian days
TM_37	Update time for Chicago SS/Canal load	Julian days
TM_38	Update time for Straits of Mackinac load	Julian days
TM_39	Update time for atmospheric interface load	Julian days
TM_40	Update time for sediment interface load	Julian days
TM_41	Update time for unmonitored Kalamazoo River load	Julian days
TM_42	Update time for unmonitored Pere Marquette River load	Julian days
TM_43	Update time for Milecoquins River load	Julian days
TM_44	Update time for Galien River load	Julian days
TM_45	Update time for shoreline erosion load values	Julian days
TM_46	Update time for atmospheric interface load	Julian days

Table 11.9. Load File Mass Load zVariables

gAttribute Name	Description	Units
LD_01	Load for Menominee River load	$kg \cdot s^{-1}$
LD_02	Load for Fox River load	$kg \cdot s^{-1}$
LD_03	Load for Sheboygan River load	$kg \cdot s^{-1}$
LD_04	Load for Milwaukee River load	$kg \cdot s^{-1}$
LD_05	Load for Grand Calumet River load	$kg \cdot s^{-1}$
LD_06	Load for St. Joseph River load	$kg \cdot s^{-1}$
LD_07	Load for Kalamazoo River load	$kg \cdot s^{-1}$
LD_08	Load for Grand River load	$kg \cdot s^{-1}$
LD_09	Load for Muskegon River load	$kg \cdot s^{-1}$
LD_10	Load for Pere Marquette River load	$kg \cdot s^{-1}$
LD_11	Load for Manistique River load	$kg \cdot s^{-1}$
LD_12	Load for Cedar River load	$kg \cdot s^{-1}$
LD_13	Load for Peshtigo River load	$kg \cdot s^{-1}$
LD_14	Load for Oconto River load	$kg \cdot s^{-1}$
LD_15	Load for Pensaukee River load	$kg \cdot s^{-1}$
LD_16	Load for Duck River load	$kg \cdot s^{-1}$
LD_17	Load for Kewaunee River load	$kg \cdot s^{-1}$
LD_18	Load for East Twin River load	$kg \cdot s^{-1}$
LD_19	Load for West Twin River load	$kg \cdot s^{-1}$
LD_20	Load for Manitowoc River load	$kg \cdot s^{-1}$
LD_21	Load for Root River load	$kg \cdot s^{-1}$
LD_22	Load for Black (SH) River load	$kg \cdot s^{-1}$
LD_23	Load for Black (HD) River load	$kg \cdot s^{-1}$
LD_24	Load for Pigeon River load	$kg \cdot s^{-1}$
LD_25	Load for White River load	$kg \cdot s^{-1}$
LD_26	Load for Pentwater River load	$kg \cdot s^{-1}$
LD_27	Load for Big Sable River load	$kg \cdot s^{-1}$
LD_28	Load for Manistee River load	$kg \cdot s^{-1}$
LD_29	Load for Betsie River load	$kg \cdot s^{-1}$
LD_30	Load for Boardman River load	$kg \cdot s^{-1}$
LD_31	Load for Jordan River load	$kg \cdot s^{-1}$
LD_32	Load for Sturgeon River load	$kg \cdot s^{-1}$
LD_33	Load for Whitefish River load	$kg \cdot s^{-1}$
LD_34	Load for Rapid River load	$kg \cdot s^{-1}$
LD_35	Load for Escanaba River load	$kg \cdot s^{-1}$
LD_36	Load for Ford River load	$kg \cdot s^{-1}$
LD_37	Load for Chicago SS/Canal load	$kg \cdot s^{-1}$
LD_38	Load for Straits of Mackinac load	$kg \cdot s^{-1}$
LD_39	Load for atmospheric interface load	$kg \cdot s^{-1}$
LD_40	Load for sediment interface load	$kg \cdot s^{-1}$
LD_41	Load for unmonitored Kalamazoo River load	$kg \cdot s^{-1}$
LD_42	Load for unmonitored Pere Marquette River load	$kg \cdot s^{-1}$
LD_43	Load for Milecoquins River load	$kg \cdot s^{-1}$
LD_44	Load for Galien River load	$kg \cdot s^{-1}$
LD_45	Load for shoreline erosion load values	$kg \cdot s^{-1}$
LD_46	Load for atmospheric interface load	$kg \cdot s^{-1}$

Table 11.10. FIREAD Eutrophication Load Names and Recommended Input Files

State Name	Description	Recommended File Path
DOC	Dissolved organic carbon	/home/llrs_model/dsets/LM3/R0105/ld/ld_doc.cdf
DIA	Diatom carbon	/home/llrs_model/dsets/LM3/R0105/ld/ld_dia.cdf
GRE	Non-diatom phytoplankton carbon	/home/llrs_model/dsets/LM3/R0105/ld/ld_gre.cdf
LOC	Labile particulate organic carbon	/home/llrs_model/dsets/LM3/R0105/ld/ld_lpoc.cdf
ROC	Refractory particulate organic carbon	/home/llrs_model/dsets/LM3/R0105/ld/ld_rpoc.cdf
SRP	Soluble reactive phosphorus	/home/llrs_model/dsets/LM3/R0105/ld/ld_srp.cdf
DOP	Dissolved organic phosphorus	/home/llrs_model/dsets/LM3/R0105/ld/ld_dop.cdf
LOP	Labile particulate organic phosphorus	/home/llrs_model/dsets/LM3/R0105/ld/ld_lop.cdf
ROP	Refractory particulate organic phosphorus	/home/llrs_model/dsets/LM3/R0105/ld/ld_rop.cdf
NH_4	Ammonia	/home/llrs_model/dsets/LM3/R0105/ld/ld_nh4.cdf
NO_3	Nitrate	/home/llrs_model/dsets/LM3/R0105/ld/ld_no3.cdf
DON	Dissolved organic nitrogen	/home/llrs_model/dsets/LM3/R0105/ld/ld_don.cdf
LON	Labile particulate organic nitrogen	/home/llrs_model/dsets/LM3/R0105/ld/ld_lon.cdf
RON	Refractory particulate organic nitrogen	/home/llrs_model/dsets/LM3/R0105/ld/ld_ron.cdf
SA	Available silica	/home/llrs_model/dsets/LM3/R0105/ld/ld_sa.cdf
SU	Particulate biogenic silica	No loads file
ZOO	Zooplankton	No loads file

Table 11.11. FIREAD PCB Load Names and Recommended Input Files

State Name	Description	Recommended File Path
PCB_5+8	PCB congener 5+8 concentration	/home/llrs_model/dsets/LM3/R0105/ld/ld_pcb_5+8.cdf
PCB_15+17	PCB congener 15+17 concentration	/home/llrs_model/dsets/LM3/R0105/ld/ld_pcb_15+17.cdf
PCB_16+32	PCB congener 16+32 concentration	/home/llrs_model/dsets/LM3/R0105/ld/ld_pcb_16+32.cdf
PCB_18	PCB congener 18 concentration	/home/llrs_model/dsets/LM3/R0105/ld/ld_pcb_18.cdf
PCB_26	PCB congener 26 concentration	/home/llrs_model/dsets/LM3/R0105/ld/ld_pcb_26.cdf
PCB_28+31	PCB congener 28+31 concentration	/home/llrs_model/dsets/LM3/R0105/ld/ld_pcb_28+31.cdf
PCB_33	PCB congener 33 concentration	/home/llrs_model/dsets/LM3/R0105/ld/ld_pcb_33.cdf
PCB_37+42	PCB congener 37+42 concentration	/home/llrs_model/dsets/LM3/R0105/ld/ld_pcb_37+42.cdf
PCB_44	PCB congener 44 concentration	/home/llrs_model/dsets/LM3/R0105/ld/ld_pcb_44.cdf
PCB_49	PCB congener 49 concentration	/home/llrs_model/dsets/LM3/R0105/ld/ld_pcb_49.cdf
PCB_52	PCB congener 52 concentration	/home/llrs_model/dsets/LM3/R0105/ld/ld_pcb_52.cdf
PCB_56+60	PCB congener 56+60 concentration	/home/llrs_model/dsets/LM3/R0105/ld/ld_pcb_56+60.cdf
PCB_66	PCB congener 66 concentration	/home/llrs_model/dsets/LM3/R0105/ld/ld_pcb_66.cdf
PCB_70+76	PCB congener 70+76 concentration	/home/llrs_model/dsets/LM3/R0105/ld/ld_pcb_70+76.cdf
PCB_74	PCB congener 74 concentration	/home/llrs_model/dsets/LM3/R0105/ld/ld_pcb_74.cdf
PCB_77+110	PCB congener 77+110 concentration	/home/llrs_model/dsets/LM3/R0105/ld/ld_pcb_77+110.cdf
PCB_81	PCB congener 81 concentration	/home/llrs_model/dsets/LM3/R0105/ld/ld_pcb_81.cdf
PCB_87	PCB congener 87 concentration	/home/llrs_model/dsets/LM3/R0105/ld/ld_pcb_87.cdf
PCB_84+92	PCB congener 84+92 concentration	/home/llrs_model/dsets/LM3/R0105/ld/ld_pcb_84+92.cdf
PCB_89	PCB congener 89 concentration	/home/llrs_model/dsets/LM3/R0105/ld/ld_pcb_89.cdf
PCB_99	PCB congener 99 concentration	/home/llrs_model/dsets/LM3/R0105/ld/ld_pcb_99.cdf
PCB_101	PCB congener 101 concentration	/home/llrs_model/dsets/LM3/R0105/ld/ld_pcb_101.cdf
PCB_118	PCB congener 118 concentration	/home/llrs_model/dsets/LM3/R0105/ld/ld_pcb_118.cdf
PCB_123+149	PCB congener 123+149 concentration	/home/llrs_model/dsets/LM3/R0105/ld/ld_pcb_123+149.cdf
PCB_105+132+153	PCB congener 105+132+153 concentration	/home/llrs_model/dsets/LM3/R0105/ld/ld_pcb_105+132+153.cdf
PCB_151	PCB congener 151 concentration	/home/llrs_model/dsets/LM3/R0105/ld/ld_pcb_151.cdf
PCB_138+163	PCB congener 138+163 concentration	/home/llrs_model/dsets/LM3/R0105/ld/ld_pcb_138+163.cdf
PCB_170+190	PCB congener 170+190 concentration	/home/llrs_model/dsets/LM3/R0105/ld/ld_pcb_170+190.cdf
PCB_172+197	PCB congener 172+197 concentration	/home/llrs_model/dsets/LM3/R0105/ld/ld_pcb_172+197.cdf
PCB_180	PCB congener 180 concentration	/home/llrs_model/dsets/LM3/R0105/ld/ld_pcb_180.cdf
PCB_182+187	PCB congener 182+187 concentration	/home/llrs_model/dsets/LM3/R0105/ld/ld_pcb_182+187.cdf
PCB_195+208	PCB congener 195+208 concentration	/home/llrs_model/dsets/LM3/R0105/ld/ld_pcb_195+208.cdf
PCB_196+203	PCB congener 196+203 concentration	/home/llrs_model/dsets/LM3/R0105/ld/ld_pcb_196+203.cdf
PCB_201	PCB congener 201 concentration	/home/llrs_model/dsets/LM3/R0105/ld/ld_pcb_201.cdf
PCB_146	PCB congener 146 concentration	/home/llrs_model/dsets/LM3/R0105/ld/ld_pcb_146.cdf
PCB_85	PCB congener 85 concentration	/home/llrs_model/dsets/LM3/R0105/ld/ld_pcb_85.cdf

Table 11.12. FIREAD Atrazine, Chloride, and Clay Load Names and Recommended Input Files

State Name	Description	Recommended File Path
ATZ	Atrazine	/home/llrs_model/dsets/LM3/R0105/ld/ld_atz.cdf or /home/llrs_model/dsets/LM3/R0105/ld/ld_041129atz.cdf or /home/llrs_model/dsets/LM3/R0105/ld/ld_041129atz.cdf
CL	Chloride	/home/llrs_model/dsets/LM3/R0105/ld/ld_cl.cdf
CLAY	Clay	No loads file at the present time

11.2.1 Timestamps

Three timestamp zVariables are written for each CDF output record. These correspond to a conventional time value that can be interpreted by analysis software, a time value that can be interpreted by the CDF applications and libraries and viewed as an easily recognizable calendar date, and the internal timestamp used by the LM3 simulation.

Quoting from the SNIPPETS library source: “Julian Day Numbers (JDN) are used by astronomers as a date/time measure independent of calendars and convenient for computing the elapsed time between dates. The JDN for any date/time is the number of days (including fractional days) elapsed since noon, 1 Jan 4713 BC. JDN was originated by Joseph Scaliger in 1582 and named after his father Julius. They are not related to the Julian calendar”. The JDN at the start of the LMMBP study period was 2,449,352.5. The calendar date value is most often used by users perusing the CDF output directly using tools such as cdfedit to infer calendar values for each of the corresponding output records. The actual format of this data is intrinsic to the CDF library and application distribution and must be interpreted through CDF. The LM3 timestamp is a counter of the number of simulated seconds beginning with the start of the hydrodynamic input; this does not necessarily correspond to the start of the LM3 simulation run because several days of hydrodynamic input are typically skipped to account for stabilization of the simulated current and dispersion fields. A summary of these timestamps and their corresponding CDF zVariables is provided in Table 11.13.

11.2.2 Hydrodynamics

Hydrodynamics are not simulated by the LM3 program, but are simply read bootstrap and used in performing transport of simulated states. Values passed from input to the LM3 output have

Table 11.13. Output Simulation Time zVariables

zVariable	Description	Units	Data Type	Dimension
TIME_JULIAN	Julian day number	Days	DOUBLE	Scalar
TIME_EPOCH	Calendar date and time	None	CDF_EPOCH	Scalar
TIME_RUNSEC	Simulation timestamp	s	DOUBLE	Scalar

already been described in Section 11.1.2. Unlike the input hydrodynamics values, output values are written at considerably longer time intervals, and zVariables for the output variables are not split by vertical layer number ℓ. Also, current velocities are written instead of flows, and these, and values for the dispersion rates, are averaged between flow faces that bound the cells so that the resulting values are defined at the cell centers instead of the flow faces. A list of these output zVariables is provided in Table 11.14.

Table 11.14. Output Hydrodynamics zVariables. Array Dimensions Refer to the Number of Cells n_c

zVariable	Description	Units	Data Type	Dimension
VOL	Cell volumes	m^3	DOUBLE	$(1:n_c)$
TEM	Temperature	°C	REAL	$(1:n_c)$
X_CURRENT	Currents X-component	$m \cdot s^{-1}$	REAL	$(1:n_c)$
Y_CURRENT	Currents Y-component	$m \cdot s^{-1}$	REAL	$(1:n_c)$
AVGDIFF	Horizontal dispersion fields	$m^2 \cdot s^{-1}$	REAL	$(1:n_c)$
AVGZDIFF	Vertical dispersion fields	$m^2 \cdot s^{-1}$	REAL	$(1:n_c)$

11.2.3 Shear Calculations

Shear stress calculations can be performed in LM3. Shear stress is not being used within LM3 because a fully developed sediment resuspension scheme has not been implemented as of this writing. We are expecting to have a more complete sediment transport algorithm in the future. Nevertheless, the shear stress calculation is available in the program for anybody to test and output. Variables that can be written to output correspond to values of bottom shear stress. Also other variables that can be written are the input values of wave direction, wave height, and wave period. A list of these output zVariables is provided in Table 11.15.

11.2.4 Eutrophication

Eutrophication and the autochthonous production of solids involve the simulation of more than a dozen of states and their relationships *via* kinetic processes that can transform one state to

Table 11.15. Shear Calculation zVariables. Array Dimensions Refer to the Number of Surface Cells n_{sb}

zVariable	Description	Units	Data Type	Dimension
SHEAR_STRESS	Bottom shear stress	$N \cdot m^{-2}$	REAL	$(1:n_{sb})$
WVD	Wave direction	R	REAL	$(1:n_{sb})$
WVH	Significant wave height	m	REAL	$(1:n_{sb})$
WVP	Significant wave period	s	REAL	$(1:n_{sb})$

another (see Chapter 6). Carbon in the forms of dissolved organic carbon (DOC), diatoms, non-diatom phytoplankton, zooplankton, refractory detrital carbon (RDC), and labile detrital carbon (LDC) are simulated and output as volumetric concentrations. There are several phosphorus, nitrogen, and silica states. The specific states include the inorganic/available and organic/unavailable components of these nutrients. The total list of concentration zVariables for these eutrophication simulation states is provided in Table 11.16.

Table 11.16. Eutrophication State Concentration zVariables. Array Dimensions Refer to the Number of Cells n_c

zVariable	Description	Units	Data Type	Dimension
DOC	Dissolved organic carbon concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
DIA	Diatom carbon concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
GRE	Non-diatom phytoplankton concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
ZOO	Zooplankton concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
LOC	Labile organic carbon concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
ROC	Refractory organic carbon concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
SRP	Soluble reactive phosphorus concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
DOP	Dissolved organic phosphorus concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
LOP	Labile organic phosphorus concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
ROP	Refractory organic phosphorus concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
NH_4	Ammonia concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
NO_3	Nitrate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
DON	Dissolved organic nitrogen concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
LON	Labile organic nitrogen concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
RON	Refractory organic nitrogen concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
SA	Available silica concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
SU	Particulate biogenic silica concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$

A lake-wide mass and time-integrated load calculation for each of the eutrophication states is also written to output when mass balance calculations are enabled. The CDF zVariables corresponding to these lake-wide masses are listed in Table 11.17. Time-integrated settling loss of state masses and settling rates for the phytoplankton, particulate detrital carbon (PDC), and particulate nutrient states are also written as part of the mass balance information for the eutrophication study, and corresponding zVariables are summarized in Table 11.18.

Growth limitation factors for the eutrophication kinetics can also be written to the output CDF. These include the calculated limitation factors for light, temperature, nitrogen, silica, and phosphorus for diatoms; and light, temperature, nitrogen, and phosphorus for the non-diatom phytoplankton state variable. zVariables for the growth limitation factors are provided in Table 11.19.

11.2.5 PCB Congeners

There are 36 PCB congener groups currently simulated within the LM3 application. PCBs undergo transport and kinetics processes in the water column. PCBs partition with respect to carbon-based solids and their water concentration can exist in dissolved and particulate forms.

Table 11.17. Lake-wide Eutrophication State Mass and Load zVariables

zVariable	Description	Units	Data Type	Dimension
DOC_MS	Dissolved organic carbon mass	kg	REAL	Scalar
DIA_MS	Diatom carbon mass	kg	REAL	Scalar
GRE_MS	Non-diatom phytoplankton mass	kg	REAL	Scalar
ZOO_MS	Zooplankton mass	kg	REAL	Scalar
LOC_MS	Labile particulate organic carbon mass	kg	REAL	Scalar
ROC_MS	Refractory particulate organic carbon mass	kg	REAL	Scalar
SRP_MS	Soluble reactive phosphorus mass	kg	REAL	Scalar
DOP_MS	Dissolved organic phosphorus mass	kg	REAL	Scalar
LOP_MS	Labile particulate organic phosphorus mass	kg	REAL	Scalar
ROP_MS	Refractory particulate organic phosphorus mass	kg	REAL	Scalar
NH4_MS	Ammonia mass	kg	REAL	Scalar
NO3_MS	Nitrate mass	kg	REAL	Scalar
DON_MS	Dissolved organic nitrogen mass	kg	REAL	Scalar
LON_MS	Labile particulate organic nitrogen mass	kg	REAL	Scalar
RON_MS	Refractory particulate organic nitrogen mass	kg	REAL	Scalar
SA_MS	Available silica mass	kg	REAL	Scalar
SU_MS	Particulate biogenic silica mass	kg	REAL	Scalar
DOC_LD	Dissolved organic carbon load	kg	REAL	Scalar
DIA_LD	Diatom carbon load	kg	REAL	Scalar
GRE_LD	Non-diatom phytoplankton load	kg	REAL	Scalar
ZOO_LD	Zooplankton load	kg	REAL	Scalar
LOC_LD	Labile particulate organic carbon load	kg	REAL	Scalar
ROC_LD	Refractory particulate organic carbon load	kg	REAL	Scalar
SRP_LD	Soluble reactive phosphorus load	kg	REAL	Scalar
DOP_LD	Dissolved organic phosphorus load	kg	REAL	Scalar
LOP_LD	Labile particulate organic phosphorus load	kg	REAL	Scalar
ROP_LD	Refractory particulate organic phosphorus load	kg	REAL	Scalar
NH4_LD	Ammonia load	kg	REAL	Scalar
NO3_LK	Nitrate load	kg	REAL	Scalar
DON_LD	Dissolved organic nitrogen load	kg	REAL	Scalar
LON_LD	Labile particulate organic nitrogen load	kg	REAL	Scalar
RON_LD	Refractory particulate organic nitrogen load	kg	REAL	Scalar
SA_LD	Available silica load	kg	REAL	Scalar
SU_LD	Particulate biogenic silica load	kg	REAL	Scalar

Table 11.18 Eutrophication Settling Losses and Rates zVariables. Array Dimensions Refer to the Number of Surface Cells n_{sb}

zVariable	Description	Units	Data Type	Dimension
DIA_SET	Diatom carbon settling loss	$kg \cdot s^{-1}$	REAL	$(1:n_{sb})$
GRE_SET	Non-diatom phytoplankton settling loss	$kg \cdot s^{-1}$	REAL	$(1:n_{sb})$
LOC_SET	Labile organic carbon settling loss	$kg \cdot s^{-1}$	REAL	$(1:n_{sb})$
ROC_SET	Refractory organic carbon settling loss	$kg \cdot s^{-1}$	REAL	$(1:n_{sb})$
LOP_SET	Labile organic phosphorus settling loss	$kg \cdot s^{-1}$	REAL	$(1:n_{sb})$
ROP_SET	Refractory organic phosphorus settling loss	$kg \cdot s^{-1}$	REAL	$(1:n_{sb})$
DON_SET	Dissolved organic nitrogen settling loss	$kg \cdot s^{-1}$	REAL	$(1:n_{sb})$
LON_SET	Labile organic nitrogen settling loss	$kg \cdot s^{-1}$	REAL	$(1:n_{sb})$
RON_SET	Refractory organic nitrogen settling loss	$kg \cdot s^{-1}$	REAL	$(1:n_{sb})$
SU_SET	Particulate biogenic silica settling loss	$kg \cdot s^{-1}$	REAL	$(1:n_{sb})$
DIA_SETRT	Diatom carbon settling loss rate	$kg \cdot s^{-1}$	REAL	$(1:n_{sb})$
GRE_SETRT	Non-diatom phytoplankton settling loss rate	$kg \cdot s^{-1}$	REAL	$(1:n_{sb})$
LOC_SETRT	Labile organic carbon settling loss rate	$kg \cdot s^{-1}$	REAL	$(1:n_{sb})$
ROC_SETRT	Refractory organic carbon settling loss rate	$kg \cdot s^{-1}$	REAL	$(1:n_{sb})$
LOP_SETRT	Labile organic phosphorus settling loss rate	$kg \cdot s^{-1}$	REAL	$(1:n_{sb})$
ROP_SETRT	Refractory organic phosphorus settling loss rate	$kg \cdot s^{-1}$	REAL	$(1:n_{sb})$
DON_SETRT	Dissolved organic nitrogen settling loss rate	$kg \cdot s^{-1}$	REAL	$(1:n_{sb})$
LON_SETRT	Labile organic nitrogen settling loss rate	$kg \cdot s^{-1}$	REAL	$(1:n_{sb})$
RON_SETRT	Refractory organic nitrogen settling loss rate	$kg \cdot s^{-1}$	REAL	$(1:n_{sb})$
SU_SETRT	Particulate biogenic silica settling loss rate	$kg \cdot s^{-1}$	REAL	$(1:n_{sb})$

Table 11.19. Phytoplankton Growth Limitation Factor zVariables. Array Dimensions Refer to the Number of Cells n_c

zVariable	Description	Units	Data Type	Dimension
PFD	Phosphorus limitation for diatoms	None	REAL	$(1:n_c)$
SFD	Silica limitation for diatoms	None	REAL	$(1:n_c)$
NFD	Nitrogen limitation for diatoms	None	REAL	$(1:n_c)$
IFD	Light limitation for diatoms	None	REAL	$(1:n_c)$
TFD	Temperature limitation for diatoms	None	REAL	$(1:n_c)$
PFG	Phosphorus limitation for non-diatom phytoplankton	None	REAL	$(1:n_c)$
NFG	Nitrogen limitation for non-diatom phytoplankton	None	REAL	$(1:n_c)$
IFG	Light limitation for non-diatom phytoplankton	None	REAL	$(1:n_c)$
TFG	Temperature limitation for non-diatom phytoplankton	None	REAL	$(1:n_c)$

There are three separate PCB congener water concentrations that are written to output: total, dissolved, and particulate. The list of the PCB congener total water concentration zVariables is given in Table 11.20. The list of the PCB congener dissolved water concentration zVariables is given in Table 11.21. The list of the PCB congener particulate water concentration zVariables is given in Table 11.22.

A lake-wide mass and time-integrated total load mass calculation for each of the PCB congener states is also written to output when mass balance calculations are enabled. The CDF zVariables corresponding to the lake-wide PCB congener masses are listed in Table 11.23.

Table 11.20. PCB Congener Total Water Concentration zVariables. Array Dimensions Refer to the Number of Cells n_c

zVariable	Description	Units	Data Type	Dimension
PCB_5+8	PCB congener 5+8 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB15+17	PCB congener 15+17 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_16+32	PCB congener 16+32 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_18	PCB congener 18 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_26	PCB congener 26 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_28+31	PCB congener 28+31 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_33	PCB congener 33 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_37+42	PCB congener 37+42 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_44	PCB congener 44 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_49	PCB congener 49 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_52	PCB congener 52 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_56+60	PCB congener 56+60 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_66	PCB congener 66 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_70+76	PCB congener 70+76 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_74	PCB congener 74 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_77+110	PCB congener 77+110 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_81	PCB congener 81 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_87	PCB congener 87 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_84+92	PCB congener 84+92 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_89	PCB congener 89 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_99	PCB congener 99 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_101	PCB congener 101 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_118	PCB congener 118 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_123+149	PCB congener 123+149 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_105+132 +153	PCB congener 105+132+153 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_151	PCB congener 151 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_138+163	PCB congener 138+163 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB _170+190	PCB congener 170+190 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_172+197	PCB congener 172+197 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_180	PCB congener 180 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_182+187	PCB congener 182+187 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_195+208	PCB congener 195+208 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_196+203	PCB congener 196+203 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_201	PCB congener 201 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_146	PCB congener 146 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_85	PCB congener 85 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$

Table 11.21. PCB Congener Dissolved Water Concentration zVariables. Array Dimensions Refer to the Number of Cells n_c

zVariable	Description	Units	Data Type	Dimension
PCB_5+8_DIS	PCB congener 5+8 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB15+17_DIS	PCB congener 15+17 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_16+32_DIS	PCB congener 16+32 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_18_DIS	PCB congener 18 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_26_DIS	PCB congener 26 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_28+31_DIS	PCB congener 28+31 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_33_DIS	PCB congener 33 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_37+42_DIS	PCB congener 37+42 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_44_DIS	PCB congener 44 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_49_DIS	PCB congener 49 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_52_DIS	PCB congener 52 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_56+60_DIS	PCB congener 56+60 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_66_DIS	PCB congener 66 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_70+76_DIS	PCB congener 70+76 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_74_DIS	PCB congener 74 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_77+110_DIS	PCB congener 77+110 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_81_DIS	PCB congener 81 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_87_DIS	PCB congener 87 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_84+92_DIS	PCB congener 84+92 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_89_DIS	PCB congener 89 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_99_DIS	PCB congener 99 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_101_DIS	PCB congener 101 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_118_DIS	PCB congener 118 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_123+149_DIS	PCB congener 123+149 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_105+132+153_DIS	PCB congener 105+132+153 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_151_DIS	PCB congener 151 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_138+163_DIS	PCB congener 138+163 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB _170+190_DIS	PCB congener 170+190 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_172+197_DIS	PCB congener 172+197 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_180_DIS	PCB congener 180 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_182+187_DIS	PCB congener 182+187 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_195+208_DIS	PCB congener 195+208 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_196+203_DIS	PCB congener 196+203 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_201_DIS	PCB congener 201 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_146_DIS	PCB congener 146 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_85_DIS	PCB congener 85 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$

Table 11.22. PCB Congener Particulate Water Concentration zVariables. Array Dimensions Refer to the Number of Cells n_c

zVariable	Description	Units	Data Type	Dimension
PCB_5+8_PART	PCB congener 5+8 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB15+17_PART	PCB congener 15+17 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_16+32_PART	PCB congener 16+32 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_18_PART	PCB congener 18 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_26_PART	PCB congener 26 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_28+31_PART	PCB congener 28+31 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_33_PART	PCB congener 33 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_37+42_PART	PCB congener 37+42 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_44_PART	PCB congener 44 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_49_PART	PCB congener 49 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_52_PART	PCB congener 52 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_56+60_PART	PCB congener 56+60 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_66_PART	PCB congener 66 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_70+76_PART	PCB congener 70+76 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_74_PART	PCB congener 74 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_77+110_PART	PCB congener 77+110 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_81_PART	PCB congener 81 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_87_PART	PCB congener 87 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_84+92_PART	PCB congener 84+92 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_89_PART	PCB congener 89 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_99_PART	PCB congener 99 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_101_PART	PCB congener 101 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_118_PART	PCB congener 118 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_123+149_PART	PCB congener 123+149 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_105+132+153_PART	PCB congener 105+132+153 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_151_PART	PCB congener 151 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_138+163_PART	PCB congener 138+163 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB _170+190_PART	PCB congener 170+190 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_172+197_PART	PCB congener 172+197 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_180_PART	PCB congener 180 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_182+187_PART	PCB congener 182+187 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_195+208_PART	PCB congener 195+208 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_196+203_PART	PCB congener 196+203 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_201_PART	PCB congener 201 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_146_PART	PCB congener 146 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
PCB_85_PART	PCB congener 85 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$

Table 11.23. PCB Congener Water Column Mass zVariables

zVariable	Description	Units	Data Type	Dimension
PCB_5+8_MS	PCB congener 5+8 water column mass	kg	REAL	Scalar
PCB15+17_MS	PCB congener 15+17 water column mass	kg	REAL	Scalar
PCB_16+32_MS	PCB congener 16+32 water column mass	kg	REAL	Scalar
PCB_18_MS	PCB congener 18 water column mass	kg	REAL	Scalar
PCB_26_MS	PCB congener 26 water column mass	kg	REAL	Scalar
PCB_28+31_MS	PCB congener 28+31 water column mass	kg	REAL	Scalar
PCB_33_MS	PCB congener 33 water column mass	kg	REAL	Scalar
PCB_37+42_MS	PCB congener 37+42 water column mass	kg	REAL	Scalar
PCB_44_MS	PCB congener 44 water column mass	kg	REAL	Scalar
PCB_49_MS	PCB congener 49 water column mass	kg	REAL	Scalar
PCB_52_MS	PCB congener 52 water column mass	kg	REAL	Scalar
PCB_56+60_MS	PCB congener 56+60 water column mass	kg	REAL	Scalar
PCB_66_MS	PCB congener 66 water column mass	kg	REAL	Scalar
PCB_70+76_MS	PCB congener 70+76 water column mass	kg	REAL	Scalar
PCB_74_MS	PCB congener 74 water column mass	kg	REAL	Scalar
PCB_77+110_MS	PCB congener 77+110 water column mass	kg	REAL	Scalar
PCB_81_MS	PCB congener 81 water column mass	kg	REAL	Scalar
PCB_87_MS	PCB congener 87 water column mass	kg	REAL	Scalar
PCB_84+92_MS	PCB congener 84+92 water column mass	kg	REAL	Scalar
PCB_89_MS	PCB congener 89 water column mass	kg	REAL	Scalar
PCB_99_MS	PCB congener 99 water column mass	kg	REAL	Scalar
PCB_101_MS	PCB congener 101 water column mass	kg	REAL	Scalar
PCB_118_MS	PCB congener 118 water column mass	kg	REAL	Scalar
PCB_123+149_MS	PCB congener 123+149 water column mass	kg	REAL	Scalar
PCB_105+132+153_MS	PCB congener 105+132+153 water column mass	kg	REAL	Scalar
PCB_151_MS	PCB congener 151 water column mass	kg	REAL	Scalar
PCB_138+163_MS	PCB congener 138+163 water column mass	kg	REAL	Scalar
PCB _170+190_MS	PCB congener 170+190 water column mass	kg	REAL	Scalar
PCB_172+197_MS	PCB congener 172+197 water column mass	kg	REAL	Scalar
PCB_180_MS	PCB congener 180 water column mass	kg	REAL	Scalar
PCB_182+187_MS	PCB congener 182+187 water column mass	kg	REAL	Scalar
PCB_195+208_MS	PCB congener 195+208 water column mass	kg	REAL	Scalar
PCB_196+203_MS	PCB congener 196+203 water column mass	kg	REAL	Scalar
PCB_201_MS	PCB congener 201 water column mass	kg	REAL	Scalar
PCB_146_MS	PCB congener 146 water column mass	kg	REAL	Scalar
PCB_85_MS	PCB congener 85 water column mass	kg	REAL	Scalar

The CDF zVariables corresponding to the PCB congener total load masses are given in Table 11.24.

PCB congeners are subject to settling in the water column because they partition to carbon-based states of which some settle. The list of the CDF zVariables corresponding to the PCB congener masses lost due to settling is given in Table 11.25 The list of the CDF zVariables corresponding to the PCB congener settling loss rates is given in Table 11.26

Volatilization is an important sink or source for PCB congeners in the lake. There are three separate volatilization process-related masses written to output. The transfer of PCBs from the air into the water column represents an absorption process and, therefore, an increase in the amount of PCB lake-wide masses. The list of the CDF zVariables corresponding to the lake-wide absorbed mass of a given PCB congener is given in Table 11.27. The transfer of PCBs from the water column to the atmosphere represents a loss of PCB mass from the lake. The CDF zVariables corresponding to these lost masses are given in Table 11.28.

The net transfer of PCBs across the air-water interface is represented by the CDF zVariables listed in Table 11.29.

11.2.6 Clay

The clay state and its output information was added to LM3. Clay is analogous to several of the eutrophication states in that it describes a suspended solid subject to settling. Concentration, lake-wide mass, loads, settling losses, and settling loss rates are calculated by LM3 and written to the output CDF. The corresponding zVariables are all summarized in Table 11.30. Unlike the eutrophication study, clay is presumed not to undergo any kinetics or transport-independent process, making it an ideal state variable for examining the simulation of high-resolution transport for the LMMBP.

11.2.7 Atrazine

Atrazine is one of the simplest of the LM3 state variables as it is subject only to advection, dispersion, volatilization, and a lumped-sum decay factor. Atrazine is not subject to settling in the water column. Consequently, it does not accumulate in the sediment bed. This means that no sediment transport is needed when running an atrazine simulation.

Atrazine state concentrations, lake-wide mass, and time-integrated load values are written to the output CDF and these are shown in Table 11.31 under the zVariables names of ATZ, ATZ_MS, and ATZ_LD, respectively.

Atrazine volatilization is written to the CDF output as three separate items. These items are the mass absorbed into the water column from the atmosphere, the mass lost from the water column into the atmosphere, and the net mass gained in the lake. These items are shown in Table 11.31 by the zVariable names of ATZ_VLT_IN, ATZ_VLT_OUT, and ATZ_VLT_NET, respectively.

The atrazine mass that is lost in the water column due to a lumped-sum decay is written to the output CDF under the zVariable name of ATZ_DECAY_MASS as shown in Table 11.31.

Table 11.24. PCB Congener Load Mass zVariables

zVariable	Description	Units	Data Type	Dimension
PCB_5+8_LD	PCB congener 5+8 load	kg	REAL	Scalar
PCB15+17_LD	PCB congener 15+17 load	kg	REAL	Scalar
PCB_16+32_LD	PCB congener 16+32 load	kg	REAL	Scalar
PCB_18_LD	PCB congener 18 load	kg	REAL	Scalar
PCB_26_LD	PCB congener 26 load	kg	REAL	Scalar
PCB_28+31_LD	PCB congener 28+31 load	kg	REAL	Scalar
PCB_33_LD	PCB congener 33 load	kg	REAL	Scalar
PCB_37+42_LD	PCB congener 37+42 load	kg	REAL	Scalar
PCB_44_LD	PCB congener 44 load	kg	REAL	Scalar
PCB_49_LD	PCB congener 49 load	kg	REAL	Scalar
PCB_52_LD	PCB congener 52 load	kg	REAL	Scalar
PCB_56+60_LD	PCB congener 56+60 load	kg	REAL	Scalar
PCB_66_LD	PCB congener 66 load	kg	REAL	Scalar
PCB_70+76_LD	PCB congener 70+76 load	kg	REAL	Scalar
PCB_74_LD	PCB congener 74 load	kg	REAL	Scalar
PCB_77+110_LD	PCB congener 77+110 load	kg	REAL	Scalar
PCB_81_LD	PCB congener 81 load	kg	REAL	Scalar
PCB_87_LD	PCB congener 87 load	kg	REAL	Scalar
PCB_84+92_LD	PCB congener 84+92 load	kg	REAL	Scalar
PCB_89_LD	PCB congener 89 load	kg	REAL	Scalar
PCB_99_LD	PCB congener 99 load	kg	REAL	Scalar
PCB_101_LD	PCB congener 101 load	kg	REAL	Scalar
PCB_118_LD	PCB congener 118 load	kg	REAL	Scalar
PCB_123+149_LD	PCB congener 123+149 load	kg	REAL	Scalar
PCB_105+132+153_LD	PCB congener 105+132+153 load	kg	REAL	Scalar
PCB_151_LD	PCB congener 151 load	kg	REAL	Scalar
PCB_138+163_LD	PCB congener 138+163 load	kg	REAL	Scalar
PCB _170+190_LD	PCB congener 170+190 load	kg	REAL	Scalar
PCB_172+197_LD	PCB congener 172+197 load	kg	REAL	Scalar
PCB_180_LD	PCB congener 180 load	kg	REAL	Scalar
PCB_182+187_LD	PCB congener 182+187 load	kg	REAL	Scalar
PCB_195+208_LD	PCB congener 195+208 load	kg	REAL	Scalar
PCB_196+203_LD	PCB congener 196+203 load	kg	REAL	Scalar
PCB_201_LD	PCB congener 201 load	kg	REAL	Scalar
PCB_146_LD	PCB congener 146 load	kg	REAL	Scalar
PCB_85_LD	PCB congener 85 load	kg	REAL	Scalar

Table 11.25. PCB Congener Settling Loss zVariables. Array Dimensions Refer to the Number of Surface Cells n_{sb}

zVariable	Description	Units	Data Type	Dimension
PCB_5+8_SET	PCB congener 5+8 settling loss	kg	REAL	$(1:n_{sb})$
PCB15+17_SET	PCB congener 15+17 settling loss	kg	REAL	$(1:n_{sb})$
PCB_16+32_SET	PCB congener 16+32 settling loss	kg	REAL	$(1:n_{sb})$
PCB_18_SET	PCB congener 18 settling loss	kg	REAL	$(1:n_{sb})$
PCB_26_SET	PCB congener 26 settling loss	kg	REAL	$(1:n_{sb})$
PCB_28+31_SET	PCB congener 28+31 settling loss	kg	REAL	$(1:n_{sb})$
PCB_33_SET	PCB congener 33 settling loss	kg	REAL	$(1:n_{sb})$
PCB_37+42_SET	PCB congener 37+42 settling loss	kg	REAL	$(1:n_{sb})$
PCB_44_SET	PCB congener 44 settling loss	kg	REAL	$(1:n_{sb})$
PCB_49_SET	PCB congener 49 settling loss	kg	REAL	$(1:n_{sb})$
PCB_52_SET	PCB congener 52 settling loss	kg	REAL	$(1:n_{sb})$
PCB_56+60_SET	PCB congener 56+60 settling loss	kg	REAL	$(1:n_{sb})$
PCB_66_SET	PCB congener 66 settling loss	kg	REAL	$(1:n_{sb})$
PCB_70+76_SET	PCB congener 70+76 settling loss	kg	REAL	$(1:n_{sb})$
PCB_74_SET	PCB congener 74 settling loss	kg	REAL	$(1:n_{sb})$
PCB_77+110_SET	PCB congener 77+110 settling loss	kg	REAL	$(1:n_{sb})$
PCB_81_SET	PCB congener 81 settling loss	kg	REAL	$(1:n_{sb})$
PCB_87_SET	PCB congener 87 settling loss	kg	REAL	$(1:n_{sb})$
PCB_84+92_SET	PCB congener 84+92 settling loss	kg	REAL	$(1:n_{sb})$
PCB_89_SET	PCB congener 89 settling loss	kg	REAL	$(1:n_{sb})$
PCB_99_SET	PCB congener 99 settling loss	kg	REAL	$(1:n_{sb})$
PCB_101_SET	PCB congener 101 settling loss	kg	REAL	$(1:n_{sb})$
PCB_118_SET	PCB congener 118 settling loss	kg	REAL	$(1:n_{sb})$
PCB_123+149_SET	PCB congener 123+149 settling loss	kg	REAL	$(1:n_{sb})$
PCB_105+132+153_SET	PCB congener 105+132+153 settling loss	kg	REAL	$(1:n_{sb})$
PCB_151_SET	PCB congener 151 settling loss	kg	REAL	$(1:n_{sb})$
PCB_138+163_SET	PCB congener 138+163 settling loss	kg	REAL	$(1:n_{sb})$
PCB _170+190_SET	PCB congener 170+190 settling loss	kg	REAL	$(1:n_{sb})$
PCB_172+197_SET	PCB congener 172+197 settling loss	kg	REAL	$(1:n_{sb})$
PCB_180_SET	PCB congener 180 settling loss	kg	REAL	$(1:n_{sb})$
PCB_182+187_SET	PCB congener 182+187 settling loss	kg	REAL	$(1:n_{sb})$
PCB_195+208_SET	PCB congener 195+208 settling loss	kg	REAL	$(1:n_{sb})$
PCB_196+203_SET	PCB congener 196+203 settling loss	kg	REAL	$(1:n_{sb})$
PCB_201_SET	PCB congener 201 settling loss	kg	REAL	$(1:n_{sb})$
PCB_146_SET	PCB congener 146 settling loss	kg	REAL	$(1:n_{sb})$
PCB_85_SET	PCB congener 85 settling loss	kg	REAL	$(1:n_{sb})$

Table 11.26. PCB Congener Settling Loss Rate zVariables. Array Dimensions Refer to the Number of Surface Cells n_{sb}

zVariable	Description	Units	Data Type	Dimension
PCB_5+8_SETRT	PCB congener 5+8 settling loss rate	kg	REAL	$(1:n_{sb})$
PCB15+17_SETRT	PCB congener 15+17 settling loss rate	kg	REAL	$(1:n_{sb})$
PCB_16+32_SETRT	PCB congener 16+32 settling loss rate	kg	REAL	$(1:n_{sb})$
PCB_18_SETRT	PCB congener 18 settling loss rate	kg	REAL	$(1:n_{sb})$
PCB_26_SETRT	PCB congener 26 settling loss rate	kg	REAL	$(1:n_{sb})$
PCB_28+31_SETRT	PCB congener 28+31 settling loss rate	kg	REAL	$(1:n_{sb})$
PCB_33_SETRT	PCB congener 33 settling loss rate	kg	REAL	$(1:n_{sb})$
PCB_37+42_SETRT	PCB congener 37+42 settling loss rate	kg	REAL	$(1:n_{sb})$
PCB_44_SETRT	PCB congener 44 settling loss rate	kg	REAL	$(1:n_{sb})$
PCB_49_SETRT	PCB congener 49 settling loss rate	kg	REAL	$(1:n_{sb})$
PCB_52_SETRT	PCB congener 52 settling loss rate	kg	REAL	$(1:n_{sb})$
PCB_56+60_SETRT	PCB congener 56+60 settling loss rate	kg	REAL	$(1:n_{sb})$
PCB_66_SETRT	PCB congener 66 settling loss rate	kg	REAL	$(1:n_{sb})$
PCB_70+76_SETRT	PCB congener 70+76 settling loss rate	kg	REAL	$(1:n_{sb})$
PCB_74_SETRT	PCB congener 74 settling loss rate	kg	REAL	$(1:n_{sb})$
PCB_77+110_SETRT	PCB congener 77+110 settling loss rate	kg	REAL	$(1:n_{sb})$
PCB_81_SETRT	PCB congener 81 settling loss rate	kg	REAL	$(1:n_{sb})$
PCB_87_SETRT	PCB congener 87 settling loss rate	kg	REAL	$(1:n_{sb})$
PCB_84+92_SETRT	PCB congener 84+92 settling loss rate	kg	REAL	$(1:n_{sb})$
PCB_89_SETRT	PCB congener 89 settling loss rate	kg	REAL	$(1:n_{sb})$
PCB_99_SETRT	PCB congener 99 settling loss rate	kg	REAL	$(1:n_{sb})$
PCB_101_SETRT	PCB congener 101 settling loss rate	kg	REAL	$(1:n_{sb})$
PCB_118_SETRT	PCB congener 118 settling loss rate	kg	REAL	$(1:n_{sb})$
PCB_123+149_SETRT	PCB congener 123+149 settling loss rate	kg	REAL	$(1:n_{sb})$
PCB_105+132+153_SETRT	PCB congener 105+132+153 settling loss rate	kg	REAL	$(1:n_{sb})$
PCB_151_SETRT	PCB congener 151 settling loss rate	kg	REAL	$(1:n_{sb})$
PCB_138+163_SETRT	PCB congener 138+163 settling loss rate	kg	REAL	$(1:n_{sb})$
PCB _170+190_SETRT	PCB congener 170+190 settling loss rate	kg	REAL	$(1:n_{sb})$
PCB_172+197_SETRT	PCB congener 172+197 settling loss rate	kg	REAL	$(1:n_{sb})$
PCB_180_SETRT	PCB congener 180 settling loss rate	kg	REAL	$(1:n_{sb})$
PCB_182+187_SETRT	PCB congener 182+187 settling loss rate	kg	REAL	$(1:n_{sb})$
PCB_195+208_SETRT	PCB congener 195+208 settling loss rate	kg	REAL	$(1:n_{sb})$
PCB_196+203_SETRT	PCB congener 196+203 settling loss rate	kg	REAL	$(1:n_{sb})$
PCB_201_SETRT	PCB congener 201 settling loss rate	kg	REAL	$(1:n_{sb})$
PCB_146_SETRT	PCB congener 146 settling loss rate	kg	REAL	$(1:n_{sb})$
PCB_85_SETRT	PCB congener 85 settling loss rate	kg	REAL	$(1:n_{sb})$

Table 11.27. Volatilization Absorbed Mass Output zVariables

zVariable	Description	Units	Data Type	Dimension
PCB_5+8_VLT_IN	PCB congener 5+8 volatilization absorption mass	kg	DOUBLE	Scalar
PCB15+17_VLT_IN	PCB congener 15+17 volatilization absorption mass	kg	DOUBLE	Scalar
PCB_16+32_VLT_IN	PCB congener 16+32 volatilization absorption mass	kg	DOUBLE	Scalar
PCB_18_VLT_IN	PCB congener 18 volatilization absorption mass	kg	DOUBLE	Scalar
PCB_26_VLT_IN	PCB congener 26 volatilization absorption mass	kg	DOUBLE	Scalar
PCB_28+31_VLT_IN	PCB congener 28+31 volatilization absorption mass	kg	DOUBLE	Scalar
PCB_33_VLT_IN	PCB congener 33 volatilization absorption mass	kg	DOUBLE	Scalar
PCB_37+42_VLT_IN	PCB congener 37+42 volatilization absorption mass	kg	DOUBLE	Scalar
PCB_44_VLT_IN	PCB congener 44 volatilization absorption mass	kg	DOUBLE	Scalar
PCB_49_VLT_IN	PCB congener 49 volatilization absorption mass	kg	DOUBLE	Scalar
PCB_52_VLT_IN	PCB congener 52 volatilization absorption mass	kg	DOUBLE	Scalar
PCB_56+60_VLT_IN	PCB congener 56+60 volatilization absorption mass	kg	DOUBLE	Scalar
PCB_66_VLT_IN	PCB congener 66 volatilization absorption mass	kg	DOUBLE	Scalar
PCB_70+76_VLT_IN	PCB congener 70+76 volatilization absorption mass	kg	DOUBLE	Scalar
PCB_74_VLT_IN	PCB congener 74 volatilization absorption mass	kg	DOUBLE	Scalar
PCB_77+110_VLT_IN	PCB congener 77+110 volatilization absorption mass	kg	DOUBLE	Scalar
PCB_81_VLT_IN	PCB congener 81 volatilization absorption mass	kg	DOUBLE	Scalar
PCB_87_VLT_IN	PCB congener 87 volatilization absorption mass	kg	DOUBLE	Scalar
PCB_84+92_VLT_IN	PCB congener 84+92 volatilization absorption mass	kg	DOUBLE	Scalar
PCB_89_VLT_IN	PCB congener 89 volatilization absorption mass	kg	DOUBLE	Scalar
PCB_99_VLT_IN	PCB congener 99 volatilization absorption mass	kg	DOUBLE	Scalar
PCB_101_VLT_IN	PCB congener 101 volatilization absorption mass	kg	DOUBLE	Scalar
PCB_118_VLT_IN	PCB congener 118 volatilization absorption mass	kg	DOUBLE	Scalar
PCB_123+149_VLT_IN	PCB congener 123+149 volatilization absorption mass	kg	DOUBLE	Scalar
PCB_105+132+153_VLT_IN	PCB congener 105+132+153 volatilization absorption mass	kg	DOUBLE	Scalar
PCB_151_VLT_IN	PCB congener 151 volatilization absorption mass	kg	DOUBLE	Scalar
PCB_138+163_VLT_IN	PCB congener 138+163 volatilization absorption mass	kg	DOUBLE	Scalar
PCB _170+190_VLT_IN	PCB congener 170+190 volatilization absorption mass	kg	DOUBLE	Scalar
PCB_172+197_VLT_IN	PCB congener 172+197 volatilization absorption mass	kg	DOUBLE	Scalar
PCB_180_VLT_IN	PCB congener 180 volatilization absorption mass	kg	DOUBLE	Scalar
PCB_182+187_VLT_IN	PCB congener 182+187 volatilization absorption mass	kg	DOUBLE	Scalar
PCB_195+208_VLT_IN	PCB congener 195+208 volatilization absorption mass	kg	DOUBLE	Scalar
PCB_196+203_VLT_IN	PCB congener 196+203 volatilization absorption mass	kg	DOUBLE	Scalar
PCB_201_VLT_IN	PCB congener 201 volatilization absorption mass	kg	DOUBLE	Scalar
PCB_146_VLT_IN	PCB congener 146 volatilization absorption mass	kg	DOUBLE	Scalar
PCB_85_VLT_IN	PCB congener 85 volatilization absorption mass	kg	DOUBLE	Scalar

Table 11.28. Volatilization Loss Mass Output zVariables

zVariable	Description	Units	Data Type	Dimension
PCB_5+8_VLT_OUT	PCB congener 5+8 volatilization lost mass	kg	DOUBLE	Scalar
PCB15+17_VLT_OUT	PCB congener 15+17 volatilization lost mass	kg	DOUBLE	Scalar
PCB_16+32_VLT_OUT	PCB congener 16+32 volatilization lost mass	kg	DOUBLE	Scalar
PCB_18_VLT_OUT	PCB congener 18 volatilization lost mass	kg	DOUBLE	Scalar
PCB_26_VLT_OUT	PCB congener 26 volatilization lost mass	kg	DOUBLE	Scalar
PCB_28+31_VLT_OUT	PCB congener 28+31 volatilization lost mass	kg	DOUBLE	Scalar
PCB_33_VLT_OUT	PCB congener 33 volatilization lost mass	kg	DOUBLE	Scalar
PCB_37+42_VLT_OUT	PCB congener 37+42 volatilization lost mass	kg	DOUBLE	Scalar
PCB_44_VLT_OUT	PCB congener 44 volatilization lost mass	kg	DOUBLE	Scalar
PCB_49_VLT_OUT	PCB congener 49 volatilization lost mass	kg	DOUBLE	Scalar
PCB_52_VLT_OUT	PCB congener 52 volatilization lost mass	kg	DOUBLE	Scalar
PCB_56+60_VLT_OUT	PCB congener 56+60 volatilization lost mass	kg	DOUBLE	Scalar
PCB_66_VLT_OUT	PCB congener 66 volatilization lost mass	kg	DOUBLE	Scalar
PCB_70+76_VLT_OUT	PCB congener 70+76 volatilization lost mass	kg	DOUBLE	Scalar
PCB_74_VLT_OUT	PCB congener 74 volatilization lost mass	kg	DOUBLE	Scalar
PCB_77+110_VLT_OUT	PCB congener 77+110 volatilization lost mass	kg	DOUBLE	Scalar
PCB_81_VLT_OUT	PCB congener 81 volatilization lost mass	kg	DOUBLE	Scalar
PCB_87_VLT_OUT	PCB congener 87 volatilization lost mass	kg	DOUBLE	Scalar
PCB_84+92_VLT_OUT	PCB congener 84+92 volatilization lost mass	kg	DOUBLE	Scalar
PCB_89_VLT_OUT	PCB congener 89 volatilization lost mass	kg	DOUBLE	Scalar
PCB_99_VLT_OUT	PCB congener 99 volatilization lost mass	kg	DOUBLE	Scalar
PCB_101_VLT_OUT	PCB congener 101 volatilization lost mass	kg	DOUBLE	Scalar
PCB_118_VLT_OUT	PCB congener 118 volatilization lost mass	kg	DOUBLE	Scalar
PCB_123+149_VLT_OUT	PCB congener 123+149 volatilization lost mass	kg	DOUBLE	Scalar
PCB_105+132+153_VLT_OUT	PCB congener 105+132+153 volatilization lost mass	kg	DOUBLE	Scalar
PCB_151_VLT_OUT	PCB congener 151 volatilization lost mass	kg	DOUBLE	Scalar
PCB_138+163_VLT_OUT	PCB congener 138+163 volatilization lost mass	kg	DOUBLE	Scalar
PCB _170+190_VLT_OUT	PCB congener 170+190 volatilization lost mass	kg	DOUBLE	Scalar
PCB_172+197_VLT_OUT	PCB congener 172+197 volatilization lost mass	kg	DOUBLE	Scalar
PCB_180_VLT_OUT	PCB congener 180 volatilization lost mass	kg	DOUBLE	Scalar
PCB_182+187_VLT_OUT	PCB congener 182+187 volatilization lost mass	kg	DOUBLE	Scalar
PCB_195+208_VLT_OUT	PCB congener 195+208 volatilization lost mass	kg	DOUBLE	Scalar
PCB_196+203_VLT_OUT	PCB congener 196+203 volatilization lost mass	kg	DOUBLE	Scalar
PCB_201_VLT_OUT	PCB congener 201 volatilization lost mass	kg	DOUBLE	Scalar
PCB_146_VLT_OUT	PCB congener 146 volatilization lost mass	kg	DOUBLE	Scalar
PCB_85_VLT_OUT	PCB congener 85 volatilization lost mass	kg	DOUBLE	Scalar

Table 11.29. Net Volatilization Mass Output zVariables

zVariable	Description	Units	Data Type	Dimension
PCB_5+8_VLT_NET	PCB congener 5+8 net volatilization mass	kg	DOUBLE	Scalar
PCB15+17_VLT_NET	PCB congener 15+17 net volatilization mass	kg	DOUBLE	Scalar
PCB_16+32_VLT_NET	PCB congener 16+32 net volatilization mass	kg	DOUBLE	Scalar
PCB_18_VLT_NET	PCB congener 18 net volatilization mass	kg	DOUBLE	Scalar
PCB_26_VLT_NET	PCB congener 26 net volatilization mass	kg	DOUBLE	Scalar
PCB_28+31_VLT_NET	PCB congener 28+31 net volatilization mass	kg	DOUBLE	Scalar
PCB_33_VLT_NET	PCB congener 33 net volatilization mass	kg	DOUBLE	Scalar
PCB_37+42_VLT_NET	PCB congener 37+42 net volatilization mass	kg	DOUBLE	Scalar
PCB_44_VLT_NET	PCB congener 44 net volatilization mass	kg	DOUBLE	Scalar
PCB_49_VLT_NET	PCB congener 49 net volatilization mass	kg	DOUBLE	Scalar
PCB_52_VLT_NET	PCB congener 52 net volatilization mass	kg	DOUBLE	Scalar
PCB_56+60_VLT_NET	PCB congener 56+60 net volatilization mass	kg	DOUBLE	Scalar
PCB_66_VLT_NET	PCB congener 66 net volatilization mass	kg	DOUBLE	Scalar
PCB_70+76_VLT_NET	PCB congener 70+76 net volatilization mass	kg	DOUBLE	Scalar
PCB_74_VLT_NET	PCB congener 74 net volatilization mass	kg	DOUBLE	Scalar
PCB_77+110_VLT_NET	PCB congener 77+110 net volatilization mass	kg	DOUBLE	Scalar
PCB_81_VLT_NET	PCB congener 81 net volatilization mass	kg	DOUBLE	Scalar
PCB_87_VLT_NET	PCB congener 87 net volatilization mass	kg	DOUBLE	Scalar
PCB_84+92_VLT_NET	PCB congener 84+92 net volatilization mass	kg	DOUBLE	Scalar
PCB_89_VLT_NET	PCB congener 89 net volatilization mass	kg	DOUBLE	Scalar
PCB_99_VLT_NET	PCB congener 99 net volatilization mass	kg	DOUBLE	Scalar
PCB_101_VLT_NET	PCB congener 101 net volatilization mass	kg	DOUBLE	Scalar
PCB_118_VLT_NET	PCB congener 118 net volatilization mass	kg	DOUBLE	Scalar
PCB_123+149_VLT_NET	PCB congener 123+149 net volatilization mass	kg	DOUBLE	Scalar
PCB_105+132+153_VLT_NET	PCB congener 105+132+153 net volatilization mass	kg	DOUBLE	Scalar
PCB_151_VLT_NET	PCB congener 151 net volatilization mass	kg	DOUBLE	Scalar
PCB_138+163_VLT_NET	PCB congener 138+163 net volatilization mass	kg	DOUBLE	Scalar
PCB _170+190_VLT_NET	PCB congener 170+190 net volatilization mass	kg	DOUBLE	Scalar
PCB_172+197_VLT_NET	PCB congener 172+197 net volatilization mass	kg	DOUBLE	Scalar
PCB_180_VLT_NET	PCB congener 180 net volatilization mass	kg	DOUBLE	Scalar
PCB_182+187_VLT_NET	PCB congener 182+187 net volatilization mass	kg	DOUBLE	Scalar
PCB_195+208_VLT_NET	PCB congener 195+208 net volatilization mass	kg	DOUBLE	Scalar
PCB_196+203_VLT_NET	PCB congener 196+203 net volatilization mass	kg	DOUBLE	Scalar
PCB_201_VLT_NET	PCB congener 201 net volatilization mass	kg	DOUBLE	Scalar
PCB_146_VLT_NET	PCB congener 146 net volatilization mass	kg	DOUBLE	Scalar
PCB_85_VLT_NET	PCB congener 85 net volatilization mass	kg	DOUBLE	Scalar

Table 11.30. Clay zVariables. Array Dimensions Refer to the Number of Cells n_c and the Number of Surface Cells n_{sb}

zVariable	Description	Units	Data Type	Dimension
CLAY	Clay concentration	$kg \cdot m^{-3}$	REAL	$(l{:}n_c)$
CLAY_MS	Clay mass	kg	REAL	Scalar
CLAY_LD	Clay load	kg	REAL	Scalar
CLAY_SET	Clay settling loss	$kg \cdot cell^{-1}$	REAL	$(1{:}n_{sb})$
CLAY_SETRT	Clay settling rate	$kg \cdot cell^{-1} \cdot s^{-1}$	REAL	$(1{:}n_{sb})$

Table 11.31. Atrazine zVariables. Array Dimensions Refer to the Number of Cells

zVariable	Description	Units	Data Type	Dimension
ATZ	Atrazine concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
ATZ_MS	Atrazine mass	kg	REAL	Scalar
ATZ_LD	Atrazine load	kg	REAL	Scalar
ATZ_VLT_IN	Atrazine volatilization absorption	kg	REAL	Scalar
ATZ_VLT_OUT	Atrazine volatilization loss	kg	REAL	Scalar
ATZ_VLT_NET	Atrazine net volatilization	kg	REAL	Scalar
ATZ_DECAY_MASS	Atrazine decay mass	kg	REAL	Scalar
ATZ_CHIFL_MASS	Atrazine mass lost through the Chicago River outflow	kg	REAL	Scalar
ATZ_MACFL_MASS	Atrazine mass lost through the Straits of Mackinac outflow	kg	REAL	Scalar
ATZ_OUTFL_MASS	Atrazine mass lost through the outflows	kg	REAL	Scalar

The amount of atrazine that is lost in the lake through the outflows is written to the output CDF as three separate items. These items are the Chicago River outflow, the Straits of Mackinac outflow, and the combined Chicago River-Straits of Mackinac outflows which are shown in Table 11.31 under the zVariable names of ATZ_CHIFL_MASS, ATZ_MACFL_MASS, and ATZ_OUTFL_MASS, respectively.

11.2.8 Chloride

Chloride is the other conservative LM3 state variable as it is subject only to advection and dispersion. Like atrazine, chloride is not subject to settling in the water column. Consequently, there is no need to use any kind of sediment transport to run a chloride simulation.

Chloride state concentrations, lake-wide mass, and time-integrated load values are written to the output CDF, and these are shown in Table 11.32 under the zVariable names of CL, CL_MS, and CL_LD, respectively.

11.2.9 Sediment

Sediment transport for the branches 3.2.15, 3.2.18, and 3.2.19 of LM3 has been developed with some limitations. The eutrophication part of the sediment bed is not complete because no kinetic processes have been implemented as of this writing. In addition, only the concentrations of the carbon state variables, DOC and refractory organic carbon (ROC), are written to the output CDF. Their corresponding zVariables are shown in Table 11.33. The other eutrophication states shown in Table 11.33 represent placeholders for their sediment concentrations in anticipation that a sediment diagenesis model will be developed at LLRS.

The only contaminants that are included in the sediment bed are PCBs. PCB congeners sorb to carbon. Consequently, PCB congener sediment concentrations are written to the output CDF as three separate items. These items are the total, dissolved, and particulate concentrations of the PCB congener. The list of the CDF zVariables corresponding to the PCB congener total concentrations is given in Table 11.34. The list of CDF zVariables corresponding to the PCB

Table 11.32. Chloride zVariables. Array Dimensions Refer to the Number of Cells n_c

zVariable	Description	Units	Data Type	Dimension
CL	Chloride concentration	$kg \cdot m^{-3}$	REAL	$(1:n_c)$
CL_MS	Chloride mass	kg	REAL	Scalar
CL_LD	Chloride load	kg	REAL	Scalar

Table 11.33. Sediment Eutrophication Output zVariables. Array Dimensions Refer to the Number of Sediment Cells n_{sed}

zVariable	Description	Units	Data Type	Dimension
DOC_SED	Dissolved organic carbon concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
ROC_SED	Refractory particulate organic carbon concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
SRP_SED	Soluble reactive phosphorus concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
DOP_SED	Dissolved organic phosphorus concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
ROP_SED	Refractory particulate organic phosphorus concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
NH4_SED	Ammonia concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
NO3_SED	Nitrate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
DON_SED	Dissolved organic nitrogen concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
RON_SED	Refractory particulate organic nitrogen concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
SA_SED	Available silica concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
SU_SED	Particulate biogenic silica concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$

Table 11.34. Sediment PCB Output zVariables. Array Dimensions Refer to the Number of Sediment Cells n_{sed}

zVariable	Description	Units	Data Type	Dimension
PCB_5+8_SEDTC	PCB congener 5+8 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB15+17_SEDTC	PCB congener 15+17 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_16+32_SEDTC	PCB congener 16+32 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_18_SEDTC	PCB congener 18 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_26_SEDTC	PCB congener 26 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_28+31_SEDTC	PCB congener 28+31 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_33_SEDTC	PCB congener 33 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_37+42_SEDTC	PCB congener 37+42 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_44_SEDTC	PCB congener 44 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_49_SEDTC	PCB congener 49 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_52_SEDTC	PCB congener 52 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_56+60_SEDTC	PCB congener 56+60 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_66_SEDTC	PCB congener 66 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_70+76_SEDTC	PCB congener 70+76 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_74_SEDTC	PCB congener 74 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_77+110_SEDTC	PCB congener 77+110 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_81_SEDTC	PCB congener 81 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_87_SEDTC	PCB congener 87 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_84+92_SEDTC	PCB congener 84+92 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_89_SEDTC	PCB congener 89 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_99_SEDTC	PCB congener 99 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_101_SEDTC	PCB congener 101 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_118_SEDTC	PCB congener 118 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_123+149_SEDTC	PCB congener 123+149 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_105+132+153_SEDTC	PCB congener 105+132+153 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_151_SEDTC	PCB congener 151 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_138+163_SEDTC	PCB congener 138+163 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB _170+190_SEDTC	PCB congener 170+190 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_172+197_SEDTC	PCB congener 172+197 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_180_SEDTC	PCB congener 180 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_182+187_SEDTC	PCB congener 182+187 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_195+208_SEDTC	PCB congener 195+208 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_196+203_SEDTC	PCB congener 196+203 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_201_SEDTC	PCB congener 201 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_146_SEDTC	PCB congener 146 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_85_SEDTC	PCB congener 85 total concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$

congener dissolved concentrations is given in Table 11.35. The list of the CDF zVariables corresponding to the PCB congener particulate concentrations is given in Table 11.36.

There is a placeholder for clay sediment concentrations in the program but as of this writing, clay is not being simulated. Nevertheless, the CDF zVariable corresponding to the clay sediment concentration is given in Table 11.37.

Table 11.35. Sediment PCB Dissolved Output zVariables. Array Dimensions Refer to the Number of Sediment Cells n_{sed}

zVariable	Description	Units	Data Type	Dimension
PCB_5+8_SEDDC	PCB congener 5+8 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB15+17_SEDDC	PCB congener 15+17 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_16+32_SEDDC	PCB congener 16+32 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_18_SEDDC	PCB congener 18 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_26_SEDDC	PCB congener 26 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_28+31_SEDDC	PCB congener 28+31 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_33_SEDDC	PCB congener 33 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_37+42_SEDDC	PCB congener 37+42 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_44_SEDDC	PCB congener 44 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_49_SEDDC	PCB congener 49 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_52_SEDDC	PCB congener 52 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_56+60_SEDDC	PCB congener 56+60 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_66_SEDDC	PCB congener 66 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_70+76_SEDDC	PCB congener 70+76 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_74_SEDDC	PCB congener 74 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_77+110_SEDDC	PCB congener 77+110 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_81_SEDDC	PCB congener 81 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_87_SEDDC	PCB congener 87 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_84+92_SEDDC	PCB congener 84+92 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_89_SEDDC	PCB congener 89 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_99_SEDDC	PCB congener 99 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_101_SEDDC	PCB congener 101 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_118_SEDDC	PCB congener 118 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_123+149_SEDDC	PCB congener 123+149 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_105+132+153_SEDDC	PCB congener 105+132+153 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_151_SEDDC	PCB congener 151 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_138+163_SEDDC	PCB congener 138+163 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB _170+190_SEDDC	PCB congener 170+190 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_172+197_SEDDC	PCB congener 172+197 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_180_SEDDC	PCB congener 180 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_182+187_SEDDC	PCB congener 182+187 dissolved concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$

Table 11.36. Sediment PCB Particulate Output zVariables. Array Dimensions Refer to the Number of Sediment Cells n_{sed}

zVariable	Description	Units	Data Type	Dimension
PCB_5+8_SEDPC	PCB congener 5+8 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB15+17_SEDPC	PCB congener 15+17 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_16+32_SEDPC	PCB congener 16+32 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_18_SEDPC	PCB congener 18 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_26_SEDPC	PCB congener 26 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_28+31_SEDPC	PCB congener 28+31 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_33_SEDPC	PCB congener 33 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_37+42_SEDPC	PCB congener 37+42 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_44_SEDPC	PCB congener 44 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_49_SEDPC	PCB congener 49 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_52_SEDPC	PCB congener 52 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_56+60_SEDPC	PCB congener 56+60 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_66_SEDPC	PCB congener 66 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_70+76_SEDPC	PCB congener 70+76 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_74_SEDPC	PCB congener 74 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_77+110_SEDPC	PCB congener 77+110 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_81_SEDPC	PCB congener 81 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_87_SEDPC	PCB congener 87 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_84+92_SEDPC	PCB congener 84+92 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_89_SEDPC	PCB congener 89 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_99_SEDPC	PCB congener 99 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_101_SEDPC	PCB congener 101 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_118_SEDPC	PCB congener 118 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_123+149_SEDPC	PCB congener 123+149 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_105+132+153_SEDPC	PCB congener 105+132+153 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_151_SEDPC	PCB congener 151 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_138+163_SEDPC	PCB congener 138+163 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB _170+190_SEDPC	PCB congener 170+190 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_172+197_SEDPC	PCB congener 172+197 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_180_SEDPC	PCB congener 180 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_182+187_SEDPC	PCB congener 182+187 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_195+208_SEDPC	PCB congener 195+208 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_196+203_SEDPC	PCB congener 196+203 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_201_SEDPC	PCB congener 201 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_146_SEDPC	PCB congener 146 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$
PCB_85_SEDPC	PCB congener 85 particulate concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$

Table 11.37. Sediment Output zVariables. Array Dimensions Refer to the Number of Sediment Cells $_{sed}$

zVariable	Description	Units	Data Type	Dimension
CLAY_SED	Clay concentration	$kg \cdot m^{-3}$	REAL	$(1:n_{sed})$

CHAPTER 12

SUPPORT UTILITIES

The input and output data handling requirements of the LM3 calculation precluded terminal entry and cut-paste of program output. The Common Data Format (CDF) simplified the task of preparing program input considerably because utility programs could be written that directly interacted with the target data structure, i.e., the input files of the simulation. Similarly, large volumes of program output could be handled using automated data handling techniques that hide the LM3 data structures from the user.

A simple program was written that translates ASCII data into the CDF format read by the LM3 simulation. The program a2cdf is used to help prepare the load and initial condition data currently used by LM3. A similar tool named x2cdf was written to translate from the C library XDR data format into CDF variables.

A variety of programs and applications have been prepared by the other users and institutions for reading CDF data files, and some of these have been applied to interpreting and processing both the input and output of the LM3 simulation. These include National Space Science Data Center (NSSDC) CDF applications cdfedit and cdfexport as well as custom-designed graphics packages written for IDL and Scilab.

12.1 Editing and Filling CDF Structures

The a2cdf program was written to fill CDF variables using ASCII formatted data as input. The tool is used to fill the LM3 input data structures for initial conditions and load CDF files. The format of the command is

a2cdf -R|-Z -c|-e|-i|-j|-r -n <array_size> <variable> <cdf>

The flags -R and -Z were used to select output to either the CDF rVariable type as used in the LM3 initial conditions file or the zVariable type as used in the load files. One of these flags must be provided. The next group of flags specifies the creation of a character variable -c, an epoch or calendar date value -e, an integer value -i, a Julian day number (JDN) -j, or a real CDF variable -r. The creation of CDF epoch and Julian time values requires that the user supply the calendar month, day, and year as input. The next argument array_size, specifies the number of successive input values that are to be grouped together as the consecutive elements of an array that will constitute a record of the output CDF. For example, setting this parameter equal to two causes a2cdf to write two input values into each record of the output CDF, forming an array with

two elements in each record. The variable argument specifies the name of the CDF variable being written. Finally, the CDF argument specifies the name of the CDF output file, without the .cdf extension, to be filled. A more complete example demonstrating the use of this command is provided in Section 13.5.

The x2cdf program is less generic than a2cdf and is specifically adapted to fill zVariables in the LM3 hydrodynamic input fields corresponding to radiation, heat fluxes, and wave information. This program takes FIREAD formatted input parameter data from standard input stdin and has the format:

```
x2cdf   <   <input_file>
```

The program is specifically written to work with the XDR format of meteorology data created for Lake Michigan studies by the Great Lakes Environmental Research Laboratory (GLERL). Parameter keys read in the input file supply the x2cdf conversion tool with the names of XDR input files to be processed, the name of the output CDF file to be filled, the name of the zVariable to be filled, whether flow faces or cells are to be written, the number of records to be processed, the number of input XDR records to be averaged for each output CDF record, the option to skip the first input XDR record, and the name of the GLERL bathymetry grid file used to define the geometry. A summary of the FIREAD keys corresponding to these parameters is provided in Table 12.1. An example demonstrating the use of x2cdf is given in Section 13.6.

Table 12.1. x2cdf FIREAD Input Key Definitions

Key	Description	Data Type
BOUNDS	Designates placement with respect to a cell or boundary face 1: Align with cell center 2: Align with cell x-face 3: Align with cell y-face	INTEGER
CDFFILE	Output CDF file name	CHARACTER
FNAME	Input XDR file name	CHARACTER
GBATH	Input bathymetry file name	CHARACTER
ISIGMA	Sigma level to sample	INTEGER
NRAV	Number of input XDR records to be averaged for each output CDF record	INTEGER
NREC	Number of records in each XDR file	INTEGER
SCALE	Scale factor by which input is multiplied to produce output	REAL
SKIP0	Flag designating skip of first input XDR record 0 - do not skip the first input XDR record of each file 1 - skip the first input XDR record of each file	INTEGER
ZVARNAME	zVariable name in the output CDF to be filled	CHARACTER

12.2 Examining Program Output

The LM3 simulation can generate large amounts of data ranging from a few megabytes to tens of gigabytes, depending on the user's choice of output, spatial segmentation, and state variables. Output generated on the segmentation level of the hydrodynamic grid is particularly challenging and time-consuming to analyze, especially when dealing with gigabytes worth of output; simple editing of such a volume of output for the purpose of performing analysis is neither practical nor convenient. However, individual data points can be examined and program output can be perused for debugging and other purposes.

The NSSDC group supplied two applications for interrogating CDF data structures and examining the element-by-element and variable-by-variable output of the program. The format of the command is:

cdfedit <file_name>

The cdfedit tool (National Aeronautics and Space Administration, 1996a,b,c) was used for examining detailed information concerning CDF data files. It can be used to provide a list of all variables described within the file, the number of records present, and display individual data items corresponding to specific records, variables, and variable array elements. The cdfedit tool is also useful for displaying documentation concerning each of the data variables; such as, a description of the variable, its designated units, and allowed minimum and maximum values. The cdfedit runs within the user's VT terminal window and is navigated using arrow keys and keyboard entry. Screenshots of the cdfedit can be viewed in Figures 12.1 and 12.2.

Figure 12.1 displays a window containing four sections: label, header, items, and trailer. The label is a general title for the window we are looking at: in this case it reads "CDF "bdump" zVariables", indicating that the name of the CDF file is "bdump" and that it contains zVariables. The header section shows the number of variables contained in the file, the maximum number of records that are output, and descriptive labels. The items section shows a listing of seven columns: variable names, data-types, dimensionality of the data, variance of the data, "zEntries" or metadata, numerical values, and "more" which contains extra information such as compression and padding values. The trailer section shows a menu of options and their corresponding keys that are available to the user.

Figure 12.2 displays a window that also contains four sections (similarly to Figure 12.1): label, header, items, and trailer. This window displays the metadata of one selected variable: in this case "DIA", which stands for diatoms. The diatoms metadata displayed contains five items: full name of the variable, units, format, and valid minimum and maximum values.

The cdfexport tool (National Aeronautics and Space Administration, 1996a,b,c) is similar to cdfedit in its implementation. The application is operated within the user's terminal window and is navigated using keyboard and arrow keys. The format of the command is:

cdfexport <file_name>

```
wmelende on superior: /mnt/model/jobs/94
                          CDF "bdump" zVariables
87 zVariables, (maximum of 6095 zRecords)
Name              DataSpec  Dimensions        Varys

"TIME_JULIAN"     REAL8/1   0:[]              T/      <zEntries> <values> <more>
"TIME_EPOCH"      EPOCH/1   0:[]              T/      <zEntries> <values> <more>
"TIME_RUNSEC"     REAL8/1   0:[]              T/      <zEntries> <values> <more>
"VOL"             REAL8/1   1:[44042]         T/T     <zEntries> <values> <more>
"TEM"             REAL4/1   1:[44042]         T/T     <zEntries> <values> <more>
"X_CURRENT"       REAL4/1   1:[44042]         T/T     <zEntries> <values> <more>
"Y_CURRENT"       REAL4/1   1:[44042]         T/T     <zEntries> <values> <more>
"AVGDIFF"         REAL4/1   1:[44042]         T/T     <zEntries> <values> <more>
"AVGZDIFF"        REAL4/1   1:[44042]         T/T     <zEntries> <values> <more>
"DOC"             REAL4/1   1:[44042]         T/T     <zEntries> <values> <more>
"DIA"             REAL4/1   1:[44042]         T/T     <zEntries> <values> <more>
"GRE"             REAL4/1   1:[44042]         T/T     <zEntries> <values> <more>
"ZOO"             REAL4/1   1:[44042]         T/T     <zEntries> <values> <more>
                                                                              Top
Select: <Return>   Create variable: <Ctrl-A>   Help: <Ctrl-K>
Exit:   <Ctrl-E>   Delete variable: <Ctrl-D>
```

Figure 12.1. The cdfedit tool main window.

```
wmelende on superior: /mnt/model/jobs/94
                          zVariable "DIA" zEntries
5 vAttributes
AttrName          DataSpec  Value(s)

"FIELDNAME"       CHAR/20   "Diatom Concentration"
"UNITS"           CHAR/7    "kg/m**3"
"FORMAT"
"VALIDMIN"        REAL4/1   -1.0e-10
"VALIDMAX"        REAL4/1   1.0e+24

                                                                              All
Select: <Return>   Create attribute: <Ctrl-A>   Next variable: <Ctrl-J>
Exit:   <Ctrl-E>   Create entry:     <Ctrl-B>   Delete entry:  <Ctrl-D>
Help:   <Ctrl-K>
```

Figure 12.2. The cdfedit tool zEntries window.

This program supports the translation of entire CDF groups (variables, records, or the contents of the entire CDF[1]) to the terminal or to an ASCII file. A screenshot of cdfexport terminal session is provided in Figure 12.3.

Figure 12.3 displays a window containing four individually-labeled sections: "CDFexport", "SelectionWindow, part 1", "Key Definitions", and "MessageBuffer". The "CDFexport" section shows the name of the CDF file, the number of variables and records, and whether compression of the data was performed. The "SelectionWindow, part 1" section shows all the variables available for export to the terminal or to an ASCII file. Note that the variable "ZOO" (which stands for zooplankton) is the only one chosen to be exported or translated. The "KeyDefinitions" section displays the available functions and their corresponding keys. The "MessageBuffer" section shows errors/instructions as necessary.

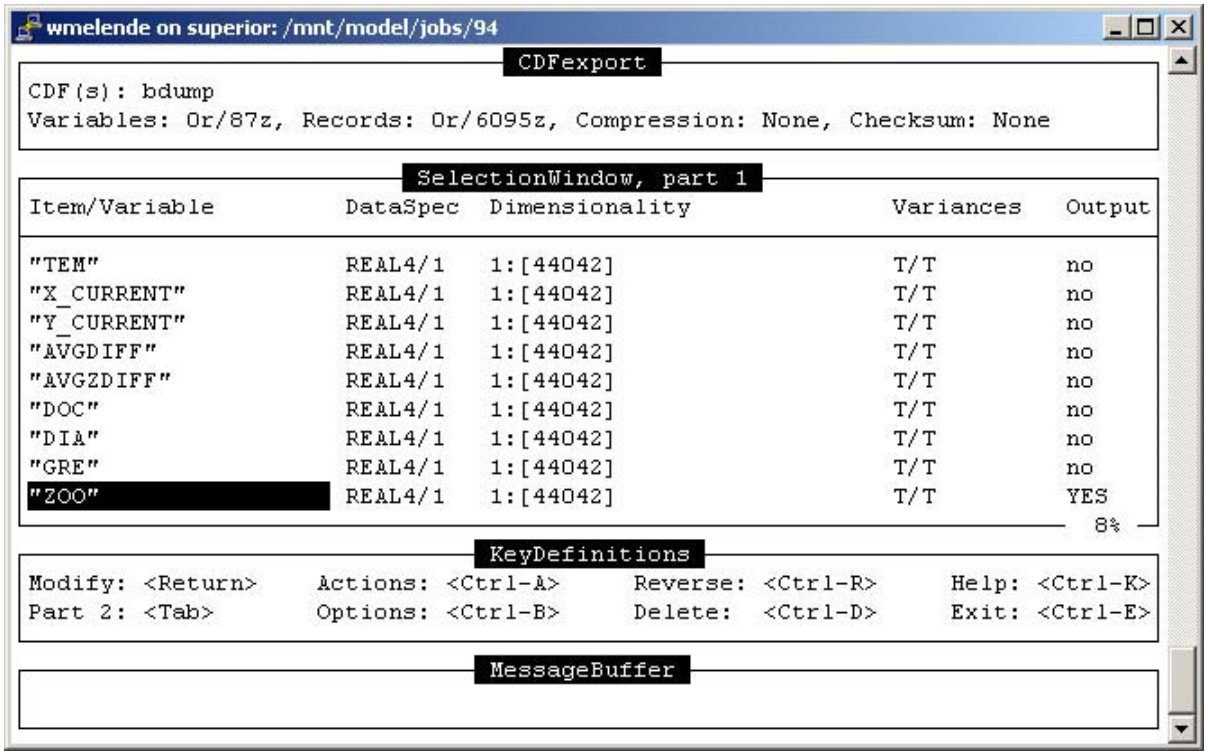


Figure 12.3. The cdfexport tool window.

[1]Again, the size of the LM3 output can be of the order of gigabytes and the option of exporting all of the program output to an ASCII file is not recommended.

CHAPTER 13

EXAMPLES

Unlike similar water quality programs that run in the terminal foreground, the LM3 calculation runs in the background in batch mode. Executing the program is typically time-consuming; it is most conveniently detached from the user's terminal when running. A batch submission tool has been prepared to run programs like LM3 whose run times preclude interactive execution and whose use of computer space is extensive. It distributes the computing load and organizes the resulting jobs in an orderly way. The resulting jobs are consumptive of disk space. It is not uncommon, for instance, for high-resolution water quality model jobs to take up to five gigabytes of space. The batch submission tool is also useful for organizing the multiple program input and output files for different instances of the execution, such as a large number of model calibration runs.

13.1 Submitting LM3 Simulations

The batch submission script /home/llrs_model/bin/submit has been written to provide a user-friendly interface to the Distributed Queuing System (DQS) batch submission qsub and to organize execution and output of the resulting batch jobs into sequentially ordered directories for future reference. Its broader goal is to organize both computer time and disk usage in an orderly fashion. The format of the command is:

```
submit  -q  <queue> -e  <exec>  [<input_file>...]
```

where queue can be Linux computers such as ausable_queue, erie_queue, michigan_queue, superior_queue, or huron_queue; exec refers to the executable to be run; and input_file... is a list of the input files to be passed for processing by the program in question. The first input file is designated by the submission script as being read on standard input; i.e., exec <input_file.

An example of the batch submission command that can be used to run the program is:

```
submit  -q  ausable_queue  -e  lm3-3.2.13  firead.inp
```

where the name of the batch queue, the name of the executable program (in this case, the LM3 binary name) and the name of the input file are specified. In instances where batch submission of the LM3 calculation is not desired, the program can be run directly at the user's home directory using the syntax

```
./lm3-3.2.13  <  firead.inp  >&  job.log  &
```

where the program binary, the input file name, and the output log file are specified. In each instance, the program binary and input file names may be varied to accommodate the different versions of each that are available to the user.

Specific examples of LM3 input files are provided below. The studies for atrazine, chloride, eutrophication, and polychlorinated biphenyl (PCB) congener (with the sediment bed) simulations are discussed, and example input files are provided. In each case, the Linux syntax for running the respective model calculation is as described above, using the batch submission method or running the program directly on the command line from the user's home directory.

13.2 LM3 Example Atrazine Calculation

The atrazine calculation provides one of the simpler simulations that can be run in LM3.

The input file shown below demonstrates the following:

1. The selection of the atrazine study, simulation of water column transport, mass balance calculations, and calculation of atrazine volatilization are specified.

2. Basic program control options including selection of the starting and ending simulation times are demonstrated.

3. The parameters that specify the starting and ending days for 1994 and 1995 for which the atrazine loads are to be scaled are input.

4. Atrazine decay rate and molecular weight are input.

5. Required input files including hydrodynamics and simulation geometry are input.

6. The atrazine load file is provided. This automatically enables the simulation of mass loads for this state variable.

7. Scale factors for each river and for the atmosphere are applied to the loads.

A copy of this file is available at /mnt/model/llrs_model/production/LM3/examples/firead_atrazine.inp as of this writing.

```
LIST                          ! Echo the contents of this file

C-----------------------------------------------------------------------------------------------------------
C  Example LM3 Atrazine simulation input deck
C-----------------------------------------------------------------------------------------------------------
*DO ATZ                       ! Perform atrazine calculations
*DO TRANS                     ! Perform water column transport
*DO MASSBALANCE               ! Perform mass balance calculations
*DO VOLAT                     ! Calculate volatilization
C-----------------------------------------------------------------------------------------------------------
```

```
C  Controlling program output, simulation times
C-----------------------------------------------------------------------------------------------------------------------
OVERWRITE 1                    ! Overwrite previous output dumps
CDFCOARSE 1                    ! Write coarse (41-segment) concentrations
CDFFINE 1                      ! Write fine (5-km) concentrations
DD2WRT 1                       ! How often to write output [days]
TMSTRT 0.0                     ! First simulated day (starts at zero)
TMEND 1825.0                   ! Last simulated data
C-----------------------------------------------------------------------------------------------------------------------
C  Starting and ending days that specify when to scale atrazine loads in 1994 and 1995
C-----------------------------------------------------------------------------------------------------------------------
START_DAY_SCALE_LOAD_94 104.0    ! Starting day of scaling loads in 1994
END_DAY_SCALE_LOAD_94 193.0      ! Ending day of scaling loads in 1994
START_DAY_SCALE_LOAD_95 469.0    ! Starting day of scaling loads in 1995
END_DAY_SCALE_LOAD_95 558.0      ! Ending day of scaling loads in 1995
C-----------------------------------------------------------------------------------------------------------------------
C  Increase the warning and error concentration limits
C-----------------------------------------------------------------------------------------------------------------------
*CMAXIMUM WARNING ATZ     1.0E+24
*CMAXIMUM ERROR  ATZ      1.0E+24
C-----------------------------------------------------------------------------------------------------------------------
C  Set Atrazine decay rate
C-----------------------------------------------------------------------------------------------------------------------
DECAY_RATE_ATRAZINEE 2.854E-10   ! Lumped decay rate constant for atrazine in units of
                                 ! 1/second
C-----------------------------------------------------------------------------------------------------------------------
C  Set Atrazine molecular weight
C-----------------------------------------------------------------------------------------------------------------------
MOLWT_ATZ 215.69               ! Molecular weight of atrazine in grams/mole
C-----------------------------------------------------------------------------------------------------------------------
C  Set water column transport Crank-Nicholson parameter
C-----------------------------------------------------------------------------------------------------------------------
TH 0.55                        ! Crank-Nicholson parameter
C-----------------------------------------------------------------------------------------------------------------------
C  Input files
C-----------------------------------------------------------------------------------------------------------------------
*FILE MAP                      /usr/model/dsets/LM3/R0105/geo/map.dat
*FILE GEO                      /usr/model/dsets/LM3/R0105/geo/geo.dat
*FILE AGG                      /usr/model/dsets/LM3/R0105/agg/map1312.dat
*FILE HYDRO                    /usr/model/dsets/LM3/R0105/3h/hydro94.cdf
*FILE ICI                      /usr/model/dsets/LM3/R0105/ici/ici_eutro_pcb.cdf
*FILE WIND                     /usr/model/dsets/LM3/R0105/WIND/wind_speed_level_3.cdf
*FILE AIR_CONC                 /usr/model/dsets/LM3/R0105/CHEM_AIR/VAPOR_
                               CONCENTRATION/air_concs_level_3_at z9.cdf
C-----------------------------------------------------------------------------------------------------------------------
C  Loads
C-----------------------------------------------------------------------------------------------------------------------
*FILE LDLOC                    /usr/model/dsets/LM3/R0105/ld/ldlocations.cdf
*FILE LOAD ATZ                 /usr/model/dsets/LM3/R0105/ld/ld_atz_041221.cdf
```

```
C-----------------------------------------------------------------------------------------------------------------------
C  Scale the loads
C-----------------------------------------------------------------------------------------------------------------------
*SCALE  1.0                    LOAD_MENOMINEE  ATZ
*SCALE  1.0                    LOAD_FOX  ATZ
*SCALE  1.0                    LOAD_SHEBOYGAN  ATZ
*SCALE  1.0                    LOAD_MILWAUKEE  ATZ
*SCALE  1.0                    LOAD_GRAND_CALUMET  ATZ
*SCALE  1.0                    LOAD_ST_JOSEPH  ATZ
*SCALE  1.0                    LOAD_KALAMAZOO  ATZ
*SCALE  1.0                    LOAD_GRAND  ATZ
*SCALE  1.0                    LOAD_MUSKEGON  ATZ
*SCALE  1.0                    LOAD_PERE_MARQUETTE  ATZ
*SCALE  1.0                    LOAD_MANISTIQUE  ATZ
*SCALE  1.0                    LOAD_CEDAR  ATZ
*SCALE  1.0                    LOAD_PESHTIGO  ATZ
*SCALE  1.0                    LOAD_OCONTO  ATZ
*SCALE  1.0                    LOAD_PENSAUKEE  ATZ
*SCALE  1.0                    LOAD_KEWAUNEE  ATZ
*SCALE  1.0                    LOAD_MANITOWOC  ATZ
*SCALE  1.0                    LOAD_ROOT  ATZ
*SCALE  1.0                    LOAD_BLACK_SH  ATZ
*SCALE  1.0                    LOAD_MANISTEE  ATZ
*SCALE  1.0                    LOAD_BETSIE  ATZ
*SCALE  1.0                    LOAD_JORDAN  ATZ
*SCALE  1.0                    LOAD_STURGEON  ATZ
*SCALE  1.0                    LOAD_WHITEFISH  ATZ
*SCALE  1.0                    LOAD_ESCANABA  ATZ
*SCALE  1.0                    LOAD_WET_ATMOSPHERIC  ATZ
*SCALE  1.0                    LOAD_MILECOQUINS  ATZ
*SCALE  1.0                    LOAD_GALIEN  ATZ

STOP
```

13.3 LM3 Example Chloride Calculation

The chloride calculation demonstrated here is similar to the case for atrazine. The chloride simulation is the simplest one that can be run in LM3.

The input file shown below demonstrates the following:

1. Selection of the chloride study, simulation of water column transport, and the activation of mass balance calculations are specified.

2. Basic program output options are input. The starting and ending days of the simulation are specified.

3. Required input files including hydrodynamics and simulation geometry are input.

4. The chloride load file is provided. This automatically enables the simulation of mass loads for this state variable.

5. The scaling factors are applied to the monitored, unmonitored, and atmospheric loads.

A copy of this file is at /mnt/model/llrs_model/production/LM3/examples/ firead_chloride.inp as of this writing.

```
LIST                          ! Echo the contents of this file

C-----------------------------------------------------------------------------------------------------------
C  Example LM3 Chloride simulation input deck
C-----------------------------------------------------------------------------------------------------------
C-----------------------------------------------------------------------------------------------------------
C  Basic program options
C-----------------------------------------------------------------------------------------------------------
*DO CL
*DO TRANS
*DO MASSBALANCE
C-----------------------------------------------------------------------------------------------------------
C  Controlling program output, simulation times
C-----------------------------------------------------------------------------------------------------------
OVERWRITE 1                   ! Overwrite previous output dumps
CDFCOARSE 1                   ! Write coarse (41-segment) concentrations
CDFFINE 1                     ! Write fine (5-km) concentrations
DD2WRT 1                      ! How often to write output [days]
TMSTRT 0.0                    ! First simulated day (starts at zero)
TMEND  730..0                 ! Last simulated data
C-----------------------------------------------------------------------------------------------------------
C  Set water column transport Crank-Nicholson parameter
C-----------------------------------------------------------------------------------------------------------
TH 0.55
C-----------------------------------------------------------------------------------------------------------
C  Input files
C-----------------------------------------------------------------------------------------------------------
*FILE MAP                     /usr/llrs_model/dsets/LM3/R0105/geo/map.dat
*FILE GEO                     /usr/llrs_model/dsets/LM3/R0105/geo/geo.dat
*FILE AGG                     /usr/llrs_model/dsets/LM3/R0105/agg/map1312.dat
*FILE HYDRO                   /usr/llrs_model/dsets/LM3/R0105/3h/hydro94.cdf
*FILE ICI                     /usr/LLRS_model/dsets/LM3/R0105/ici/ici_eutro_pcb.cdf
C-----------------------------------------------------------------------------------------------------------
C  Loads
C-----------------------------------------------------------------------------------------------------------
*FILE LDLOC                   /home/llrs_model/dsets/LM3/R0105/ld/ldlocations.cdf
*FILE LOAD CL                 /home/llrs_model/dsets/LM3/R0105/ld/ld_cl.cdf
C-----------------------------------------------------------------------------------------------------------
C  Scale the loads
C-----------------------------------------------------------------------------------------------------------
*SCALE  1.1                   LOAD_MON  CL
```

```
*SCALE  1.2                    LOAD_UNMON  CL
*SCALE  1.5                    LOAD_ATM  CL

STOP
```

13.4 LM3 Example Eutrophication Calculation

The eutrophication calculation is more complex than those for the atrazine or chloride studies because of the number of state variables involved, there are 17 of them, and the relatively large size of parameters available to it, which number over 100 (Table 6.1).

The input file shown below demonstrates the following:

1. Selection of the eutrophication study, the simulation of water column transport, the activation of mass balance calculations, and the enabling of growth limitation factors output are specified.

2. Basic program output control options are input. The option to overwrite previously generated output Common Data Format (CDF) files is enabled. The modeled time interval is set to include the study period of 1994 and 1995.

3. The warning and error concentration limits are increased for all the eutrophication state variables.

4. Nitrogen has been turned off as a nutrient limiting factor.

5. A fairly large number of parameters, which are used only for the eutrophication, are input. Most of these parameters are related to the eutrophication kinetics.

6. The settling velocities of the eutrophication states that settle are input.

7. Required input files including hydrodynamics and simulation geometry are input.

8. The eutrophication load files are provided. This automatically enables the simulation of mass loads for the state variables.

9. The eutrophication state information is used to create biotic carbon loads for the level 2 water quality simulation, and file results are directed to the ASCII output file I2prod.dat.

10. Scaling factors for the eutrophication loads are input. Most values are set to the default values of one except for the silica loads, the soluble reactive phosphorus (SRP) sediment load, and the ammonia-nitrogen (NH_4) sediment load.

A copy of this file is available at /mnt/model/llrs_model/production/LM3/examples/ firead_eutrophication.inp as of this writing.

```
LIST                    ! Echo the contents of this file

C-----------------------------------------------------------------------------------------------------------
C  Example LM3 Eutrophication simulation input deck
C-----------------------------------------------------------------------------------------------------------
C-----------------------------------------------------------------------------------------------------------
C  Basic program options
C-----------------------------------------------------------------------------------------------------------
*DO EUTRO                   ! Perform eutrophication calculations
*DO TRANS                   ! Perform water column transport
*DO MASSBALANCE             ! Perform mass balance calculations
*DO GRLIM                   ! Enable the output of growth limitation factors
C-----------------------------------------------------------------------------------------------------------
C  Controlling program output, simulation times
C-----------------------------------------------------------------------------------------------------------
OVERWRITE 1                 ! Overwrite previous output dumps
CDFCOARSE 1                 ! Write coarse (41-segment) concentrations?
CDFFINE 1                   ! Write fine (5-km) concentrations?
DD2WRT 1                    ! How often to write output [days]
TMSTRT 0.0                  ! First simulated day (starts at zero)
TMEND  730.0                ! Last simulated data
C-----------------------------------------------------------------------------------------------------------
C  Increase the warning and error concentration limits
C-----------------------------------------------------------------------------------------------------------
*CMAXIMUM WARNING   DOC   1.0E+24
*CMAXIMUM WARNING   DIA   1.0E+24
*CMAXIMUM WARNING   GRE   1.0E+24
*CMAXIMUM WARNING   ZOO   1.0E+24
*CMAXIMUM WARNING   LOC   1.0E+24
*CMAXIMUM WARNING   ROC   1.0E+24
*CMAXIMUM WARNING   SRP   1.0E+24
*CMAXIMUM WARNING   DOP   1.0E+24
*CMAXIMUM WARNING   LOP   1.0E+24
*CMAXIMUM WARNING   ROP   1.0E+24
*CMAXIMUM WARNING   NH4   1.0E+24
*CMAXIMUM WARNING   NO3   1.0E+24
*CMAXIMUM WARNING   DON   1.0E+24
*CMAXIMUM WARNING   LON   1.0E+24
*CMAXIMUM WARNING   RON   1.0E+24
*CMAXIMUM WARNING   SA    1.0E+24
*CMAXIMUM WARNING   SU    1.0E+24
*CMAXIMUM ERROR     DOC   1.0E+24
*CMAXIMUM ERROR     DIA   1.0E+24
*CMAXIMUM ERROR     GRE   1.0E+24
*CMAXIMUM ERROR     ZOO   1.0E+24
*CMAXIMUM ERROR     LOC   1.0E+24
*CMAXIMUM ERROR     ROC   1.0E+24
*CMAXIMUM ERROR     SRP   1.0E+24
*CMAXIMUM ERROR     DOP   1.0E+24
```

```
*CMAXIMUM ERROR      LOP    1.0E+24
*CMAXIMUM ERROR      ROP    1.0E+24
*CMAXIMUM ERROR      NH4    1.0E+24
*CMAXIMUM ERROR      NO3    1.0E+24
*CMAXIMUM ERROR      DON    1.0E+24
*CMAXIMUM ERROR      LON    1.0E+24
*CMAXIMUM ERROR      RON    1.0E+24
*CMAXIMUM ERROR      SA     1.0E+24
*CMAXIMUM ERROR      SU     1.0E+24
C-----------------------------------------------------------------------------------------------------------------
C  Set the nutrient limiting factors
C  SILIM  ==  1  Nitrogen and Phosphorus
C  SILIM  ==  2  Nitrogen, Phosphorus, and Silica
C  SILIM  ==  3  Phosphorus and Silica
C-----------------------------------------------------------------------------------------------------------------
SILIM  3
C-----------------------------------------------------------------------------------------------------------------
C  Set the values of the eutrophication parameters
C-----------------------------------------------------------------------------------------------------------------
ANCP  0.25                    ! N:C ratio
APCP  0.01                    ! P:C ratio
ASCD  2.3                     ! Si:C ratio
BMRD 8.60E-07                 ! Diatom mortality
BMRG 8.60E-07                 ! Greens mortality
CCHLD  40                     ! C:chl ratio for diatoms
CCHLG  4.0                    ! C:chl ratio for greens
CGZ  3.10E-06                 ! Zooplankton grazing rate
FCDD  0.05                    ! Fraction mortality = dissolved organic carbon (diatom)
FCDG  0.05                    ! Fraction mortality = dissolved organic carbon (greens)
FCLD  0.3                     ! Fraction mortality = labile organic carbon (diatom)
FCLG  0.3                     ! Fraction mortality = labile organic carbon (greens)
FCRD  0.3                     ! Fraction mortality = refractory organic carbon (diatom)
FCRG  0.3                     ! Fraction mortality = refractory organic carbon (greens)
FCDP  0.35                    ! Fraction predation = dissolved organic carbon
FCDZ  0                       ! Fraction z-mortality = dissolved organic carbon
FCLP  0.5                     ! Fraction predation = labile particulate detrital carbon
FCLZ  0.4                     ! Fraction z-mortality = labile particulate detrital carbon
FCRP  0.15                    ! Fraction predation = refractory particulate detrital carbon
FCRZ  0.1                     ! Fraction z-mortality = refractory particulate detrital carbon
FNDD  0.5                     ! Fraction mortality = dissolved organic nitrogen (diatom)
FNDG  0.5                     ! Fraction mortality = dissolved organic nitrogen (greens)
FNDP  0                       ! Fraction predation = dissolved organic nitrogen
FNDZ  0                       ! Fraction z-mortality = dissolved organic nitrogen
FNID  0.5                     ! Fraction mortality = dissolved inorganic nitrogen (diatom)
FNIG  0.5                     ! Fraction mortality = dissolved inorganic nitrogen (greens)
FNIP  0.5                     ! Fraction predation = dissolved inorganic nitrogen
FNIZ  0.5                     ! Fraction z-mortality = dissolved inorganic nitrogen
FNLD  0                       ! Fraction mortality = labile organic nitrogen (diatom)
FNLG  0                       ! Fraction mortality = labile organic nitrogen (greens)
```

```
FNLP  0.4                    ! Fraction predation = labile organic nitrogen
FNLZ  0.4                    ! Fraction z-mortality = labile organic nitrogen
FNRD  0                      ! Fraction mortality = refractory organic nitrogen (diatom)
FNRG  0                      ! Fraction mortality = refractory organic nitrogen (greens)
FNRP  0.1                    ! Fraction predation = refractory organic nitrogen
FNRZ  0.1                    ! Fraction z-mortality = refractory organic nitrogen
FPDD  0.10                   ! Fraction mortality = dissolved organic phosphorus (diatom)
FPDG  0.10                   ! Fraction mortality = dissolved organic phosphorus (greens)
FPDP  0.2                    ! Fraction predation = dissolved organic phosphorus
FPDZ  0.2                    ! Fraction z-mortality =  dissolved organic phosphorus
FPID  0.30                   ! Fraction mortality = dissolved inorganic phosphorus (diatom)
FPIG  0.30                   ! Fraction mortality = dissolved inorganic phosphorus  (greens)
FPIP  0.5                    ! Fraction predation = dissolved inorganic phosphorus
FPIZ  0.5                    ! Fraction z-mortality = dissolved inorganic phosphorus
FPLD  0.30                   ! Fraction mortality = labile organic phosphorus (diatom)
FPLG  0.30                   ! Fraction mortality = labile organic phosphorus (greens)
FPLP  0.15                   ! Fraction predation = labile organic phosphorus
FPLZ  0.15                   ! Fraction z-mortality labile organic phosphorus
FPRD  0.30                   ! Fraction mortality = refractory organic phosphorus (diatom)
FPRG  0.30                   ! Fraction mortality = refractory organic phosphorus (greens)
FPRP  0.15                   ! Fraction predation = refractory organic phosphorus
FPRZ  0.15                   ! Fraction z-mortality = refractory organic phosphorus
FSAP  0                      ! Fraction predation = dissolved silica
GREFF  0.6                   ! Zooplankton grazing coefficient
I0WT  0                      ! Weighted light day0
I1WT  0                      ! Weighted light day1
I2WT  0                      ! Weighted light day2
ILMUL  1                     ! Scaling for surface solar radiation
ILUMO  0                     ! Constant value of illumination
ISMIN  400                   ! Optimal illumination factor mineralization
KDC  1.16E-08                ! Dissolved organic carbon mineralization
KDCALG  0.00E+00             ! Dissolved organic carbon algae dependence
KDN  1.74E-07                ! Dissolved organic nitrogen mineralization
KDNALG  0.00E+00             ! Dissolved organic nitrogen algae dependence
KDP  1.16E-09                ! Dissolved organic phosphorus mineralization
KDPALG  6.00E-03             ! Dissolved organic phosphorus algae dependence
KE  0.15                     ! Background light attenuation
KECHL  1.7E+04               ! Light attenuation for chlorophyll-a
KHN  2.50E-05                ! Organic nitrogen decay
KHND  2.50E-05               ! Mean nitrogen half-saturation (diatom)
KHNG  2.50E-05               ! Mean nitrogen half-saturation (greens)
NHNNT  0.0000                ! Half-saturation of NH3 for nitrogen
KHP  5.00E-07                ! Organic phosphorus decay
KHPD  5.00E-07               ! Mean phosphorus half-saturation (diatom)
KHPG  5.00E-07               ! Mean phosphorus half-saturation (greens)
KHSD  6.00E-05               ! Mean silica half-saturation (diatom)
KLC  1.00E-07                ! Mineralization hydrolysis for labile organic carbon
KLCALG  0.00E+00             ! Labile organic carbon algae dependence
KLN  3.47E-07                ! Mineralization hydrolysis for labile organic nitrogen
```

```
KLNALG  0.00E+00          ! Labile organic nitrogen algae dependence
KLP  1.00E-09             ! Mineralization hydrolysis for labile organic phosphorus
KLPALG  6.00E-03          ! Labile organic phosphorus algae dependence
KRC  1.00E-07             ! Mineralization hydrolysis for refractory organic phosphorus
KRCALG  0.00E+00          ! Refractory organic carbon algae dependence
KRN  3.47E-08             ! Mineralization hydrolysis for refractory organic nitrogen
KRNALG  0.00E+00          ! Refractory organic nitrogen algae dependence
KRP  1.00E-09             ! Mineralization hydrolysis for refractory organic phosphorus
KRPALG  6.00E-03          ! Refractory organic phosphorus algae dependence
KSUA  2.50E-07            ! Biogenic silica algae dependence
KSZ  1.00E-04             ! Zooplankton half-saturation phytoplankton
KTBD  0.074               ! Diatom mortality temperature coefficient
KTBG  0.074               ! Greens mortality temperature coefficient
KTGD1  0.0025             ! Temperature coefficient < (diatom)
KTGD2  0.006              ! Temperature coefficient > (diatom)
KTGG1  0.0025             ! Temperature coefficient < (greens)
KTGG2  0.006              ! Temperature coefficient > (greens)
KTHDR  9.9E-02            ! Hydrolysis temperature dependence
KTMNL  7.4E-02            ! Mineralization temperature dependence
KTNT1  0.004              ! Temperature coefficient < nitrogen
KTNT2  0.004              ! Temperature coefficient > nitrogen
KTSUA  0.069              ! Dissolved silica temperature coefficient
NTM  2.50E-11             ! Nitrification rate
PMD  2.90E-05             ! Diatom production
PMG  2.60E-05             ! Greens production
TH  0.55                  ! Crank-Nicholson parameter
TMD  18                   ! Optimal temperature diatom
TMG  18                   ! Optimal temperature greens
TMNT  30                  ! Optimal temperature nitrogen
TRD  20                   ! Optimal metabolic temperature (diatom)
TRG  20                   ! Optimal metabolic temperature (greens)
TRHDR  20                 ! Optimal hydrolysis temperature
TRMNL  20                 ! Optimal mineralization temperature
TRSUA  20                 ! Optimal dissolved silica temperature
TZREF  20                 ! Optimal predation temperature
ZDTH  5.0E-07             ! Zooplankton mortality
ZTHET  1.l00              ! Temperature coefficient predation
C-----------------------------------------------------------------------------------------------------------------------
C  Settling velocities
C-----------------------------------------------------------------------------------------------------------------------
VDIA                      1.15E-06
VGRE                      0.85E-06
VLOC                      2.00E-06
VROC                      2.00E-06
VLON                      2.00E-06
VRON                      2.00E-06
VLOP                      2.00E-06
VROP                      2.00E-06
VSU                       2.00E-06
```

```
C-----------------------------------------------------------------------------------------------------------------------
C  Output aggregated BIC carbon for level 2 model
C-----------------------------------------------------------------------------------------------------------------------
*FILE  L2PROD  I2prod.dat
C-----------------------------------------------------------------------------------------------------------------------
C  Input files
C-----------------------------------------------------------------------------------------------------------------------
*FILE MAP                    /usr/model/dsets/LM3/R0105/geo/map.dat
*FILE GEO                    /usr/model/dsets/LM3/R0105/geo/geo.dat
*FILE AGG                    /usr/model/dsets/LM3/R0105/agg/map1312.dat
*FILE HYDRO                  /usr/model/dsets/LM3/R0105/3h_modified/hydro94.cdf
*FILE ICI                    /usr/model/dsets/LM3/R0105/ici/ici.cdf
C-----------------------------------------------------------------------------------------------------------------------
C  Loads
C-----------------------------------------------------------------------------------------------------------------------
*FILE LDLOC                  /usr/model/dsets/LM3/R0105/ld/ldlocations.cdf
*FILE LOAD DOC               /usr/model dsets/LM3/R0105/ld/ld_doc.cdf
*FILE LOAD DIA               /usr/model/dsets/LM3/R0105/ld/ld_dia.cdf
*FILE LOAD GRE               /usr/model/dsets/LM3/R0105/ld/ld_gre.cdf
*FILE LOAD LOC               /usr/model/dsets/LM3/R0105/ld/ld_lpoc.cdf
*FILE LOAD ROC               /usr/model/dsets/LM3/R0105/ld/ld_rpoc.cdf
*FILE LOAD SRP               /usr/model/dsets/LM3/R0105/ld/ld_srp.cdf
*FILE LOAD DOP               /usr/model/dsets/LM3/R0105/ld/ld_dop.cdf
*FILE LOAD LOP               /usr/model/dsets/LM3/R0105/ld/ld_lop.cdf
*FILE LOAD ROP               /usr/model/dsets/LM3/R0105/ld/ld_rop.cdf
*FILE LOAD NH4               /usr/model/dsets/LM3/R0105/ld/ld_nh4.cdf
*FILE LOAD NO3               /usr/model/dsets/LM3/R0105/ld/ld_no3.cdf
*FILE LOAD DON               /usr/model/dsets/LM3/R0105/ld/ld_don.cdf
*FILE LOAD LON               /usr/model/dsets/LM3/R0105/ld/ld_lon.cdf
*FILE LOAD RON               /usr/model/dsets/LM3/R0105/ld/ld_ron.cdf
*FILE LOAD SA                /usr/model/dsets/LM3/R0105/ld/ld_sa.cdf
C-----------------------------------------------------------------------------------------------------------------------
C  Scale the loads
C-----------------------------------------------------------------------------------------------------------------------
*SCALE    1.0       LOAD_MON DOC
*SCALE    1.0       LOAD_MON DIA
*SCALE    1.0       LOAD_MON GRE
*SCALE    1.0       LOAD_MON LOC
*SCALE    1.0       LOAD_MON ROC
*SCALE    1.0       LOAD_MON SRP
*SCALE    1.0       LOAD_MON DOP
*SCALE    1.0       LOAD_MON LOP
*SCALE    1.0       LOAD_MON ROP
*SCALE    1.0       LOAD_MON NH4
*SCALE    1.0       LOAD_MON NO3
*SCALE    1.0       LOAD_MON DON
*SCALE    1.0       LOAD_MON LON
*SCALE    1.0       LOAD_MON RON
*SCALE    1.075     LOAD_MON SA
```

```
*SCALE    1.0       LOAD_UNMON DOC
*SCALE    1.0       LOAD_UNMON DIA
*SCALE    1.0       LOAD_UNMON GRE
*SCALE    1.0       LOAD_UNMON LOC
*SCALE    1.0       LOAD_UNMON ROC
*SCALE    1.0       LOAD_UNMON SRP
*SCALE    1.0       LOAD_UNMON DOP
*SCALE    1.0       LOAD_UNMON LOP
*SCALE    1.0       LOAD_UNMON ROP
*SCALE    1.0       LOAD_UNMON NH4
*SCALE    1.0       LOAD_UNMON NO3
*SCALE    1.0       LOAD_UNMON DON
*SCALE    1.0       LOAD_UNMON LON
*SCALE    1.0       LOAD_UNMON RON
*SCALE    1.075     LOAD_UNMON SA
*SCALE    1.0       LOAD_ATM DOC
*SCALE    1.0       LOAD_ATM DIA
*SCALE    1.0       LOAD_ATM GRE
*SCALE    1.0       LOAD_ATM LOC
*SCALE    1.0       LOAD_ATM ROC
*SCALE    1.0       LOAD_ATM SRP
*SCALE    1.0       LOAD_ATM DOP
*SCALE    1.0       LOAD_ATM LOP
*SCALE    1.0       LOAD_ATM ROP
*SCALE    1.0       LOAD_ATM NH4
*SCALE    1.0       LOAD_ATM NO3
*SCALE    1.0       LOAD_ATM DON
*SCALE    1.0       LOAD_ATM LON
*SCALE    1.0       LOAD_ATM RON
*SCALE    1.1       LOAD_ATM SA
*SCALE    1.0       LOAD_SED DOC
*SCALE    1.0       LOAD_SED DIA
*SCALE    1.0       LOAD_SED GRE
*SCALE    1.0       LOAD_SED LOC
*SCALE    1.0       LOAD_SED ROC
*SCALE    26141.5   LOAD_SED SRP
*SCALE    1.0       LOAD_SED DOP
*SCALE    1.0       LOAD_SED LOP
*SCALE    1.0       LOAD_SED ROP
*SCALE    20093.0   LOAD_SED NH4
*SCALE    1.0       LOAD_SED NO3
*SCALE    1.0       LOAD_SED DON
*SCALE    1.0       LOAD_SED LON
*SCALE    1.0       LOAD_SED RON
*SCALE    30255.0   LOAD_SED SA
*SCALE    1.0       LOAD_SHORE_EROSION ROC

STOP
```

13.5 LM3 Example PCB Calculation

The PCB calculation is currently the most complex of the studies supported by LM3 because enabling this calculation also activates the eutrophication and the sediment transport.

The input file shown below demonstrates the following:

1. Selection of the PCB study, the selection of the eutrophication study, the simulation of water column transport, the activation of mass balance calculations, the output of growth limitation factors, the simulation of sediment transport, and the calculation of volatilization are specified.

2. Basic program output control options are input. The modeled time interval is set to include the study period of 1994 and 1995.

3. The warning and error concentration limits are increased for the eutrophication states and the PCB congener.

4. Nitrogen has been turned off as a nutrient limiting factor.

5. A fairly large set of eutrophication kinetics parameters are input.

6. Water column transport parameters as well as sediment transport parameters are also input.

7. Settling velocities for the eutrophication states that settle are input.

8. PCB congener-specific volatilization parameters are input.

9. PCB congener-specific partition coefficients are input.

10. Level 2 aggregated biotic inorganic carbon output is generated.

11. Input files for the water column, sediment bed, and the volatilization calculation are specified. The input files for the water column include the hydrodynamics and the simulation geometry.

12. The load files for the eutrophication states and the PCB congener are included.

13. Scaling factors for the loads are specified.

A copy of this file is at /mnt/model/llrs_model/production/LM3/examples/firead_pcb_congener.inp as of this writing.

```
LIST                                                    ! Echo the contents of this file

C-----------------------------------------------------------------------------------------------------------------------
C  Example LM3 PCB-congener simulation input deck
C-----------------------------------------------------------------------------------------------------------------------
```

```
C-----------------------------------------------------------------------------------------------------------------------
C  Basic program options
C-----------------------------------------------------------------------------------------------------------------------
*DO PCB_28+31                                    ! Simulate PCB congener 28+31
*DO EUTRO                                        ! Perform eutrophication calculations
*DO TRANS                                        ! Perform water column transport
*DO MASSBALANCE                                  ! Perform mass balance calculations
*DO GRLIM                                        ! Output growth limitation factors
*DO SEDIMENT                                     ! Perform sediment bed transport
*DO VOLAT                                        ! Calculate volatilization
C-----------------------------------------------------------------------------------------------------------------------
C  Controlling program output, simulation times
C-----------------------------------------------------------------------------------------------------------------------
OVERWRITE 1                                      ! Overwrite previous output dumps
CDFCOARSE 1                                      ! Write coarse (41-segment) concentrations
CDFFINE 0                                        ! Write fine (5-km) concentrations
DD2WRT 1                                         ! How often to write output [days]
TMSTRT 0.0                                       ! First simulated day (starts at zero)
TMEND  730.0                                     ! Last simulated data
C-----------------------------------------------------------------------------------------------------------------------
C  Increase the warning and error concentration limits
C-----------------------------------------------------------------------------------------------------------------------
*CMAXIMUM WARNING    DOC          1.0E+24
*CMAXIMUM WARNING    DIA          1.0E+24
*CMAXIMUM WARNING    GRE          1.0E+24
*CMAXIMUM WARNING    ZOO          1.0E+24
*CMAXIMUM WARNING    LOC          1.0E+24
*CMAXIMUM WARNING    ROC          1.0E+24
*CMAXIMUM WARNING    SRP          1.0E+24
*CMAXIMUM WARNING    DOP          1.0E+24
*CMAXIMUM WARNING    LOP          1.0E+24
*CMAXIMUM WARNING    ROP          1.0E+24
*CMAXIMUM WARNING    NH4          1.0E+24
*CMAXIMUM WARNING    NO3          1.0E+24
*CMAXIMUM WARNING    DON          1.0E+24
*CMAXIMUM WARNING    LON          1.0E+24
*CMAXIMUM WARNING    RON          1.0E+24
*CMAXIMUM WARNING    SA           1.0E+24
*CMAXIMUM WARNING    SU           1.0E+24
*CMAXIMUM WARNING    PCB_28+31    1.0E+24
*CMAXIMUM ERROR      DOC          1.0E+24
*CMAXIMUM ERROR      DIA          1.0E+24
*CMAXIMUM ERROR      GRE          1.0E+24
*CMAXIMUM ERROR      ZOO          1.0E+24
*CMAXIMUM ERROR      LOC          1.0E+24
*CMAXIMUM ERROR      ROC          1.0E+24
*CMAXIMUM ERROR      SRP          1.0E+24
*CMAXIMUM ERROR      DOP          1.0E+24
*CMAXIMUM ERROR      LOP          1.0E+24
```

```
*CMAXIMUM ERROR      ROP                 1.0E+24
*CMAXIMUM ERROR      NH4                 1.0E+24
*CMAXIMUM ERROR      NO3                 1.0E+24
*CMAXIMUM ERROR      DON                 1.0E+24
*CMAXIMUM ERROR      LON                 1.0E+24
*CMAXIMUM ERROR      RON                 1.0E+24
*CMAXIMUM ERROR      SA                  1.0E+24
*CMAXIMUM ERROR      SU                  1.0E+24
*CMAXIMUM ERROR      PCB_28+31_DIS       1.0E+24
*CMAXIMUM ERROR      PCB_28+31_PART      1.0E+24
C-- ------------------------------------------------------------------------------------------------------------------
C  Set the nutrient limiting factors
C  SILIM  ==  1  Nitrogen and Phosphorus
C  SILIM  ==  2  Nitrogen, Phosphorus, and Silica
C  SILIM  ==  3  Phosphorus and Silica
C------------------------------------------------------------------------------------------------------------------
SILIM  3
C------------------------------------------------------------------------------------------------------------------
C  Set the values of the eutrophication parameters
C------------------------------------------------------------------------------------------------------------------
ANCP  0.25                 ! N:C ratio
APCP  0.01                 ! P:C ratio
ASCD  2.3                  ! Si:C ratio
BMRD 8.60E-07              ! Diatom mortality
BMRG 8.60E-07              ! Greens mortality
CCHLD  40                  ! C:chl ratio for diatom
CCHLG  4.0                 ! C:chl ratio for greens
CGZ  3.10E-06              ! Zooplankton grazing rate
FCDD  0.05                 ! Fraction mortality = dissolved organic carbon (diatom)
FCDG  0.05                 ! Fraction mortality = dissolved organic carbon (greens)
FCLD  0.3                  ! Fraction mortality = labile organic carbon (diatom)
FCLG  0.3                  ! Fraction mortality = labile organic carbon (greens)
FCRD  0.3                  ! Fraction mortality = refractory organic carbon (diatom)
FCRG  0.3                  ! Fraction mortality = refractory organic carbon (greens)
FCDP  0.35                 ! Fraction predation = dissolved organic carbon
FCDZ  0                    ! Fraction z-mortality = dissolved organic carbon
FCLP  0.5                  ! Fraction predation = labile particulate detrital carbon
FCLZ  0.4                  ! Fraction z-mortality = labile particulate detrital carbon
FCRP  0.15                 ! Fraction predation = refractory particulate detrital carbon
FCRZ  0.1                  ! Fraction z-mortality = refractory particulate detrital carbon
FNDD  0.5                  ! Fraction mortality = dissolved organic nitrogen (diatom)
FNDG  0.5                  ! Fraction mortality = dissolved organic nitrogen (greens)
FNDP  0                    ! Fraction predation = dissolved organic nitrogen
FNDZ  0                    ! Fraction z-mortality = dissolved organic nitrogen
FNID  0.5                  ! Fraction mortality = dissolved inorganic nitrogen (diatom)
FNIG  0.5                  ! Fraction mortality = dissolved inorganic nitrogen (greens)
FNIP  0.5                  ! Fraction predation = dissolved inorganic nitrogen
FNIZ  0.5                  ! Fraction z-mortality = dissolved inorganic nitrogen
FNLD  0                    ! Fraction mortality = labile organic nitrogen (diatom)
```

```
FNLG  0              ! Fraction mortality = labile organic nitrogen (greens)
FNLP  0.4            ! Fraction predation = labile organic nitrogen
FNLZ  0.4            ! Fraction z-mortality = labile organic nitrogen
FNRD  0              ! Fraction mortality = refractory organic nitrogen (diatom)
FNRG  0              ! Fraction mortality = refractory organic nitrogen (greens)
FNRP  0.1            ! Fraction predation = refractory organic nitrogen
FNRZ  0.1            ! Fraction z-mortality = refractory organic nitrogen
FPDD  0.10           ! Fraction mortality = dissolved organic phosphorus (diatom)
FPDG  0.10           ! Fraction mortality = dissolved organic phosphorus (greens)
FPDP  0.2            ! Fraction predation= dissolved organic phosphorus
FPDZ  0.2            ! Fraction z-mortality = dissolved organic phosphorus
FPID  0.30           ! Fraction mortality = dissolved inorganic phosphorus (diatom)
FPIG  0.30           ! Fraction mortality = dissolved inorganic phosphorus (greens)
FPIP  0.5            ! Fraction predation = dissolved inorganic phosphorus
FPIZ  0.5            ! Fraction z-mortality = dissolved inorganic phosphorus
FPLD  0.30           ! Fraction mortality = labile organic phosphorus (diatom)
FPLG  0.30           ! Fraction mortality = labile organic phosphorus (greens)
FPLP  0.15           ! Fraction predation = labile organic phosphorus
FPLZ  0.15           ! Fraction z-mortality = labile organic phosphorus
FPRD  0.30           ! Fraction mortality = refractory organic phosphorus (diatom)
FPRG  0.30           ! Fraction mortality = refractory organic phosphorus (greens)
FPRP  0.15           ! Fraction predation = refractory organic phosphorus
FPRZ  0.15           ! Fraction z-mortality = refractory organic phosphorus
FSAP  0              ! Fraction predation = dissolved silica
GREFF  0.6           ! Zooplankton grazing coefficient
I0WT  0              ! Weighted light day0
I1WT  0              ! Weighted light day1
I2WT  0              ! Weighted light day2
ILMUL  1             ! Scaling illumination factor
ILUMO  0             ! Constant value of illumination
ISMIN  400           ! Optimal illumination factor mineralization
KDC  1.16E-08        ! Dissolved organic carbon mineralization
KDCALG  0.00E+00     ! Dissolved organic carbon algae dependence
KDN  1.74E-07        ! Dissolved organic nitrogen mineralization
KDNALG  0.00E+00     ! Dissolved organic nitrogen algae dependence
KDP  1.16E-09        ! Dissolved organic phosphorus mineralization
KDPALG  6.00E-03     ! Dissolved organic phosphorus algae dependence
KE  0.15             ! Background light attenuation
KECHL  1.7E+04       ! Light attenuation for chlorophyll-a
KHN  2.50E-05        ! Organic nitrogen decay
KHND  2.50E-05       ! Mean nitrogen half-saturation (diatom)
KHNG  2.50E-05       ! Mean nitrogen half-saturation (greens)
NHNNT  0.0000        ! Half-saturation of NH3 for nitrogen
KHP  5.00E-07        ! Organic phosphorus decay
KHPD  5.00E-07       ! Mean phosphorus half-saturation (diatom)
KHPG  5.00E-07       ! Mean phosphorus half-saturation (greens)
KHSD  6.00E-05       ! Mean silica half-saturation (diatom)
KLC  1.00E-07        ! Mineralization hydrolysis for labile organic carbon
KLCALG  0.00E+00     ! Labile organic carbon algae dependence
```

```
KLN  3.47E-07              ! Mineralization hydrolysis for labile organic nitrogen
KLNALG  0.00E+00           ! Labile organic nitrogen algae dependence
KLP  1.00E-09              ! Mineralization hydrolysis for labile organic phosphorus
KLPALG  6.00E-03           ! Labile organic phosphorus algae dependence
KRC  1.00E-07              ! Mineralization hydrolysis for refractory organic carbon
KRCALG  0.00E+00           ! Refractory organic carbon algae dependence
KRN  3.47E-08              ! Mineralization hydrolysis for refractory organic nitrogen
KRNALG  0.00E+00           ! Refractory organic nitrogen algae dependence
KRP  1.00E-09              ! Mineralization hydrolysis refractory organic phosphorus
KRPALG  6.00E-03           ! Refractory organic phosphorus algae dependence
KSUA  2.50E-07             ! Biogenic silica algae dependence
KSZ  1.00E-04              ! Zooplankton half-saturation phytoplankton
KTBD  0.074                ! Diatom mortality temperature coefficient
KTBG  0.074                ! Greens mortality temperature coefficient
KTGD1  0.0025              ! Temperature coefficient < (diatom)
KTGD2  0.006               ! Temperature coefficient > (diatom)
KTGG1  0.0025              ! Temperature coefficient < (greens)
KTGG2  0.006               ! Temperature coefficient > (greens)
KTHDR  9.9E-02             ! Hydrolysis temperature dependence
KTMNL  7.4E-02             ! Mineralization temperature dependence
KTNT1  0.004               ! Temperature coefficient < nitrogen
KTNT2  0.004               ! Temperature coefficient > nitrogen
KTSUA  0.069               ! Dissolved silica temperature coefficient
NTM  2.50E-11              ! Nitrification rate
PMD  2.90E-05              ! Diatom production
PMG  2.60E-05              ! Greens production
TMD  18                    ! Optimal temperature diatom
TMG  18                    ! Optimal temperature greens
TMNT  30                   ! Optimal temperature nitrogen
TRD  20                    ! Optimal metabolic temperature (diatom)
TRG  20                    ! Optimal metabolic temperature (greens)
TRHDR  20                  ! Optimal hydrolysis temperature
TRMNL  20                  ! Optimal mineralization temperature
TRSUA  20                  ! Optimal dissolved silica temperature
TZREF  20                  ! Optimal predation temperature
ZDTH  5.0E-07              ! Zooplankton mortality
ZTHET  1.l00               ! Temperature coefficient predation
C-----------------------------------------------------------------------------------------------------------------------
C  Water transport parameters
C-----------------------------------------------------------------------------------------------------------------------
TH  0.55                   ! Crank-Nicholson parameter
VBC  0                     ! Volume balance diagnostics flag
XYDFC  1                   ! Water column on/off horizontal diffusion flag
ZDFC  1                    ! Water column on/off vertical diffusion flag
C-----------------------------------------------------------------------------------------------------------------------
C  Sediment transport parameters
C-----------------------------------------------------------------------------------------------------------------------
DENSITY_ROC  1200.0            ! Refractory organic carbon density in kg/m**3
MAX_RESP_VEL  1.00E-08         ! Maximum resuspension velocity allowed in m/s
```

```
QMUL_SED  0.0                    ! Sediment flows scaling factor
SLC_SED  1                       ! Sediment bed QUICKEST on/off flag
TH_SED  0.0                      ! Sediment transport Crank-Nicholson parameter
TIME_STEP_SED-SCALE  0.05        ! Scaling factor for sediment bed time step
ZDF  0.0E+00                     ! Sediment vertical diffusion
ZDFMUL_SED  0.0                  ! Sediment vertical diffusion scaling factor
C-----------------------------------------------------------------------------------------------------------------------
C  Settling velocities
C-----------------------------------------------------------------------------------------------------------------------
VDIA                             1.15E-06
VGRE                             0.85E-06
VLOC                             2.00E-06
VROC                             2.00E-06
VLON                             2.00E-06
VRON                             2.00E-06
VLOP                             2.00E-06
VROP                             2.00E-06
VSU                              2.00E-06
C-----------------------------------------------------------------------------------------------------------------------
C  PCB congener-specific volatilization parameters
C-----------------------------------------------------------------------------------------------------------------------
MOLWT_PCB_28+31  260.54          ! Molecular weight of PCB 28+31
ENTROPY_PCB_28+31  0.09          ! Entropy of PCB 28+31
ENTHALPY_PCB_28+31  37           ! Enthalpy of PCB 28+31
C-----------------------------------------------------------------------------------------------------------------------
C  PCB congener-specific partition coefficients in m**3/kg
C-----------------------------------------------------------------------------------------------------------------------
PART_COEFF_PCB_28+31 DOC         13.4896
PART_COEFF_PCB_28+31_DIA         1348.96
PART_COEFF_PCB_28+31_GRE         1348.96
PART_COEFF_PCB_28+31_ZOO         1348.96
PART_COEFF_PCB_28+31_LOC         1348.96
PART_COEFF_PCB_28+31_ROC         1348.96
C-----------------------------------------------------------------------------------------------------------------------
C  Output aggregated BIC carbon for level 2 model
C-----------------------------------------------------------------------------------------------------------------------
*FILE  L2PROD  12prod.dat
C-----------------------------------------------------------------------------------------------------------------------
C  Water column Input files
C-----------------------------------------------------------------------------------------------------------------------
*FILE MAP                        /usr/model/dsets/LM3/R0105/geo/map.dat
*FILE GEO                        /usr/model/dsets/LM3/R0105/geo/geo.dat
*FILE AGG                        /usr/model/dsets/LM3/R0105/agg/map1312.dat
*FILE HYDRO                      /usr/model/dsets/LM3/R0105/3h/hydro94.cdf
*FILE ICI                        /usr/model/dsets/LM3/R0105/ici/ici_eutro_pcb.cdf
C-----------------------------------------------------------------------------------------------------------------------
C  Sediment bed input files
C-----------------------------------------------------------------------------------------------------------------------
```

```
*FILE_ICI_SED                    /usr/model/dsets/LM3/R0105/ici/ici_sed.cdf
*FILE VHYD                       /usr/model/dsets/LM3/R0105/geo/vhyd.inp
*FILE VMAP                       /usr/model/dsets/LM3/R0105/geo/vmap.inp
*FILE VGEO                       /usr/model/dsets/LM3/R0105/geo/vgeo.inp
*FILE RESUSP_VELOCITIES          /usr/model/dsets/LM3/R0105/RESUSP_FLUXES_VELS/
                                 resuspension_velocities.cdf
C-----------------------------------------------------------------------------------------------------------------
C  Volatilization input files
C-----------------------------------------------------------------------------------------------------------------
*FILE WIND                       /usr/model/dsets/LM3/R0105/WIND/wind_speed_level3.cdf
*FILE AIR_CONC                   /usr/model/dsets/LM3/R0105/CHEM_AIR/VAPOR_
                                 CONCENTRATION/air_concentrations_level_3.cdf
C-----------------------------------------------------------------------------------------------------------------
C  Loads
C-----------------------------------------------------------------------------------------------------------------
*FILE LDLOC                      /usr/model/dsets/LM3/R0105/ld/ldlocations_pcb.cdf
*FILE LOAD DOC                   /usr/model dsets/LM3/R0105/ld/ld_doc.cdf
*FILE LOAD DIA                   /usr/model/dsets/LM3/R0105/ld/ld_dia.cdf
*FILE LOAD GRE                   /usr/model/dsets/LM3/R0105/ld/ld_gre.cdf
*FILE LOAD LOC                   /usr/model/dsets/LM3/R0105/ld/ld_lpoc.cdf
*FILE LOAD ROC                   /usr/model/dsets/LM3/R0105/ld/ld_rpoc.cdf
*FILE LOAD SRP                   /usr/model/dsets/LM3/R0105/ld/ld_srp.cdf
*FILE LOAD DOP                   /usr/model/dsets/LM3/R0105/ld/ld_dop.cdf
*FILE LOAD LOP                   /usr/model/dsets/LM3/R0105/ld/ld_lop.cdf
*FILE LOAD ROP                   /usr/model/dsets/LM3/R0105/ld/ld_rop.cdf
*FILE LOAD NH4                   /usr/model/dsets/LM3/R0105/ld/ld_nh4.cdf
*FILE LOAD NO3                   /usr/model/dsets/LM3/R0105/ld/ld_no3.cdf
*FILE LOAD DON                   /usr/model/dsets/LM3/R0105/ld/ld_don.cdf
*FILE LOAD LON                   /usr/model/dsets/LM3/R0105/ld/ld_lon.cdf
*FILE LOAD RON                   /usr/model/dsets/LM3/R0105/ld/ld_ron.cdf
*FILE LOAD SA                    /usr/model/dsets/LM3/R0105/ld/ld_sa.cdf
*FILE LOAD PCB_28+31             /usr/model/dsets/LM3/R0105/ld/ld_pcb_28+31.cdf
C-----------------------------------------------------------------------------------------------------------------
C  Scale the loads
C-----------------------------------------------------------------------------------------------------------------
*SCALE    1.0        LOAD_MON DOC
*SCALE    1.0        LOAD_MON DIA
*SCALE    1.0        LOAD_MON GRE
*SCALE    1.0        LOAD_MON LOC
*SCALE    1.0        LOAD_MON ROC
*SCALE    1.0        LOAD_MON SRP
*SCALE    1.0        LOAD_MON DOP
*SCALE    1.0        LOAD_MON LOP
*SCALE    1.0        LOAD_MON ROP
*SCALE    1.0        LOAD_MON NH4
*SCALE    1.0        LOAD_MON NO3
*SCALE    1.0        LOAD_MON DON
*SCALE    1.0        LOAD_MON LON
*SCALE    1.0        LOAD_MON RON
```

```
*SCALE    1.075     LOAD_MON SA
*SCALE    1.0       LOAD_UNMON DOC
*SCALE    1.0       LOAD_UNMON DIA
*SCALE    1.0       LOAD_UNMON GRE
*SCALE    1.0       LOAD_UNMON LOC
*SCALE    1.0       LOAD_UNMON ROC
*SCALE    1.0       LOAD_UNMON SRP
*SCALE    1.0       LOAD_UNMON DOP
*SCALE    1.0       LOAD_UNMON LOP
*SCALE    1.0       LOAD_UNMON ROP
*SCALE    1.0       LOAD_UNMON NH4
*SCALE    1.0       LOAD_UNMON NO3
*SCALE    1.0       LOAD_UNMON DON
*SCALE    1.0       LOAD_UNMON LON
*SCALE    1.0       LOAD_UNMON RON
*SCALE    1.075     LOAD_UNMON SA
*SCALE    1.0       LOAD_ATM DOC
*SCALE    1.0       LOAD_ATM DIA
*SCALE    1.0       LOAD_ATM GRE
*SCALE    1.0       LOAD_ATM LOC
*SCALE    1.0       LOAD_ATM ROC
*SCALE    1.0       LOAD_ATM SRP
*SCALE    1.0       LOAD_ATM DOP
*SCALE    1.0       LOAD_ATM LOP
*SCALE    1.0       LOAD_ATM ROP
*SCALE    1.0       LOAD_ATM NH4
*SCALE    1.0       LOAD_ATM NO3
*SCALE    1.0       LOAD_ATM DON
*SCALE    1.0       LOAD_ATM LON
*SCALE    1.0       LOAD_ATM RON
*SCALE    1.0       LOAD_ATM SA
*SCALE    1.0       LOAD_SED DOC
*SCALE    1.0       LOAD_SED DIA
*SCALE    1.0       LOAD_SED GRE
*SCALE    1.0       LOAD_SED LOC
*SCALE    1.0       LOAD_SED ROC
*SCALE    26141.5   LOAD_SED SRP
*SCALE    1.0       LOAD_SED DOP
*SCALE    1.0       LOAD_SED LOP
*SCALE    1.0       LOAD_SED ROP
*SCALE    20093.0   LOAD_SED NH4
*SCALE    1.0       LOAD_SED NO3
*SCALE    1.0       LOAD_SED DON
*SCALE    1.0       LOAD_SED LON
*SCALE    1.0       LOAD_SED RON
*SCALE    30255.0   LOAD_SED SA
*SCALE    1.0       LOAD_SHORE_EROSION ROC

STOP
```

13.6 Example of a2cdf Calculation

A common case in which the a2cdf command is applied is the modification of initial conditions for the LM3 simulation. A sequence of Linux commands that might be used to create new initial conditions input CDF is shown below. The first two commands are used to copy the "public copy" of the LM3 initial conditions file ici.cdf into the user's local directory. Next, the a2cdf command is used to specify a constant value of initial concentration for the chloride state variable across all 42 segments of the level 2 segmentation. Alternatively, the a2cdf command can be used to accept sequential values from a prepared input file with 42 values (see the last command).

```
% cp /usr/model/dsets/LM3/R0105/ici/ici.cdf   my_ici_1.cdf
% cp /usr/model/dsets/LM3/R0105/ici/ici.cdf   my_ici_2.cdf
% repeat 41 echo “2.0”  | a2cdf  -R  -r  -n 1 CL my_ici_1
% a2cdf  -R  -r  -n 1 CL my_ici_1 < my_ascii_ici_values.dat
```

A more complex example involves the assignment of loads for the eutrophication state variables. The calculation is performed outside of LM3 in the Linux shell environment. Even for one of the eutrophication state variables, the manipulation involved to fill the CDF load zVariable eventually read by LM3 is not straightforward without knowledge of the Linux shell scripting. The complete Linux script responsible for this calculation is shown below and can be found in the directory /mnt/model/home/wam/ALPHAS_FILES/wam/CDF_DATA with the input files containing the load values themselves. Note the presence of the script /mnt/model/home/wam/ALPHAS_FILES/wam/CDF_DATA/all loads.csh which fills CDF load files for all of the eutrophication states. The technique can be readily modified and improved for future state variables to be represented by the LM3 simulation.

```
#!/bin/csh
#
# Fill tributary and atmospheric loads into
# CDF file used by model calculation
#

set Ldfile = “ld_srp”

set Tribfile1   =   “Menominee.csv”
set Tribfile2   =   “Fox.csv”
set Tribfile3   =   “Sheboygan.csv”
set Tribfile4   =   “Milwaukee.csv”
set Tribfile5   =   “Calumet.csv”
set Tribfile6   =   “St_Joseph.csv”
set Tribfile7   =   “Kalamazoo.csv”
set Tribfile8   =   “Grand.csv”
set Tribfile9   =   “Muskegon.csv”
set Tribfile10 =   “Pere.csv”
set Tribfile11 =   “Manistique.csv”

#
# Input atmospheric loads in ASCII format
#
set Atmfile1    =   “atm94_1.csv”
set Atmfile2    =   “atm94_2.csv”
```

```
set Atmfile3   =  “atm94_3.csv”
set Atmfile4   =  “atm94_4.csv”
set Atmfile5   =  “atm94_5.csv”
set Atmfile6   =  “atm94_6.csv”
set Atmfile7   =  “atm94_7.csv”
set Atmfile8   =  “atm94_8.csv”
set Atmfile9   =  “atm94_9.csv”
set Atmfile10  =  “atm94_10.csv”
set Atmfile11  =  “atm94_11.csv”
set Atmfile12  =  “atm94_12.csv”
set Atmfile13  =  “atm95_13.csv”
set Atmfile14  =  “atm95_14.csv”
set Atmfile15  =  “atm95_15.csv”
set Atmfile16  =  “atm95_16.csv”
set Atmfile17  =  “atm95_17.csv”
set Atmfile18  =  “atm95_18.csv”
set Atmfile19  =  “atm95_19.csv”
set Atmfile20  =  “atm95_20.csv”
set Atmfile21  =  “atm95_21.csv”
set Atmfile22  =  “atm95_22.csv”
set Atmfile23  =  “atm95_23.csv”
set Atmfile24  =  “atm95_24.csv”

#
# Prepare an empty load file
#
sed  -e ‘s/xxx/srp/’ ld_xxx.skt > $Ldfile.skt
touch $Ldfile.cdf ; rm  -f $Ldfile.cdf

#
# Trib time values
#
if ($1 == “tributimes”  || $1 ==- “all”) then
tail +4 $Tribfile1   |  awk -F, ‘ {print $1}’  |  sed “s/\// /g”  |  a2cdf  -Z  -j  -n 1 TM_01   $Ldfile
tail +4 $Tribfile2   |  awk -F, ‘ {print $1}’  |  sed “s/\// /g”  |  a2cdf  -Z  -j  -n 1 TM_01   $Ldfile
tail +4 $Tribfile3   |  awk -F, ‘ {print $1}’  |  sed “s/\// /g”  |  a2cdf  -Z  -j  -n 1 TM_01   $Ldfile
tail +4 $Tribfile4   |  awk -F, ‘ {print $1}’  |  sed “s/\// /g”  |  a2cdf  -Z  -j  -n 1 TM_01   $Ldfile
tail +4 $Tribfile5   |  awk -F, ‘ {print $1}’  |  sed “s/\// /g”  |  a2cdf  -Z  -j  -n 1 TM_01   $Ldfile
tail +4 $Tribfile6   |  awk -F, ‘ {print $1}’  |  sed “s/\// /g”  |  a2cdf  -Z  -j  -n 1 TM_01   $Ldfile
tail +4 $Tribfile7   |  awk -F, ‘ {print $1}’  |  sed “s/\// /g”  |  a2cdf  -Z  -j  -n 1 TM_01   $Ldfile
tail +4 $Tribfile8   |  awk -F, ‘ {print $1}’  |  sed “s/\// /g”  |  a2cdf  -Z  -j  -n 1 TM_01   $Ldfile
tail +4 $Tribfile9   |  awk -F, ‘ {print $1}’  |  sed “s/\// /g”  |  a2cdf  -Z  -j  -n 1 TM_01   $Ldfile
tail +4 $Tribfile10  |  awk -F, ‘ {print $1}’  |  sed “s/\// /g”  |  a2cdf  -Z  -j  -n 1 TM_01   $Ldfile
tail +4 $Tribfile11  |  awk -F, ‘ {print $1}’  |  sed “s/\// /g”  |  a2cdf  -Z  -j  -n 1 TM_01   $Ldfile
endif

#
# Trib load values
#
if ($1 == “tribloads”  || $1 == “all”) then
tail +4 $Tribfile1   |  awk -F, ‘ {print $7/86400}’  |  float  |  a2cdf  -Z  -j  -n 1 LD_01   $Ldfile
tail +4 $Tribfile2   |  awk -F, ‘ {print $7/86400}’  |  float  |  a2cdf  -Z  -j  -n 1 LD_01   $Ldfile
tail +4 $Tribfile3   |  awk -F, ‘ {print $7/86400}’  |  float  |  a2cdf  -Z  -j  -n 1 LD_01   $Ldfile
tail +4 $Tribfile4   |  awk -F, ‘ {print $7/86400}’  |  float  |  a2cdf  -Z  -j  -n 1 LD_01   $Ldfile
```

```
tail +4 $Tribfile5    |   awk -F, ‘ {print $7/86400}’   |   float   |   a2cdf  -Z  -j  -n 1 LD_01   $Ldfile
tail +4 $Tribfile6    |   awk -F, ‘ {print $7/86400}’   |   float   |   a2cdf  -Z  -j  -n 1 LD_01   $Ldfile
tail +4 $Tribfile7    |   awk -F, ‘ {print $7/86400}’   |   float   |   a2cdf  -Z  -j  -n 1 LD_01   $Ldfile
tail +4 $Tribfile8    |   awk -F, ‘ {print $7/86400}’   |   float   |   a2cdf  -Z  -j  -n 1 LD_01   $Ldfile
tail +4 $Tribfile9    |   awk -F, ‘ {print $7/86400}’   |   float   |   a2cdf  -Z  -j  -n 1 LD_01   $Ldfile
tail +4 $Tribfile10  |   awk -F, ‘ {print $7/86400}’   |   float   |   a2cdf  -Z  -j  -n 1 LD_01   $Ldfile
tail +4 $Tribfile11  |   awk -F, ‘ {print $7/86400}’   |   float   |   a2cdf  -Z  -j  -n 1 LD_01   $Ldfile
endif

#
# Atmospheric load time values
#
if ($1 == “atmtimes”  || $1 == “all”) then
touch
rm  -f tmp.dat
echo   “1    1   94"   >>   tmp.dat
echo   “2    1   94"   >>   tmp.dat
echo   “3    1   94"   >>   tmp.dat
echo   “4    1   94"   >>   tmp.dat
echo   “5    1   94"   >>   tmp.dat
echo   “6    1   94"   >>   tmp.dat
echo   “7    1   94"   >>   tmp.dat
echo   “8    1   94"   >>   tmp.dat
echo   “9    1   94"   >>   tmp.dat
echo   “10  1   94"   >>   tmp.dat
echo   “11  1   94"   >>   tmp.dat
echo   “12  1   94"   >>   tmp.dat
echo   “1    1   95"   >>   tmp.dat
echo   “2    1   95"   >>   tmp.dat
echo   “3    1   95"   >>   tmp.dat
echo   “4    1   95"   >>   tmp.dat
echo   “5    1   95"   >>   tmp.dat
echo   “6    1   95"   >>   tmp.dat
echo   “7    1   95"   >>   tmp.dat
echo   “8    1   95"   >>   tmp.dat
echo   “9    1   95"   >>   tmp.dat
echo   “10  1   95"   >>   tmp.dat
echo   “11  1   95"   >>   tmp.dat
echo   “12  1   95"   >>   tmp.dat
a2cdf   -Z  -j  -n 1 TM_39 $Ldfile < tmp.dat
endif
#
# Atmospheric load values
#
if ($1 == “atmloads”  || $1 == “all”) then

touch tmp.dat
rm  -f tmp.dat

tail   +4 $Atmfile1    |   awk  -F, ‘{print #3}’   >>   tmp.dat
tail   +4 $Atmfile2    |   awk  -F, ‘{print #3}’   >>   tmp.dat
tail   +4 $Atmfile3    |   awk  -F, ‘{print #3}’   >>   tmp.dat
tail   +4 $Atmfile4    |   awk  -F, ‘{print #3}’   >>   tmp.dat
tail   +4 $Atmfile5    |   awk  -F, ‘{print #3}’   >>   tmp.dat
```

```
tail  +4 $Atmfile6   |  awk  -F, '{print #3}'  >>  tmp.dat
tail  +4 $Atmfile7   |  awk  -F, '{print #3}'  >>  tmp.dat
tail  +4 $Atmfile8   |  awk  -F, '{print #3}'  >>  tmp.dat
tail  +4 $Atmfile9   |  awk  -F, '{print #3}'  >>  tmp.dat
tail  +4 $Atmfile10  |  awk  -F, '{print #3}'  >>  tmp.dat
tail  +4 $Atmfile11  |  awk  -F, '{print #3}'  >>  tmp.dat
tail  +4 $Atmfile12  |  awk  -F, '{print #3}'  >>  tmp.dat
tail  +4 $Atmfile13  |  awk  -F, '{print #3}'  >>  tmp.dat
tail  +4 $Atmfile14  |  awk  -F, '{print #3}'  >>  tmp.dat
tail  +4 $Atmfile15  |  awk  -F, '{print #3}'  >>  tmp.dat
tail  +4 $Atmfile16  |  awk  -F, '{print #3}'  >>  tmp.dat
tail  +4 $Atmfile17  |  awk  -F, '{print #3}'  >>  tmp.dat
tail  +4 $Atmfile18  |  awk  -F, '{print #3}'  >>  tmp.dat
tail  +4 $Atmfile19  |  awk  -F, '{print #3}'  >>  tmp.dat
tail  +4 $Atmfile20  |  awk  -F, '{print #3}'  >>  tmp.dat
tail  +4 $Atmfile21  |  awk  -F, '{print #3}'  >>  tmp.dat
tail  +4 $Atmfile22  |  awk  -F, '{print #3}'  >>  tmp.dat
tail  +4 $Atmfile23  |  awk  -F, '{print #3}'  >>  tmp.dat
tail  +4 $Atmfile24  |  awk  -F, '{print #3}'  >>  tmp.dat

a2cdf  -Z  -r  -n 2318 LD_39 $Ldfile < tmp.dat

endif

exit
```

13.7 Example of x2cdf Calculation

The x2cdf program is currently used for filling shortwave radiation, heat fluxes, and wave information in the hydrodynamic input CDF. An example is provided here demonstrating the method that might be used to fill missing wave period information in a hydrodynamic calculation output CDF after that calculation is complete. The Linux commands for filling the missing CDF wave period zVariable are shown below:

```
find /home/llrs_model/jobs/<job number>  -name 'hydro82.*'  -print  - exec ln  -s {} \;
find /home/llrs_model/dat/WAVES/R0102/973.llrssrv2.grl.epa.gov  -name '*.wvp'  -print  -exec  ln  -s {} \;
x2cdf-1.1 <  firead.94.wvp.1h.inp
```

The corresponding FIREAD input file, which can be found in the directory /mnt/model/home/wam/ALPHAS_FILES/wam/POM/filling, uses the parameter keys described in Table 12.1. The input XDR formatted wave period files are specified. The name of the output CDF to be filled is given along with the name of the zVariable to be filled. The BOUNDS key is used to inform the application that the data values are being applied to the center of each surface cell of the Lake Michigan high-resolution geometry. The number of input records, corresponding to hourly values over 60 days, is given and six of these are averaged to define each wave period value in the output CDF. The first input record values of each XDR file are read (not skipped) and the bathymetry grid name for the Lake Michigan study is provided. See the scripts /mnt/model/home/wam/ALPHAS_FILES/wam/POM/filling/x2cdf82 and /mnt/model /home/wam/ALPHAS_FILES/wam/POM/filling/x2cdf94 that have been used to fill light, heat flux, and wave information for the hydrodynamic calibration and study periods.

Comment: Values for sensible heat flux, latent heat flux, and total heat flux have been taken from calculations that were delivered to Large Lakes Research Station (LLRS) as part of the Great Lakes Environmental Research Laboratory (GLERL) final report on the Lake Michigan hydrodynamics study (Schwab and Beletsky, 1998). Values for these parameters are an output of POM and, in principle, should be re-calculated at LLRS. This has not been done and the heat flux values being employed are not consistent with the hydrodynamics employed. This should not be a problem for two reasons. First, only minor changes have been made to the GLERL hydrodynamic model since it was first received as a part of that final report; any differences between the heat fluxes originally calculated and those estimated at LLRS are hypothesized to be small and negligible. Second, the heat flux information is not currently used by the water quality model.

```
LIST

C
C Example X2CDF input file used for filling wave periods
C in the 1994 hydrodynamics CDF.
C

C The input file names.
FNAME 1=m94001.wvd   2=m94061.wvd   3=m94121.wvd   4=m94181.wvd   5=94241.wvd
          6=m94301.wvd   7=m94361.wvd   8=m95056.wvd   9=m95116.wvd   10=m95176.wvd
         11=m95236.wvd 12=9596.wvd

C CDF output file name
CDFFILE hydro94

C CDF zVariable name
ZVARAME WVP

C Boundary type
BOUNDS 1

C Number of records
NREC 1440
C Number of records to average
NRAV 1

C Skip first record flags.
SKIP0 1=12*0

C GLERL and LLRS/SEDZL flavor bathymetry files.
GBATH mich5.dat

END
```

CHAPTER 14

REFERENCES

Acero, J.L.; K. Stemmler, and U.V. Gunten. 2000. Degradation Kinetics of Atrazine and Its Degradation Products With Ozone and OH Radicals: A Predictive Tool for Drinking Water Treatment. Environ. Sci. Technol., 34(4):591-597.

Ambrose, R.B., T.A. Wool, J.P. Connolly, and R.W. Shanz. 1988. WASP4, A Hydrodynamic and Water Quality Model Model Theory, User's Manual, and Programmer's Guide. U.S. Environmental Protection Agency, Office of Research and Development, Environmental Research Laboratory, Athens, Georgia. EPA/600/3-87/039, 297 pp.

Arrhenius, S. 1889. On the Reaction Velocity of the Inversion of Cane Sugar by Acids. Zeitschrift fur Physikalische Chemie, 4:226-232.

Balmer, M.E. and B. Sulzberger. 1999. Atrazine Degradation in Irradiated Iron/Oxalate Systems: Effects of pH and Oxalate. Environ. Sci. Technol., 33(14):2418-2424.

Bamford, H.A., D.L. Poster, R.E. Huie, and J.E. Baker. 2002. Using Extrathermodynamic Relationships to Model the Temperature Dependence of Henry's Law Constants of 209 PCB Congeners. Environ. Sci. Technol., 36(20):4395-4402.

Bartone, C.R. and C.L. Schelske. 1982. Lake-wide Seasonal Changes in Limnological Conditions in Lake Michigan in 1976. J. Great Lakes Res., 8(3):413-427.

Bowie, G.L., W.B. Mills, D.B. Porcella, C.L. Campbell, J.R. Pagenkopf, G.L. Rupp, K.M. Johnson, P.W.H. Chan, S.A. Gherini, and C.E. Chamberlin. 1985. Rates, Constants and Kinetic Formulations in Surface Water Quality Modeling. U.S. Environmental Protection Agency, Office of Research and Development, Environmental Research Laboratory, Athens, Georgia. EPA/600/3-85/040, 455 pp.

Brooks, A.S. and B.G. Torke. 1977. Vertical and Seasonal Distribution of Chlorophyll *a* in Lake Michigan. J. Fish. Res. Board Canada, 34(12):2280-2287.

Brown, S., M. Folk, G. Goucher, and R. Rew. 1993. Software for Portable Scientific Data Management. Comp. Physics, 7(3):304-308.

Burkhard, N. and J. Guth. 1976. Photodegradation of Atrazine, Atraton, and Ametryne in Aqueous Solution With Acetone as a Photosensitizer. J. Pest. Sci., 7(1):65-71.

Carnahan, B., H.A. Luther, and J.O. Wilkes. 1969. Applied Numerical Methods. John Wiley and Sons, Incorporated, New York, New York. 604 pp.

Cerco, C.F. and T. Cole. 1993. Three-Dimensional Eutrophication Model of Chesapeake Bay. J. Environ. Engin., 119(6):1006-1025.

Cerco, C. and T. Cole. 1994. Three-Dimensional Eutrophication Model of Chesapeake Bay, Volume I: Main Report. U.S. Army Corps of Engineers, U.S. Army Engineer Waterways Experiment Station, Baltimore, Maryland. Technical Report EL-94-4, 658 pp.

Cerco, C. and T. Cole. 1995. User's Guide to the CE-QUAL-ICM Three-Dimensional Eutrophication Model. U.S. Army Corps of Engineers, U.S. Army Engineer Waterways Experiment Station, Vicksburg, Mississippi. Technical Report EL-95-15, 2,420 pp.

Chapman, R.S., T.M. Cole, and T.K. Gerald. 1997. Development of a Hydrodynamic/Water Quality (POM-IPX) Linkage for the Lake Michigan Mass Balance Project. Final Report. U.S. Environmental Protection Agency, Office of Research and Development, National Health and Environmental Effects Research Laboratory, Mid-Continent Ecology Division-Duluth, Large Lakes Research Station, Grosse Ile, Michigan. 63 pp.

Christoffersen, J.B. and I.G. Jonsson. 1985. Bed Friction and Dissipation in a Combined Current and Wave Motion. Ocean Engin., 12(5):387-423.

DePinto, J.V., R. Raghunathan, P. Sierzenga, X. Zhang, V.J. Bierman, Jr., P.W. Rodgers, and T.C. Young. 1993. Recalibration of GBTOX: An Integrated Exposure Model for Toxic Chemicals in Green Bay, Lake Michigan. Final Report. U.S. Environmental Protection Agency, Office of Research and Development, Environmental Research Laboratory-Duluth, Large Lakes Research Station, Grosse Ile, Michigan. 132 pp.

Di Toro, D.M., D.J. O'Connor, and R.V. Thomann. 1971. A Dynamic Model of the Phytoplankton Population in the Sacramento-San Joaquin Delta. In: R.F. Gould (Ed.), Non-Equilibrium Systems in Natural Water Chemistry, pp. 131-180. American Chemical Society, Washington, D.C.

Di Toro, D.M. and J.P. Connolly. 1980. Mathematical Models of Water Quality in Large Lakes, Part 2: Lake Erie. U.S. Environmental Protection Agency, Office of Research and Development, Environmental Research Laboratory-Duluth, Large Lakes Research Station, Grosse Ile, Michigan. EPA/600/3-80/065, 97 pp.

Di Toro, D.M. and W.F. Matystik, Jr. 1980. Mathematical Models of Water Quality in Large Lakes, Part 1: Lake Huron and Saginaw Bay. U.S. Environmental Protection Agency, Office of Research and Development, Environmental Research Laboratory-Duluth, Large Lakes Research Station, Grosse Ile, Michigan. EPA/600/3-80/056, 180 pp.

Endicott, D.D., W.L. Richardson, and R. Rossmann. 2006. PCBs Modeling Overview. In: R. Rossmann (Ed.), Results of the Lake Michigan Mass Balance Project: Polychlorinated Biphenyl Modeling Report, Part 1, Chapter 2, pp. 16-25. U.S. Environmental Protection Agency, Office of Research and Development, National Health and Environmental Effects Research Laboratory, Mid-Continent Ecology Division-Duluth, Large Lakes Research Station, Grosse Ile, Michigan. EPA/600/R-04/167, 579 pp.

Fahnenstiel, G.L. and D. Scavia. 1987a. Dynamics of Lake Michigan Photoplankton: Primary Production and Growth. Canadian J. Fish. Aquat. Sci., 44(3):499-508.

Fahnenstiel, G.L. and D. Scavia. 1987b. Dynamics of Lake Michigan Photoplankton: Recent Changes in Surface and Deep Communities. Canadian J. Fish. Aquat. Sci., 44(3)509-514.

Gailani, J.Z., C.K. Ziegler, and W.J. Lick. 1991. Transport of Suspended Solids in the Lower Fox River. J. Great Lakes Res., 17(4): 479-494.

Grant, W.D. and O.S. Madsen. 1979. Combined Wave and Current Interaction with a Rough Bottom. J. Geophy. Res., 84(C4):1797-1808.

Hayes, T., K. Haston, M. Tsui, A. Hoang, C. Haeffele, and A. Vonk. 2003. Atrazine-Induced Hermaphroditism at 0.1 ppb in American Leopard Frogs (*Rana pipiens*): Laboratory and Field Evidence. Environ. Health Persp., 111(4):568-573.

Konstantinou, I.K., A.K. Zarkadis, and T.A. Albanis. 2001. Photodegradation of Selected Herbicides in Various Natural Waters and Soils Under Environmental Conditions. J. Environ. Quality, 30(1):121-130.

Kremer, J. and S. Nixon. 1974. A Coastal Marine Ecosystem: Simulation and Analysis. Springer Verlag, New York, New York. 217 pp.

Leonard, B. 1979. A Stable and Accurate Convective Modelling Procedure Based on Quadratic Upstream Interpolation. Comp. Methods Appl. Mechan. Engin., 19(1):59-98.

Lick, W.J., C.K. Ziegler, and C.-H. Tsai. 1987. Resuspension, Deposition, and Transport of Fine-Grained Sediments in Rivers and Near-Shore Areas. Report to the U.S. Environmental Protection Agency, Office of Research and Development, Environmental Research Laboratory-Duluth, Large Lakes Research Station, Grosse Ile, Michigan. 94 pp.

Lick, W., L. Jin, and J. Gailani. 2004. Initiation of Movement of Quartz Particles. J. Hydraul. Engin., 130(8):755-761.

Makarewicz, J.C., P. Bertram, and T.W. Lewis. 1998. Changes in Phytoplankton Size-Class Abundance and Species Composition Coinciding With Changes in Water Chemistry and Zooplankton Community Structure of Lake Michigan, 1983 to 1992. J. Great Lakes Res., 24(3):637-657.

McNeil, J. and W. Lick. 2002. Erosion Rates and Bulk Properties of Sediments in Lake Michigan. University of California, Department of Mechanical and Environmental Engineering, Santa Barbara, California. 110 pp.

Miller, S.M. 1999. Spatial and Temporal Variability of Organic and Nutrient Compounds in Atmospheric Media Collected During the Lake Michigan Mass Balance Study. M.S. Thesis, Department of Civil, Structural, and Environmental Engineering, State University of New York, Buffalo, New York.

Monod, J. 1949. The Growth of Bacterial Cultures. Ann. Rev. Microbiol., 3:371-394.

National Aeronautics and Space Administration. 1996a. CDF User's Guide. National Aeronautics and Space Administration, National Space Science Data Center, Space Physics Data Facility, Goddard Space Flight Center, Greenbelt, Maryland. 205 pp.

National Aeronautics and Space Administration. 1996b. CDF FORTRAN Reference Manual. National Aeronautics and Space Administration, National Space Science Data Center, Space Physics Data Facility, Goddard Space Flight Center, Greenbelt, Maryland. 177 pp.

National Aeronautics and Space Administration. 1996c. CDF C Reference Manual. National Aeronautics and Space Administration, National Space Science Data Center, Space Physics Data Facility, Goddard Space Flight Center, Greenbelt, Maryland. 165 pp.

O'Connor, D.J. 1983a. The Significance of Gas Exchange in Water Quality Assessment and Planning. In: W. Brutsaert and G.H. Jirka (Eds.), Gas Transfer at Water Surfaces, Part 7, pp. 559-576. D. Reidel Publishing Company, Boston, Massachusetts. 656 pp.

O'Connor, D.J. 1983b. Wind Effects on Gas-Liquid Transfer Coefficients. J. Environ. Engin., 109(9):731-752.

Odum, E. 1971. Fundamentals of Ecology, 3rd Edition. W.B. Saunders Company, Philadelphia, Pennsylvania. 574 pp.

Parker, S.P. 1993. McGraw-Hill Encyclopedia of Physics, 2nd Edition. McGraw-Hill Companies, New York, New York. 1,624 pp.

Pauer, J.J., K.W. Taunt, and W. Melendez. 2006. LM3-Eutro. In: R. Rossmann (Ed.), Results of the Lake Michigan Mass Balance Project: Polychlorinated Biphenyl Modeling Report, Part 2, pp. 120-182. U.S. Environmental Protection Agency, Office of Research and Development, National Health and Environmental Effects Research Laboratory, Mid-Continent Ecology Division-Duluth, Large Lakes Research Station, Grosse Ile, Michigan. EPA/600/R-04/167, 579 pp.

Pratap, K. and A.T. Lemley. 1998. Fenton Electrochemical Treatment of Aqueous Atrazine and Metolachlor. J. Agricult. Food Chem., 46(8):3285-3291.

Ralebitso, T.K., E. Senior, and W.H. van Verseveld. 2002. Microbial Aspects of Atrazine Degradation in Natural Environments. Biodegradation, 13(1):11-19.

Rebelo, S.L.H., A. Melo, R. Coimbra, M.E. Azenha, M.M. Pereira, H.D. Burrows, and M. Sarakha. 2007. Photodegradation of Atrazine and Ametryn With Visible Light Using Water Soluble Porphyrins as Sensitizers. Environ. Chem. Letters, 5(1):29-33.

Ribaudo, M.O. and A. Bouzaher. 1994. Atrazine: Environmental Characteristics and Economics of Management. U.S. Department of Agriculture, Economic Research Service, Washington, D.C. Agricultural Economic Report Number 699, 22 pp.

Robbins, J.A., N.R. Morehead, R.W. Rood, D.N. Edgington, and S. Meyer. 1999. Accumulation and Near-Surface Mixing of Sediments in Lake Michigan as Determined for the Lake Michigan Mass Balance Program, Volumes 1 and 2. Project Report. U.S. Environmental Protection Agency, Office of Research and Development, National Health and Environmental Effects Research Laboratory, Mid-Continent Ecology Division-Duluth, Large Lakes Research Station, Grosse Ile, Michigan. 503 pp.

Roberts, J., R. Jepsen, and W. Lick. 1998. Effects of Particle Size and Bulk Density on the Erosion of Quartz Particles. J. Hydraul. Engin., 124(12):1261-1267.

Rockwell, D.C., D.K. Salisbury, and B.W. Lesht. 1989. Water Quality in the Middle Great Lakes: Results of the 1985 U.S. EPA Survey of Lakes Erie, Huron and Michigan. U.S. Environmental Protection Agency, Great Lakes National Program Office, Chicago, Illinois. EPA/905/6-89/001, 272 pp.

Rodgers, P.W. and D. Salisbury. 1981a. Water Quality Modeling of Lake Michigan and Consideration of the Anomalous Ice Cover of 1976-1977. J. Great Lakes Res., 7(4):467-480.

Rodgers, P.W. and D. Salisbury. 1981b. Modeling of Water Quality in Lake Michigan and the Effects of Anomalous Ice Cover of 1976-1977. Great Lakes Basin Commission, Great Lakes Environmental Planning Study, Great Lakes Basin Commission, Ann Arbor, Michigan. Contribution Number 44, 53 pp.

Rossmann, R. (Editor). 2006. Results of the Lake Michigan Mass Balance Project: Polychlorinated Biphenyl Modeling Report. U.S. Environmental Protection Agency, Office of Research and Development, National Health and Environmental Effects Research Laboratory, Mid-Continent Ecology Division-Duluth, Large Lakes Research Station, Grosse Ile, Michigan. EPA/600/R-04/167, 579 pp.

Rousar, D.C. 1973. Seasonal and Spatial Changes in Primary Production and Nutrients in Lake Michigan. Water Air Soil Pollut., 2(4):497-514.

Rygwelski, K.R. (Ed.). 2008. Results of the Lake Michigan Mass Balance Project: Atrazine Modeling Report. U.S. Environmental Protection Agency, Office of Research and Development, National Health and Environmental Effects Research Laboratory, Mid-Continent Ecology Division, Large Lakes Research Station, Grosse Ile, Michigan. EPA/600/R-08/111, 140 pp.

Scavia, D. and G.L. Fahnenstiel. 1987. Dynamics of Lake Michigan Phytoplankton Mechanisms Controlling Epilimnetic Communities. J. Great Lakes Res., 13(2):103-120.

Scavia, D., G.A. Lang, and J.F. Kitchell. 1988. Dynamics of Lake Michigan Plankton: A Model Evaluation of Nutrient Loading, Competition, and Predation. Canadian J. Fish. Aquat. Sci., 45(1):165-177.

Schabacker, D. 1991. Summary Report: Aqueous Photolysis of Carbon-14 Atrazine Under Natural and Artificial Light. Unpublished Study Prepared by Agrisearch, Incorporated, Frederick, Maryland. Submitted by Ciba-Geigy Corporation, Greensboro, North Carolina. Laboratory Project Number 12112A and 12112B, 185 pp.

Schelske, C.L. and E.F. Stoermer. 1971. Eutrophication, Silica Depletion, and Predicted Changes in Algal Quality in Lake Michigan. Science, 173(3995):423-424.

Schelske, C.L. and E.F. Stoermer. 1972. Phosphorus, Silica and Eutrophication in Lake Michigan. In: G.E. Likens (Ed.), Nutrients and Eutrophication, pp. 157-171. American Society of Limnology and Oceanography, Lawrence, Kansas.

Schelske, C.F., E.F. Stoermer, G.L. Fahnenstiel, and M. Haibach. 1986. Phosphorus Enrichment, Silica Utilization and Biogeochemical Silica Depletion in the Great Lakes. Canadian J. Fish. Aquat. Sci., 43(2):407-415.

Scholtz, M.T., B.J. Van Heyst, and A. Ivanoff. 1999. Documentation for the Gridded Hourly Atrazine Emissions Data for the Lake Michigan Mass Balance Study. U.S. Environmental Protection Agency, Great Lakes National Program Office, Chicago, Illinois. EPA/600/R-99/067, 61 pp.

Schwab, D.J. and D. Beletsky. 1998. Lake Michigan Mass Balance Study: Hydrodynamic Modeling Project. U.S. Department of Commerce, National Oceanic and Atmospheric Administration, Great Lakes Environmental Research Laboratory, Ann Arbor, Michigan. ERL GLERL-108, 53 pp.

Schwarzenbach, R.P., P.M. Gschwend, and D.M. Imboden. 1993. Environmental Organic Chemistry. John Wiley and Sons, Incorporated, New York, New York. 681 pp.

Settles, M. 1996. The FIREAD Primer: Format Independent Reading of Program Input Parameters. Internal Technical Note. U.S. Environmental Protection Agency, Office of Research and Development, National Health and Environmental Effects Research Laboratory, Mid-Continent Ecology Division-Duluth, Large Lakes Research Station, Grosse Ile, Michigan. 20 pp.

Settles, M. 1998. UT - Version 1.3.1. Internal Technical Note. U.S. Environmental Protection Agency, Office of Research and Development, National Health and Environmental Effects Research Laboratory, Mid-Continent Ecology Division-Duluth, Large Lakes Research Station, Grosse Ile, Michigan. 5 pp.

Steele, J.H. 1962. Environmental Control of Photosynthesis in the Sea. Limnol. Oceanogr., 7(2):137-150.

Stoermer, E.F. 1993. Evaluating Diatom Succession: Some Peculiarities of the Great Lakes Case. J. Paleolimnol., 8(1):71-83.

Tauber, A. and C. von Sonntag. 2000. Products and Kinetics of the OH-Radical-Induced Dealkylation of Atrazine. Acta Hydrochim. Hydrobiol., 28(1):15-23.

Thibodeaux, L.J. 1979. Chemodynamics: Environmental Movement of Chemicals in Air, Water, and Soil. John Wiley and Sons, Incorporated, New York, New York. 501 pp.

Thomann, R.V. and D.M. Di Toro. 1975. Mathematical Modeling of Phytoplankton in Lake Ontario, Part 1 - Model Development and Verification. U.S. Environmental Protection Agency, Office of Research and Development, Environmental Research Laboratory-Corvallis, Large Lakes Research Station, Grosse Ile, Michigan. EPA/600/3-75/005, 178 pp.

Thomann, R.V. and J.A. Mueller. 1987. Principles of Water Quality Modeling and Control. Harper and Row Publishers, New York, New York. 644 pp.

Treinish, L.A. and M.L. Gough. 1987. A Software Package for the Data Independent Storage of Multi-Dimensional Data. EROS Trans., 68(28):633-635.

Treinish, L.A. and G.W. Goucher. 1988. A Data Abstraction for the Source Independent Storage and Manipulation of Data. National Aeronautics and Space Administration, National Space Science Center, Goddard Space Flight Center, Greenbelt, Maryland.

Treinish, L.A. 1991. Data Structure and Access Software for Scientific Visualization. A Report on a Workshop at Siggraph '90. Comp. Graphics, 25(2):104-118.

U.S. Environmental Protection Agency. 2001. Ambient Aquatic Life Water Quality for Atrazine. U.S. Environmental Protection Agency, Office of Water, Washington, D.C. EPA/822/D-01/002, 230 pp.

Velleux, M., S. Westenbroek, J. Ruppel, M. Settles, and D. Endicott. 2001. A User's Guide to IPX, The In-Place Pollutant Export Water Quality Modeling Framework. U.S. Environmental Protection Agency, Office of Research and Development, National Health and Environmental Effects Research Laboratory, Mid-Continent Ecology Division-Duluth, Large Lakes Research Station, Grosse Ile, Michigan. 179 pp.

Wackett, L.P., M.J. Sadowsky, B. Martinez, and N. Shapir. 2002. Biodegradation of Atrazine and Related *s*-Triazine Compounds: From Enzymes to Field Studies. Appl. Microbiol. Biotechnol., 58(1):39-45.

Wanninkhof, R.J., J.R. Ledwell, and J. Crusius. 1991. Gas Transfer Velocities on Lakes Measured with Sulfur Hexafluoride. In: S.C. Wilhelm and J.S. Culliver (Eds.), Air-Water Mass Transfer, pp. 441-458. American Society of Civil Engineers, New York, New York.

Wanninkhof, R.J. 1992. Relationship Between Gas Exchange and Wind Speed Over the Ocean. J. Geophys. Res., 97:7373-7381.

Watson, D.F. 1994. An Implementation of Natural Neighbor Interpolation. Dave Watson Publisher, Claremont, Australia. 170 pp.

Weast, R.C. 1989. Handbook of Chemistry and Physics: A Ready-Reference Book of Chemical and Physical Data, 70th Ed. The Chemical Rubber Publishing Company, Cleveland, Ohio. 2,524 pp.

Whitman, W.G. 1923. A Preliminary Experimental Confirmation of the Two-Film Theory of Gas Absorption. Chem. Metall. Eng., 29:146-148.

Xia, X. 2006. Sample Data Interpolation for the LMMBP. In: R. Rossmann (Ed.), Results of the Lake Michigan Mass Balance Project: Polychlorinated Biphenyls Modeling Report, Part 4, Appendix 4.4.1, pp. 286-290. U.S. Environmental Protection Agency, Office of Research and Development, National Health and Environmental Effects Research Laboratory, Mid-Continent Ecology Division-Duluth, Large Lakes and Rivers Forecasting Research Branch, Large Lakes Research Station, Grosse Ile, Michigan. EPA/600/R-04/167, 579 pp.

Zeng, Y., C.L. Sweeney, S. Stephens, and P. Kotharu. 2004. Atrazine Pathway Map. Wackett LP. Biodegradation Database at http://umbbd.ahc.umn.edu/atr/atr_map.html.

Zepp, R.G., J. Hoigne, and H. Bader. 1987. Nitrate-Induced Photooxidation of Trace Organic Chemicals in Water. Environ. Sci. Technol., 21(5):443-450.

Zhang, X. 2006. LM2-Toxic. In: R. Rossmann (Ed.), Results of the Lake Michigan Mass Balance Project: Polychlorinated Biphenyls Modeling Report, Part 4, Chapter 4.4.3, pp. 271. U.S. Environmental Protection Agency, Office of Research and Development, National Health and Environmental Effects Research Laboratory, Mid-Continent Ecology Division-Duluth, Large Lakes and Rivers Forecasting Research Branch, Large Lakes Research Station, Grosse Ile, Michigan. EPA/600/R-04/167, 579 pp.

Ziegler, C.K. and W.J. Lick. 1986. A Numerical Model of the Resuspension, Deposition, and Transport of Fine-Grained Sediments in Shallow Water. Report to the U.S. Environmental Protection Agency, Office of Research and Development, Environmental Research Laboratory-Duluth, Large Lakes Research Station, Grosse Ile, Michigan. 179 pp.

APPENDIX A

DEVELOPMENT OF A HYDRODYNAMIC/WATER QUALITY (POM/IPX) LINKAGE FOR THE LAKE MICHIGAN MASS BALANCE PROJECT

Final Report

by

Raymond S. Chapman, Ph.D.
Ray Chapman and Associates
1725 Mac Arthur Place
Vicksburg, Mississippi 39180

Thomas M. Cole
Environmental Laboratory
U.S. Army Corps of Engineers
Waterways Experiment Station
3909 Halls Ferry Road
Vicksburg, Mississippi 39180

Terry K. Gerald
AScI, Incorporated
U.S. Army Corps of Engineers
Waterways Experiment Station
3909 Halls Ferry Road
Vicksburg, Mississippi 39180

Prepared for

U.S. Environmental Protection Agency
Office of Research and Development
National Health and Environmental Effects Research Laboratory
Mid-Continent Ecology Division-Duluth
Large Lakes Research Station
9311 Groh Road
Grosse Ile, Michigan 48138

December 11, 1997

Contents

Introduction 221
CE-QUAL-ICM Transport Concept 221
Non-Uniform Grid ULTIMATE QUICKEST Formulation 222
Vertical Implicit Transport 226
POM-IPX Linkage 228
MAPPER Routines 228
    ICM_COOR.OUT 229
    GEO.INP 229
    MAP.INP 229
    FILE95.INP 229
POM Linkage Processor 229
    Task 1 230
    Task 2 231
    Task 3 231
    Task 4 231
    IXMT Modifications 232
    IPX Linkage and Input Modifications 232
    IPXMT Transport Description 232
Guide to the Use of the Linked POMGL-IPXMT 233
Linkage Quality Assurance Testing 239
    Simplified Grid Tests 239
    Prototype Grid Tests 239
General Analytic Testing 239
References 240

Attachment 1 IPXMT Transport Input, MAP File, and GEO File Descriptions 241

Attachment 2. Additional Feature Development for the POM-IPXMT Linkage for the Lake Michigan Mass Balance Project – Task 1 Report 254

Attachment 3. Additional Feature Development for the POM-IPXMT Linkage for the Lake Michigan Mass Balance Project – Task 2 Report 258

Attachment 4 Additional Feature Development for the POM-IPXMT Linkage for the Lake Michigan Mass Balance Project – Final Report 261

Tables

A1 POMGL.OUT Standard Output File 234
A2 IPXMT Control File 235
A3 MAP.DIA Listing 237

Appendix A

Development of a Hydrodynamic/Water Quality (POM/IPX) Linkage for the Lake Michigan Mass Balance Project

Introduction

The model design for the Lake Michigan Mass Balance Project (LMMBP) is based on a linked submodel approach which, in part, includes hydrodynamic, sediment transport, eutrophication, and contaminants. It is the desire of the U.S. Environmental Protection Agency (USEPA) that the hydrodynamic model (HM) Princeton Ocean Model (POM) and water quality model (WQM) IPX linkage tasks follows the work performed by Water Quality and Contaminant Modeling Branch (WQCMB), Environmental Laboratory (EL), U.S. Army Corps of Engineers (USACE) on the Chesapeake Bay Eutrophication Study. The objectives of this research project have been to: (1) develop and implement a processor subroutine in the POM hydrodynamic model to provide input geometry, flow, and diffusion data for the IPX water quality model; (2) implement and test an ICM-type transport scheme in IPX; (3) perform linkage testing on simplified and prototype grids; and (4) document the linkage methodology and develop a user's guide. This report provides documentation on the concept, implementation, and use of the POM-IPX linkage and transport methodology.

CE-QUAL-ICM Transport Concept

Transport within the CE-QUAL-ICM (Cerco and Cole, 1995) is based on the integrated compartment method or box model methodology. The present version of ICM transport is a loose extension of the original WASP (Ambrose *et al*., 1986) code. The notion of utilizing the box model concept was retained in order to allow the coupling, *via* map files, of the WQM with hydrodynamic models of different dimensions and degrees of complexity. A basic feature of ICM is that transport is performed as a one-dimensional exchange between two adjacent cells through an individual cell face, irrespective of the dimensionality of the model. ICM handles horizontal and vertical transport during separate operations. The constituent transport equation that is solved is as follows:

$$\frac{\partial F}{\partial t} + \frac{\partial UF}{\partial x} + \frac{\partial WF}{\partial z} - \frac{\partial}{\partial x}\left(D_x \frac{\partial F}{\partial x}\right) - \frac{\partial}{\partial z}\left(D_z \frac{\partial F}{\partial z}\right) = 0 \qquad \text{(A.1)}$$

F = Constituent concentration
U = Horizontal cell face velocities
W = Vertical cell face velocities
D_x = Horizontal cell face diffusion coefficient
D_z = Vertical cell face diffusion coefficient
x = Horizontal dimension
z = Vertical dimension
t = Time

This transport equation is solved using the third-order accurate Non-Uniform Grid ULTIMATE QUICKEST algorithm in the horizontal and second-order implicit Crank-Nicholson scheme in the vertical. The details of these equations are presented below.

Non-Uniform Grid ULTIMATE QUICKEST Formulation

In the horizontal, the one-dimensional conservative control volume advective transport of a constituent F, integrated over a time step is written:

$$F_i^{n+1} = F_i^n - \frac{\Delta t}{\Delta x}\left(U_r F_r - U_l F_l\right) \tag{A.2}$$

where

F_i = Constituent concentration at a grid point
$F_{r,l}$ = Right and left cell face constituent concentrations
$U_{r,l}$ = Right and left cell face velocities
Δx = Horizontal grid spacing or box length
Δt = Time step

Recalling that transport within ICM is referenced to flow faces and not cells, mass balance for each cell is achieved by summing the flux contributions of each flow face about that cell. The estimate of cell faces mass fluxes are computed using the QUICKEST transport scheme. The QUICKEST algorithm presented by Leonard (1979) was originally derived using an upstream weighted quadrate interpolation function defined over three uniformly spaced grid points. This interpolation function estimates the value of cell face concentrations required by the conservative control volume transport scheme. For example, the right cell face concentration estimate for a flow positive to the right is written:

$$F_r = T_{i-1}F_{i-1} + T_i F_i + T_{i+1}F_{i+1} \tag{A.3}$$

where T_i are advective multipliers which weight the contribution of the three adjacent grid point concentrations. The advective multipliers are obtained by collecting the terms associated with each constituent value as defined by the QUICKEST advection operator. As an example, the advective multipliers for a uniform grid spacing QUICKEST approximation are simply T_{i+1} = 3/8, T_i = 3/4, and T_{i-1} = -1/8, where i-1, i, and i+1 correspond to Box IBL, Box IB, and Box JB, respectively.

In order to accommodate a non-uniform grid, a combination of two and three point Lagrangian divided differences interpolation functions (Henrici, 1964) are used to compute the QUICKEST estimate for the right cell face centered about Boxes IB and JB, specifically

$$F_r = P_1(x) - \frac{U\Delta t}{2} P_2'(x) + \left[D_x \Delta t - \frac{1}{6}\left[\Delta x^2 - \left(U\Delta t\right)^2 \right] \right] P_2''(x) \tag{A.4}$$

where

x = The local position of the right cell face with respect to the center of Box 1 or x_i
D_x = Diffusion coefficient

The quantities P_2' and P_2'' represent the gradient and curvature of the longitudinal distribution of the constituent F. Defining a local coordinate system of the three non-uniformly spaced grid cell denoted by x_{i-1}, x_i, and x_{i+1} with corresponding constituent values, the linear and quadratic interpolation functions required in Equation A.4 are written:

$$P_1(x) = \frac{\left(x - x_i\right)}{\left(x_{i+1} - x_i\right)} F_{i+1} + \frac{\left(x_{i+1} - x\right)}{\left(x_{i+1} - x_i\right)} F_i \tag{A.5}$$

and

$$P_2\left(x\right) = \frac{\left(x - x_i\right)\left(x - x_{i-1}\right)}{\left(x_{i+1} - x_i\right)\left(x_{i+1} - x_{i-1}\right)} F_{i+1} + \frac{\left(x - x_{i+1}\right)\left(x - x_{i-1}\right)}{\left(x_i - x_{i+1}\right)\left(x_i - x_{i-1}\right)} F_i$$

$$+ \frac{\left(x - x_{i+1}\right)\left(x - x_i\right)}{\left(x_{i-1} - x_{i+1}\right)\left(x_{i-1} - x_i\right)} F_{i-1} \tag{A.6}$$

Taking the first and second derivatives of $P_2(x)$ and substituting into Equation A.4, it is then possible to group terms and obtain the QUICKEST advective multipliers. For example, T_{i+1} multiplier is written:

$$T_{i+1} = \frac{\left(x - x_i\right)}{\left(x_{i+1} - x_i\right)} - \frac{U\Delta t}{2} \frac{\left[\left(x - x_i\right) + \left(x - x_{i-1}\right)\right]}{\left(x_{i+1} - x_i\right)\left(x_{i+1} - x_{i-1}\right)}$$

$$+ \frac{2\left(D_x \Delta t - \frac{1}{6}\left[\Delta x_i^2 - \left(U\Delta t\right)^2 \right] \right)}{\left(x_{i+1} - x_i\right)\left(x_{i+1} - x_{i-1}\right)} \tag{A.7}$$

Similar functions for the T_i and T_{i-1} multipliers are obtained by simply substituting the Lagrangian divided differences associated with F_i and F_{i-1}, respectively. The modification of the QUICKEST transport scheme utilized in ICM was originally developed and presented in Chapman (1988) with further applications and testing found in Chapman (1992), Dortch *et al*. (1991), and Chapman and Cole (1992).

The multipliers T_{i-1}, T_i, and T_{i+1} correspond to the ICM FORTRAN variables TERM(F,1), TERM(F,2), and TERM(F,3). The definition of remaining FORTRAN variables will be defined utilizing Equation A.7. In Equation A.7, the Lagrangian divided differences are defined in terms of box lengths (FORTRAN variable BL). For example, the first term in Equation A.7 (FORTRAN variable T3(F)) is defined as one-half of Box Length IB divided by the average of Box Length IB and Box Length JB. The divided differences in the second term are represented by the FORTRAN variables TP3(F,1)/DEN3(F,1), which are again defined as combinations of Box Lengths. Finally, the divided difference in the third term is simply 1/DEN3(F,1). The second subscript used in the FORTRAN variables defining the Lagrangian multipliers denotes positive (1), and negative (2) flow direction. This results from the fact that the upstream weighting of the QUICKEST scheme is direction sensitive.

In a similar fashion, a three-point approximation for the diffusive flux across a cell face is constructed utilizing the diffusive multipliers G_{i-1}, G_i and G_{i+1}. Following the rational of the advective multipliers, and that the diffusive fluxes defined in ICM are the product of a diffusion coefficient, cell face area, A, and the first derivative of the longitudinal constituent distribution, it is easily seen that:

$$G_{i+1} = D_x A \frac{\left[\left(x - x_i\right) + \left(x - x_{i-1}\right)\right]}{\left(x_{i+1} - x_i\right)\left(x_{i+1} - x_{i-1}\right)} \tag{A.8}$$

The FORTRAN variables used in ICM are:

GRAD(F,3) = DIFF(F)*A(F)*TP3(F,1)/DEN3(F,1)

Again, G_{l-1} and G_l (GRAD(F,1) and GRAD(F,2)) are obtained by substituting the Lagrangian divided difference in $P_2'(x)$ associated with the corresponding constituent value.

From a computational standpoint, it should be recognized that the geometric component of the multipliers in ICM are time invariant and, as a consequence, are computed once and stored in arrays described above. The time varying part of the multipliers (i.e., Q, Δt, D_x, A) are updated each time step during the computation of the advection (TERM) and diffusion (GRAD) arrays. The purpose of the ULTIMATE advection limiter as applied to QUICKEST is to eliminate the nonphysical overshoot and undershoot characteristics of nondiffusive advection operators (Leonard, 1991). The basic concept of the method is that the possibility of violation of local maxima or minima is avoided by limiting the cell face flux concentration so that it does not exceed nor underestimate the available constituent. This is accomplished by limiting the QUICKEST flux estimate to a value that will always maintain monotonicity. The formulation presented here is a modification of Leonard's original work specifically designed for the variable grid space version of ICM. Starting with Leonard's presentation of ULTIMATE for a constant grid space, we define the three point QUICKEST stencil comprised of F_u, F_c, and F_d, which are the upstream, central, and downstream constituent concentrations. The specification of the upstream, central, and downstream concentrations in terms of grid coordinates ILB, IB, JB, JRB in ICM depends on the sign of the velocity or flow. Given these conventions, the gradient and curvature of the concentration distribution are defined as GRAD(F) = F_d - F_u and CURV(F) = F_u - $2F_c$ + F_d. A limiting reference concentration, $F_{ref} = F_u + (F_c - F_u)/C$ is also defined where C is the absolute value of the local Courant number. Defining AGRAD and ACURV and the absolute

value of GRAD and CURV, respectively, the procedure for applying the ULTIMATE limiter is as follows:

1. Compute the normal QUICKEST cell face value, F_f.
2. If ACURV $\leq$ 0.6*AGRAD, use F_f.
3. If AGRAD > ACRUV, set $F_f = F_c$.
4. If AGRAD > 0.0, set $F_c \leq F_f \leq MIN(F_{ref}, F_d)$.
5. If AGRAD < 0.0, set $MAX(F_{ref}, F_d) \leq F_f \leq F_c$.

Step 2 represents cases where the existing concentration profile is smoothly varying and will maintain monotonicity using the regular QUICKEST flux estimate. Step 3 results in cases where there are extremely sharp gradients where monotonicity is violated and as a result the scheme reverts to upwind. Steps 4 and 5 represent the marginally monotonic situations in which the cell face concentration is constrained so that it does not violate a pure advection condition. The reference concentration is estimated by noting that for pure advection:

$$F_i^{n+1} = F_i^n - C(F_r - F_l) \tag{A.9}$$

Where F_r and F_l are the right and left cell face concentrations. For monotonicity:

$$F_i^{n+1} \leq F_{i-1}^n \tag{A.10}$$

and to a fair approximation, the concentration on the left face:

$$F_l \approx F_{i-1}^n \tag{A.11}$$

Substitution of Equations A.10 and A.11 into Equation A.9, with a little manipulation, results in a definition of the limiting reference concentration F_{ref}.

Utilizing this basic procedure, a modified version has been implemented in ICM. The first modification was to generalize the method for variable grid spacing. This was accomplished by defining a local coordinate system of the three non-uniformly spaced grid cells denoted by x_{i-1}, x_i, and x_{i+1} with corresponding constituent values. Adopting the Lagrangian interpolation approach previously developed for QUICKEST (Chapman, 1988; Chapman and Cole, 1992), a quadratic interpolation function (Henrici, 1964), $P_2(x)$ for the constituent concentration distribution is written:

$$P_2(x) = \frac{(x - x_i)(x - x_{i-1})}{(x_{i+1} - x_i)(x_{i+1} - x_{i-1})} F_{i+1} + \frac{(x - x_{i+1})(x - x_{i-1})}{(x_i - x_{i+1})(x_i - x_{i-1})} F_i$$

$$+ \frac{(x - x_{i+1})(x - x_i)}{(x_{i-1} - x_{i+1})(x_{i-1} - x_i)} F_{i-1} \tag{A.12}$$

Taking the first and second derivative of the interpolation function $P_2(x)$ and evaluating $P_2'(x)$ and $P_2''(x)$ about the point x_i, we obtain:

$$P_2'\left(x_i\right) = \frac{\left(x_i - x_{i-1}\right)}{\left(x_{i+1} - x_i\right)\left(x_{i+1} - x_{i-1}\right)} F_{i+1} + \frac{\left(x_i - x_{i+1}\right) + \left(x_i - x_{i-1}\right)}{\left(x_i - x_{i+1}\right)\left(x_i - x_{i-1}\right)} F_i$$

$$+ \frac{\left(x_i - x_{i+1}\right)}{\left(x_{i-1} - x_{i+1}\right)\left(x_{i-1} - x_i\right)} F_{i-1} \tag{A.13}$$

$$P_2''\left(x_i\right) = \frac{1}{\left(x_{i+1} - x_i\right)\left(x_{i+1} - x_{i-1}\right)} F_{i+1} + \frac{2}{\left(x_i - x_{i+1}\right)\left(x_i - x_{i-1}\right)} F_i$$

$$+ \frac{1}{\left(x_{i-1} - x_{i+1}\right)\left(x_{i-1} - x_i\right)} F_{i-1} \tag{A.14}$$

The variable grid space GRAD (named DELC in the code) and CURV (named CURL, CURP and CURM in ICM), respectively, can then be constructed by utilizing combinations of box lengths (BL's) in ICM. As an example, the quantity (x_{i+1} - x_i) in Equations A.13 and A.14 is simply the average of BL(IB(F),QD(F)) and BL(JB(F),QD(F)) in ICM. In a similar fashion to the original QUICKEST formulation, the components of the GRAD and CURV (Equations A.13 and A.14) are segregated into multipliers for each constituent value. For example, the GRAD multiplier for the (I+1) or (3) constituent is RATD(I), which equals the average of BL(ILB(F),QD(F)) and BL(JB(F),QD(F)) divided by the average of BL(IB(F),QD(F)) and BL(JB(F),QD(F)) for positive flows in ICM. These quantities are invariant and are computed once and stored in arrays. In order to account for the sense of the velocity on the computation and selection of the appropriate GRAD's and CURV's, three issues had to be addressed. The first was to arbitrarily define the GRAD to be a right minus left calculation. As a result, the correct sense of the GRAD is obtained by multiplying it by the SIGN of the velocity or flow. The second issue is the shift in the three cells utilized to compute the GRAD's and CURV's. The structure of ICM requires that separate functions are generated for positive and negative flow multipliers. The final issue is the computation of the reference quantity, F_{ref}. This is accomplished by noting the role of the up and downstream constituent concentrations are interchanged when the sign of the velocity changes.

An additional modification implemented that was not specifically addressed in Leonard (1991) was to require that the reference concentration F_{ref} always lies between F_c and F_u. Although violations of this constraint is an unusual situation when smoothly varying concentration distributions are being transported, it was encountered during testing of the algorithm.

Vertical Implicit Transport

Focusing on advective and diffusive transport in the vertical, the constituent transport can be written:

$$\frac{\partial F}{\partial t} + \frac{\partial WF}{\partial z} - \frac{\partial}{\partial z}\left(D_z \frac{\partial F}{\partial z}\right) = RHS \tag{A.15}$$

where RHS represents all horizontal transport. Integrating the transport equation in the vertical and time utilizing theta weighting for advection yields:

$$HF^{n+1} + \theta \Delta t \delta_z (WF^{n+1}) - \Delta t \delta_z \left(D_z \frac{\partial F^{n+1}}{\partial z}\right) = HF^* \tag{A.16}$$

where F* denotes all n-time level transport and H is the cell height. Expanding the differential operators in terms of central differences and collecting terms, Equation A.16 can be recast as follows:

$$A_{i+1}F_{i+1}^{n+1} + A_i F_i^{n+1} + A_{i-1}F_{i-1}^{n+1} = F^* \tag{A.17}$$

where

$$A_{i+1} = \frac{\Delta t}{H_{k,i}}\left(\frac{\theta W_{k,i}}{2} - \frac{D_{zk,i}}{AVH_k}\right) \tag{A.18}$$

where AVH is the average of adjacent cell heights.

$$A_i = 1 + \frac{\Delta t}{H_{k,i}}\left[\theta \frac{W_{k,i} - W_{k-1,i}}{2} + \frac{D_{zk,i}}{AVH_k} + \frac{D_{zk-1,i}}{AVH_{k-1}}\right] \tag{A.19}$$

and

$$A_{i-1} = -\left[\frac{\Delta t}{H_{k,i}}\left(\frac{\theta W_{k-1,i}}{2} + \frac{D_{zk-1,i}}{AVH_{k-1}}\right)\right] \tag{A.20}$$

The explicit portion of the algorithm, F* is written:

$$F^* = -(1-\theta)\Delta t \frac{W_n^k F_{k+1}^{\ n}}{2} \tag{A.21}$$

$$+1 - (1-\theta)\Delta t \frac{\left(W_k^n - W_{k-1}^n\right)F_k^n}{2} \tag{A.22}$$

$$+\left(1-\theta\right)\Delta t\left(\frac{W_{k-1}^{n}F_{k-1}^{n}}{2}\right) \tag{A.23}$$

where in terms of ICM coordinates, k-1, k, k+1 correspond to ILB, IB, JB, respectively.

The coefficients, A_k are computed once, stored in arrays, and used to update each constituent. This is accomplished by loading the theta-weighted explicit part of the algorithm F*, for each successive constituent and inverting the resulting matrix *via* a Thomas tridiagonal solver (Carnahan *et al*., 1969). The matrix coefficients, A_i, in the transport code are represented by the FORTRAN variable CT for l+1, BT for l, and AT for l-1. The explicit portion of advection F* is represented by DT using the provisional ULTIMATE QUICKEST updated CSTAR. In the code, the vertical velocities and diffusion coefficients found in Equations A.18 to A.22 are recast into Courant numbers CR2 and diffusion parameters GAMF and GAMFM1. The weighting factors that account for the use of these parameters and the variable cell heights are WAILB and WAIB for advection, and WDIB and WDILB for diffusion. In addition, it should be noted that the vertical loop is based on a vertical flow face counter.

POM-IPX Linkage

The POM-IPX linkage package consists of two components. The first is a set of mapping routines which builds the linkage files. The second is the POM processor subroutine that generates hydrodynamic input for the water quality model.

MAPPER Routines

The transport routines incorporated into IPX are written specifically for an unstructured grid geometry. The geometric relationships between grid cells are created external to IPX by a set of three subroutines located in the source file named MAPPER_SUBS.f that has been added to the POM hydrodynamic model.

These subroutines create several files that contain this "grid linkage" data. ICM_COOR.OUT, which is generated by subroutine ICM_GCF, contains basic cell geometry data that are used to create the remaining files. This file is retained because it is useful for creating grid contour plots. The files GEO.INP and MAP.INP, which are generated by subroutine ICM_BMF, contain data that are used directly by the IPX transport scheme. FILE95.INP contains data used by the POM hydrodynamics processor, WQMS.f, which generates the IPXMT input data. FILE95 results from a call to subroutine ICM_BUILD95. A diagnostic file named MAP.DIA is also created that contains the values of constants and parameters required by the IPX FORTRAN code. The IPX FORTRAN code contains PARAMETER statements that set the size of data arrays that must be adjusted to the grid. The user should note that if the grid geometry is altered in POM then the user should use MAP.DIA to adjust the PARAMETER statements in the IPX FORTRAN source file DIMS.INC. No other source code files need to be changed.

GEO, MAP, and FILE95 are generated automatically by POM when the user includes the string MAPPING as a command line argument (as in > pom MAPPING). A normal linked POM run occurs when the argument MAPPING is excluded from the command line. A general description of the linkage files is presented below.

ICM_COOR.OUT

Each line describes a cell with the I,J grid coordinate of the lower left-hand corner followed by eight floating point numbers representing the x,y coordinates of each of the four corners of the cell. This file is read in a free-field format.

GEO.INP

This file contains two sections of data. The first section specifies the cell identifier of each cell, the cell immediately above it, and its "type" identifier. Note that this is an original IPXMT variable. The second section lists cell number for each surface cell and the corresponding bottom cell number. An example of the GEO file is presented in Attachment 1, which includes the read formats.

MAP.INP

This file contains three sections. The first section specifies flow face information. For each flow face, the far upstream cell, the upstream cell, the downstream cell, and the far downstream cell identifiers are listed along with the I,J range of the flow faces (used for overlays). It is important to note the sign of the flow is assumed positive downstream. The second section specifies the number of vertical flow faces within each column. The third section lists the flow face identifier for each vertical flow face in a column. Here positive flow is up, or increasing face number. An example of the MAP is presented in Attachment 1, which includes the read formats.

FILE95.INP

This file contains three sections. The first section includes input directives for the processor WQMS.f and POM for creating the hydrodynamic flow field file. The first value represents the number of surface columns, NSB. The next value is the hydrodynamic averaging interval, NAVG. Then the iteration number when averaging begins, ITWQS, followed by ITT, the temperature output interval. The second section spans several lines and delimits the cell boundaries in terms of I,J coordinates. The first column represents the cell identifier. Columns 2 and 3 are the I dimension start and end for overlaying of cells. Columns 4 and 5 are the J dimension start and end for overlaying of cells. Column 6 represents the POM layer from which linkage data was obtained. Note that the values in Column 6 should always be 1. This serves as a check that the generated linkage is correct. The third section is an abbreviated form of the data included in the MAP.INP file discussed above. Whereas MAP.INP includes flow face descriptions for all flow faces, the WQMS.f processor only requires surface layer flow face data. FILE95.INP is read as a free-field format file. Additional details on the use of FILE95 are presented in the POM Linkage Processor discussion.

POM Linkage Processor

The modifications required in POM to accommodate the linkage processor subroutine were to: (1) insert labeled common WQM for the processing initiation iteration ITWQS, the hydrodynamic averaging interval NAVG, and the temperature output interval ITT; (2) insert POM cell to entry point WQMOUT within WQMS to output initial geometry data; (3) insert call to entry point WQMCVOL and WQMTVD within WQMS at iteration ITWQS to output initial cell volumes and

first set of time varying hydrodynamic data; (4) subsequent calls to WQMTVD are made every other POM time step to accommodate the three level time differencing scheme. If hydrodynamic output is time averaged, the processor will output IPX input at NAVG, which represents the number of two Δt time steps in the averaging interval. The processor subroutine will output the temperature array every ITT iterations subsequent to the initiation of hydrodynamic processing at iteration ITWQS. The two constraints on the parameter ITT is that it must (1) be equal to or greater than the WQMS averaging interval NAVG and (2) if it is greater than NAVG, it must be an even multiple. The option allows the user to output the temperature array at time scales where temperature changes become important, which offers significant potential for storage reduction. The calls to WQMCVOL and WQMTVD were inserted after the POM call to VERTVL.

The POM linkage processor subroutine has been designed to compute WQM cell lengths, cell volumes, cell face areas, and time averaged cell face flows and diffusion coefficients during the execution of the HM run. The definitions for cell lengths, areas, and volumes are identical to those within POM. In addition, a description of the tasks performed by the processor subroutine WQMS is provided below.

Task 1

The number of surface boxes NSB, averaging interval NAVG, the POM start iteration for processing ITWQS, and the temperature output interval counter ITT, are read from the map file FILE95. The IPX surface box number NB and the POM I,J coordinates IFIRST, ILAST, JFIRST, JLAST that define that box are then read. Although grid overlay or aggregation is not anticipated during the Lake Michigan study, the capability has been retained in the linkage processor. If an overlay grid is desired, the POM I,J coordinates specified are those that correspond to the aggregated IPX cell. An example record is shown below:

NB	IFIRST	ILAST	JFIRST	JLAST
1	2	3	2	3

In this example, the first surface IPX cell is defined by the POM I, J coordinates (2,2), (2,3), (3,2), and (3,3). If four POM cells (2 x 2) are aggregated to form a single IPX cell, ILAST and JLAST would have a value of 4. Next, the number of surface flow faces is read followed by flow face mapping parameters. These parameters include FN, the flow face number; QD, the direction normal to that flow face (X or Y); and the four IPX box numbers that bound the flow face IL, IQ, JQ, and JR. The definition of IQ and JQ is such that positive flow is from the IQ box to the JQ box. IL and JR are simply additional upstream and downstream boxes required by the QUICKEST formulation. The flow face overlay parameters are KP, which are the I or J coordinate of the POM grid point in the direction normal to the flow face. KF and KL are the span of the POM grid points which define the IPX flow face. An example of this record is shown below:

FN	QD	IL	IQ	JQ	JR	KP	KF	KL
1	1	0	1	2	3	3	2	2

This record shows that the first flow face (FN = 1) is normal to the X direction (QD = 1). Flow occurs between IPX cell IQ = 1 and JQ = 2 with IL = 0 defining a boundary cell. JR = 3 defines the next upstream cell given a negative flow. It is important to remember a sign convention

such as IQ to JQ defining a positive flow must be consistent everywhere within the grid. The IPX cell face is defined by the POM coordinates (3,2) and overlay sweep is over a single cell in the Y direction.

Task 2

During the initialization task, the real variable averaging interval AVGN is defined. Next, the column flow face variable FAREA is set to zero. The individual cell face variables AVGQ, the average horizontal flow, AVGDIF, the average horizontal coefficient, AVGZQ, the average vertical flow, and AVDIFZ, the average flow are set to zero.

Task 3

At this point, all time invariant or grid computations are performed. The first set of calculations are performed for all surface boxes. Initially, the surface box area BSAREA and the box lengths in the X and Y directions BL(SSB,1) and BL(SSB,2), respectively, are set to zero. The BSAREA is computed by summing the cell centered area array ART over the POM cells defined by the overlay parameters. Next, the box lengths are computed by a weighted sum of the overlay cell lengths DX or DY, scaled by the width of the individual cells. The column flow face area FAREA is computed by summing the product of the total depth, DT, at the cell face and the cell width DX or DY over the span of the overlay parameters. The above variables, along with the POM vertical grid spacings DZ are then output to the IPX hydrodynamic input file FILE96 in either binary or ASCII format. Finally, the initial column volumes CVOL are computed by summing the product of the individual total cell centered depth (H + ELB) and the cell area over the box overlay parameters and subsequently output to FILE96. If temperature output is desired, area averaged temperatures are computed and output to FILE97.

Task 4

Time varying computations are performed each call to entry point WQTVD. At this time, the averaging iteration counter IKNT is incremented and the temporary spatial averaging arrays BF and DF are set to zero. Next, the horizontal cell face flows and diffusion coefficients are computed by multiplying the local velocity (U or V) and diffusion coefficients (AAM) times the product of DZ and DFACE, where DZ is the POM vertical grid spacing and DFACE is the column cell face area. Spatial averaging again occurs of the overlay parameter range. The temporary BF and DF arrays are then redefined as time averaging arrays AVGQ and AVGDIF, which are assigned flow face numbers consistent with the ICM ordering convention. Three points to note are: (1) the temporary vertical array counter k is inverted to accommodate the fact that POM assigns k top to bottom and ICM requires flow faces numbered bottom to top; (2) the parameter KB in POM represents the number of cell interfaces, therefore, the number of POM layers is KB-1 or the parameter KBM1; and (3) the weighted average horizontal diffusion coefficients are computed by scaling the individual cell face areas by the sum of the overlay cell face areas. Temporary vertical flows ZQ and diffusion coefficients DIFZ are then computed by multiplying the vertical velocity and diffusion coefficients by the horizontal cell face area. The temporary ZQ and DIFZ arrays are then redefined as time averaging arrays AVGZQ and AVDIFZ, which are assigned flow face numbers consistent with the ICM ordering convention. Again, the same three points discussed above are applicable here. At this time, the increment counter IKNT is checked to see if it equals the averaging NAVG. If not, the computation is returned to POM. If it is, the column volumes CVOL are updated, the time varying data are

written to FILE96. If the temperature output interval is satisfied, area averaged temperatures are output to FILE97. Finally, the average flow and diffusion arrays are set to zero and the computation is returned to POM.

IPXMT Modifications

IPX Linkage and Input Modifications

The modifications required within IPX to enable implementation of ICM transports are as follows: (1) subroutines INITB, INITC, INITD, INITE and their associated data files have been replaced by subroutine WINITIALIZE and the ICM input files. WINITIALIZE is called within subroutine IPXINT after the call to INITA; (2) subroutines ADVECTION, QSURF, QSURF1, QSURF2, and EXCHANGES have been replaced by WTRANSPORT. The call to WTRANSPORT occurs within subroutine DERIV; (3) the function of subroutines FLOWS, IFLOWS, RFLOWS, and QFLOWS have been replaced by subroutine HYDRO within WTRANSPORT; and (4) the include file DIMS.INC has been modified to include the ICM parameter declarations. As previously mentioned, the time varying POM temperature array is made available to IPX *via* a modification within the POM processor subroutine WQMS. The temperature array TEMPERATURE is handled by two subroutines, WTEMP_INIT and WTEMP. WTEMP_INIT is called prior to WINITIALIZE where the temperature file initially is opened and the update counters are initialized. The time varying temperature reads are performed by WTEMP which is called by subroutine DERIV.

IPXMT Transport Description

Input and output options available within IPXMT are discussed in Attachment 1. This information is a modification of that presented by Cerco and Cole, 1995. The basic link between IPXMT and ICM transport are the concentration array C and the mass derivative CD. This is accomplished by the IPXMT subroutine call to WTRANSPORT where C and CD are passed to WTRANSPORT. Subsequent to transport, the mass derivative array is updated and passed back to IPXMT. The first operation within the transport routine is a simulation time check to determine if boundary conditions are to be updated by the subroutine TVDS. Next, boundary data is interpolated for the current model simulation time. If a flow update interval time is reached, subroutine HYDRO is called. HYDRO reads column volumes, flows, and diffusivities prior to the computation of individual cell volumes and assigns flow direction using the logical variable POSITIVE_FLOW for each flow face. If the auto time step option is selected, WTRANSPORT computes the new time step. Subsequent to the time step update, time varying geometry arrays such as cell volumes, vertical cell heights, the vertical cell face depth below the surface, and the horizontal QUICKEST multipliers are recomputed. Next, boundary conditions are updated. The ULTIMATE QUICKEST computations are performed and the horizontal transport concentration array CSTAR is updated. The time varying multipliers for implicit vertical transport are then computed, the tridiagonal matrix is assembled, and a Thomas matrix reduction is performed to update the constituent concentration array C2. Finally, the mass derivative change due to transport is then computed, Courant numbers, volumes and time parameters are reset, which completes the transport sequence.

Guide to the Use of the Linked POMGL-IPXMT

The purpose of this section is to provide a step-by-step procedure for running the linked POMGL-IPXMT modeling system. There are three main tasks that must be performed to complete model simulation. These tasks are the construction of linkage files, hydrodynamic output generation, and water quality model simulations. The documentation provided here assumes the linked modeling system is being run on a UNIX machine.

1. Prior to running either the linkage file generation or linked POMGL simulation, a number of parameters must be set in the linkage subroutines. First subroutine, ICM_BMF within MAPPER- SUBS requires that the parameters:

   NX = IM
   MY = JM
   MAX_NUMBER_OF_LAYERS = KBM1
   MAX_NUMBER_OF_BOXES = NFSM*KBM1 (MINIMUM)
   MAX_NUM_BOXES_LAYER = NFSM (MINIMUM)

   Next, subroutine WQMS requires that:

   NBP = NFSM (MINIMUM)
   NFP = NDUM + NDVM (MINIMUM)

   All of the above POM parameters can be found in the standard output file POMGL.OUT, which is presented in Table A1.

2. Subsequent to compiling using CL_POM.SH, the following command will automatically generate the MAP.INP, GEO.INP, MAP.DIA, and FILE95.INP using the MAPPING option.

   POM MAPPING < POMGL.IN > POMGL.OUT

   This command also generates the POM standard output file POMGL.OUT, which can be compared to MAP.DIA in order to verify the integrity of the linkage. The details of this comparison are presented in the prototype grid tests section. At this point, MAP.INP, GEO.INP, MAP.DIA should be copied to the IPXMT directory. In addition, the MAP.INP and GEO.INP must be renamed to conform to the MAP and GEO file names declared in the IPXMT control file WQM_CON.NPT (Table A2).

3. Prior to performing a normal linked POMGL simulation, the first line of FILE95 must be edited. Example free field values of the variables NAVG, ITWQS, ITT are provided within FILE95 as a reminder that they must be determined by the user. Again, NAVG is the hydrodynamic averaging or output interval, ITWQS is the number of spinup iterations, and ITT is the temperature output interval used by POMGL. As a rule of thumb:

   NAVG = 3600/(2*DTI)*(Averaging interval in hours)
   ITWQS = 3600/DTI*(Hours of spinup)
   ITT = NAVG or greater even multiple

   where DTI is the POM internal mode time step.

Table A1. POMGL.OUT Standard Output File

```
PRINCETON OCEAN MODEL - GREAT LAKES
IN  =  53 JM  =  102          KB  =  20
          [POMGL CONTROL]
BATHYMETRY FILE FGRID=
          mich5.dat
INPUT FILE PREFIX FINPUT=
          m94121
OUTPUT FILE PREFIX FOUT=
          m94121
INITIAL CONDITION FILE FINIT=
          m94061.rst
MODE  =  3

START YEAR  =                    1994
START DAY  =                     121
START HOUR  =                    0
DTE  =                           30.0
DTI  =                           300.00
IHRS  =                          144

DTWIND  =                        1.00
NLEVEL  =                        20
HMAX  =                          262.00
DS  =                            5000.0

[POMGL CONSTANTS]
ISPADV  =                        1
HORCON  =                        0.1000
TPRNU  =                         1.00
SMOTH  =                         0.100E-05
UMOL  =                          10.0
TBIAS  =                         10.0
SBIAS  =                         35.00
ALPHA  =                         0.225
RAMPDT  =                        0.00
DADD  =                          0.50
AAMZ  =                          50.00
CBCMIN  =                        0.00250
IRSTRT  =                        1
IPSWR  =                         1
ITMEAN  =                        288
ISFLUX  =                        0
SMAGOR  =                        0.00
YAMADA  =                        0.000E+00
YAMADA1  =                       0.000E+00
IAVG  =                          36

MASK ARRAY NONZERO ELEMENTS:
NFSM=2318   NDUM=2162   NDVM=2223
SIGMA LEVELS
K        Z                ZZ               DZ               DZZ
1        0.000            -0.011           0.023            0.023
2        -0.023           -0.034           0.023            0.023
3        -0.045           -0.057           0.023            0.023
4        -0.068           -0.079           0.023            0.023
```

Table A1. POMGL.OUT Standard Output File (Continued)

5	-0.091	-0.102	0.023	0.023
6	-0.113	-0.125	0.023	0.023
7	-0.136	-0.148	0.023	0.023
8	-0.159	-0.170	0.023	0.023
9	-0.182	-0.193	0.023	0.023
10	-0.204	-0.216	0.023	0.034
11	-0.227	-0.250	0.045	0.057
12	-0.272	-0.306	0.068	0.079
13	-0.340	-0.386	0.091	0.102
14	-0.431	-0.488	0.113	0.125
15	-0.545	-0.613	0.136	0.125
16	-0.681	-0.738	0.113	0.102
17	-0.794	-0.840	0.091	0.079
18	-0.885	-0.919	0.068	0.057
19	-0.953	-0.977	0.047	0.047
20	-1.000	-1.023	0.000	0.000

Table A2. IPXMT Control File

```
TITLE C.....................................TITLE...................................
Test grid water quality model
T2
T3
T4
T5
test grid from POM run
GEOM DEFINE     NB          NSB         NQF         NSHQF       NL
                44042       2318        125039      83315       19

TIME CON        TMSTRT      TIMEEND     in days
                0.0         1.00

# DLT           NDLT
                1

DLT DAY         DLTD        DLTD        DLTD        DLTD        DLTD        DLTD        DLTD        DLTD        DLTD
                0.0         800.0

DLT VAL         DLTVAL      DLTVAL      DLTVAL      DLTVAL      DLTVAL      DLTVAL      DLTVAL      DLTVAL      DLTVAL
                3600.0      600.0

DLT MAX         DLTMAX      DLTMAX      DLTMAX      DLTMAX      DLTMAX      DLTMAX      DLTMAX      DLTMAX      DLTMAX
                7200.0      2000.0

DLT FIN         DLTFTN      DLTFTN      DLTFTN      DLTFTN      DLTFTN      DLTFTN      DLTFTN      DLTFTN      DLTFTN
                0.90

HM DLT          AHMDLT      HMEND
                7200.0      1.00

TEMP DLT        ATDLT
                7200.0
```

Table A2. IPXMT Control File (Continued)

DIAGNSTICS	DIAC ON	NDIA 1							
DIA DAY	DIAD 0.	DIAD	DIAD	DIAD	DIAD	DIAD	DIAD	DIAD	DIAD
DIA FREQ	DIAF 0.01	DIAF 30.43	DIAF	DIAF	DIAF	DIAF	DIAF	DIAF	DIAF
HYD SOLTN	SLC QUICKEST	TH 1.0							
CONTROLS	AUTOC ON	VBC ON							
DEAD SEA	FLC ON	XYDFC ON	ZDFC ON						
HDIFF	XYDF 10.0	ZDFMUL 1.0	ZDFMAX 0.1						
BOUNDARY	BCC OFF								
BOUNDARY	BNDCC FIXED								
BOUNDARY	BNDSC UPWIND	BNDSC UPWIND	BNDSC UPWIND	BNDSC UPWIND	BNDSC	BNDSC	BNDSC	BINDSC	BNDSC
ACT CST	ACC OFF	ACC ON	ACC OFF	ACC OFF	ACC	ACC	ACC	ACC	ACC
#FILES	NHYDF 1	NTVDF 1	NTMPF 1						

MAP FILE..... MAPFN .....
wqm_map.inp

GEO FILE..... GEOFN .....
wqm_geo.inp

HYD FILE..... HYDFN .....
wqm_hydro.inp

TMP FILE..... TMPFIN .....(Temp from POM).....
wqm_temp.in

CBC FILE..... CBCFN .....
wqm_cbc.50b

DIA FILE..... DIAFN .....
wqm_dia.opt

4. A normal linked POM simulation can then be performed using the following command line:

   POM < POMGL.IN > POMGL.OUT

   The completed linked POM simulation will result in two output files, fort.96, the processed IPXMT hydrodynamic input data, and fort.97, the IPXMT temperature input data. At this point, fort.96 and fort.97 should be copied to the IPXMT directory and renamed to conform to the HYD and TMP file names declared in the IPXMT control file WQM_CON.NPT (Table A2).

5. In addition to the IPXMT geometry parameter provided at the bottom of MAP.DIA which is presented in Table A3, the following consistent time step data must be specified in WQM_CON.NPT:

   DLTVAL $\leq$ DLTMAX
   DLTMAX $\leq$ 2*DTI*NAVG
   AHMDLT 2*DTI*NAVG
   ATDLT = 2*DTI*ITT

6. The additional input data required to run the IPXMT transport is shown in Table A2, which represents a subset of the original CE-QUAL-ICM control file (Cerco and Cole, 1995). A discussion of these inputs specific to IPXMT transport is presented in Attachment 1.

7. At this point, the IPXMT transport can be executed.

Table A3. MAP.DIA Listing

Total number of boxes and flow faces

box	ff
44042	125039

Number of boxes for each layer. Numbers are included and cumulative.

layer	first	last
1	1	2318
2	2319	4636
3	4637	6954
4	6955	9272
5	9273	11590
6	11591	13908
7	13909	16226
8	16227	18544
9	18545	20862
10	20863	23180
11	23181	25498
12	25499	27816
13	27817	30134
14	30135	32452
15	32453	34770

Table A3. MAP.DIA Listing (Continued)

16	34771	37088
17	37089	39406
18	39407	41724
19	41725	44042

Number of horizontal flow faces for each layer. Numbers are inclusive and cumulative.

layer	first	last
1	1	4385
2	4386	8770
3	8771	13155
4	13156	17540
5	17541	21925
6	21926	26310
7	26311	30695
8	30696	35080
9	35081	39465
10	39466	43850
11	43851	48235
12	48236	52620
13	52621	57005
14	57006	61390
15	61391	65775
16	65776	70160
17	70161	74545
18	74546	78930
19	78931	83315

PARAMETERS (place in dims.inc)

NBP	=	44042
NQFP	=	125039
NHQP	=	83315
NSBP	=	2318
NLP	=	19

CONSTANTS (place in wqm_con.npt)

NB	=	44042
NSB	=	2318
NQF	=	125039
NHQF	=	83315
NSHQF	=	4385
NLP	=	19

Linkage Quality Assurance Testing

Simplified Grid Tests

Initial testing of the linkage methodology was performed utilizing a 20 x 20 x 10 test grid. The POM grid coordinate and depth file and IPXMT map file were visually inspected to ensure that they were identical. The POM hydrodynamic output file was read by IPXMT and a volume balance was performed. The comparison of POM and IPXMT grid cell volumes were identical within machine accuracy. ULTIMATE QUICKEST mass conservation testing was performed within IPXMT. Specifically, uniform concentration and spot dump mass conservation tests were performed. The first set of mass conservation tests utilized an one-hour time step in POM with no time averaging performed. This resulted in a two-hour hydrodynamic update interval in IPXMT. Subsequent mass conservation tests were performed utilizing six-hour average POM flow data, which resulted in a 12-hour hydrodynamic data update in IPXMT. During all tests mass conservation was maintained within machine accuracy.

Prototype Grid Tests

Unlike the simplified grid linkage quality assurance testing, the prototype grid due to its size required the development of a formal procedure. The POMGL grid consists of 53 x 10^2 5 km square grid cells in the horizontal with 19 sigma layers. Subsequent to the mapping process previously described, the following steps were taken to ensure the proper linkage of the POMGL grid and hydrodynamics with IPXMT. The number of model layers, cells, and flow faces were compared. Utilizing data provided by the POMGL standard output file (Table A1), the number of cells and flow faces active during the model simulation can be identified. Active surface elevation and horizontal velocity component computations are controlled by the elements of the mask arrays FSM, DUM, DVM, which are set to one for active cells and flow faces and zero for land boundaries.

During the initial geometry computations within POMGL, the mask arrays are determined from the bathymetric array H(I,J). The number of active cells and surface flow faces are accumulated as the variables NFSM, NDUM, and NDVM. A comparison of MAP.DIA (Table A3) and POMGL.OUT can be used to demonstrate the equivalence of the POMGL and IPXMT grids. The number of active POMGL columns (NFSM) and IPXMT surface boxes (NSB) is 2,318. The number of horizontal POMGL surface cell flows (NDUM + NDVM) is 4,385, which is identical to the number of flows in the surface layer (NSHQF) shown in MAP.DIA. Noting that the number of grid layers in POMGL corresponds to KB -1, it is seen that both grids contain 19 vertical sigma layers.

General Analytic Testing

The integrity of the linkage files was first checked by performing a seven-hour POMGL simulation utilizing a 300 second time step and no averaging. After six hours of model spinup, the IPXMT processor, WQMS, was enabled producing a HYDRO file (fort.96) with a 600 second average hydrodynamic update interval (AHMDLT). ULTIMATE QUICKEST mass conservation and volume balance testing was performed within IPXMT. Uniform concentration and spot dump mass conservation tests were simulated. The mass conservation tests utilized: (1) a fixed 600 second time step; and (2) auto-time stepping in IPXMT. During all tests, both mass

and volume conservation were maintained within machine accuracy. Next, a six-day POMGL simulation utilizing a 300 second time step and one-hour time averaging was performed. After five days of model spinup, WQMS was enabled, which resulted in a two-hour AHMDLT. Mass conservation tests were performed using a fixed one-hour step and auto-time stepping in IPXMT. Again, all tests maintained mass and volume conservation within machine accuracy.

References

Ambrose, R.B., S.B. Vandergrift, and T.A. Wool. 1986. WASP3, A Hydrodynamic and Water Quality Model - User's Manual and Programmer's Guide. U.S. Environmental Protection Agency, Environmental Research Laboratory, Athens, Georgia. EPA-600/3-86/034, 379 pp.

Carnahan, B., H. Luther, and J.O. Wilkes. 1969. Applied Numerical Methods. John Wiley and Sons, New York, New York. 604 pp.

Cerco, C. and T. Cole. 1995. User’s Guide to the CE-QUAL-ICM Three-Dimensional Eutrophication Model. U.S. Army Corps of Engineers, U.S. Army Engineer Waterways Experiment Station, Vicksburg, Mississippi. Technical Report EL-95-15, 2,420 pp.

Chapman, R.S. 1988. Analysis of the Numerical and Physical Mixing Characteristics of the WASP Box Model. Final Report. U.S. Army Corps of Engineers, U.S. Army Waterways Experiment Station, Vicksburg, Mississippi.

Chapman, R.S. 1992. Research and Development Towards Improving the Predictive Capability of CE-QUAL-W2. Final Report. U.S. Army Corps of Engineers, U.S. Army Waterways Experiment Station, Vicksburg, Mississippi.

Chapman, R.S. and T.M. Cole. 1992. Improved Thermal Predictions in CE-QUAL-W2. In: M. Jennings and N.G. Bhowmik (Eds.), Proceedings of the Hydraulic Engineering Sessions at Water Forum '92, pp. 158-163. American Society of Chemical Engineers, Baltimore, Maryland.

Dortch, M.S., R.S. Chapman, and S.R. Abt. 1991. Application of Three-Dimensional Residual Transport. J. Hydraul. Engin., 118(6):831-848.

Henrici, P. 1964. Elements of Numerical Analysis. John Wiley and Sons, New York, New York. 328 pp.

Leonard, B.P. 1979. A Stable and Accurate Convective Modelling Procedure Based on Upstream Interpolation. Comp. Methods Appl. Mechan. Engin., 19(1):59-98.

Leonard, B.P. 1991. The Ultimate Conservative Difference Scheme Applied to Unsteady One-Dimensional Advective. Comp. Methods Appl. Mechan. Engin., 88(1):17-74.

Attachment 1

IPXMT Transport Input, Map File, and Geo File Descriptions

The Control File

The control file contains the parameters used to run CE-QUAL-ICM. It consists of card images 80 characters in length. The format of the input files has been developed in order to take advantage of a full-screen text editor. The control file begins with three lines which can be used for file identification and are ignored by the program. The rest of the file consists of a line which contains the card identification and the names of the FORTRAN variables associated with the input card. The FORTRAN names are right-justified according to the field widths associated with the input variables. The next line consists of the actual input values. There are 10 input fields associated with each card, although the first field is not used. Each field has a length of eight characters. The next line is left blank for ease of display in a full-screen editor.

There are no optional cards in the control file – each card is required, although there may be no values associated with the card. The following contains a description of each card. All character inputs must be capitalized except the TITLE cards or the variable will take on the default value.

Title Cards (TITLE)

Field	Name	Value	Description
1-72	TITLE	Character	Text for identification of simulation

There are six title cards that help identify the model run. Each line contains up to 72 characters of text. The title card appears in every output file except for restarts. Uses for the title card include identifying the application, the dates of the application, the date of the simulation, and any other information specific to the simulation.

Example:

```
TITLE   C.......................................................Title.......................................................
Chesapeake Bay Water Quality control file input
30 year run
Using kinetic values from final calibration
March 14, 1994
Extra title line
Extra title line
```

Geometry Definition (GEOM DEFINE)

Field	Name	Value	Description
1	NB	Integer	Number of boxes in grid
2	NSB	Integer	Number of surface boxes in grid
3	NQF	Integer	Number of flow faces in grid
4	NHQF	Integer	Number of horizontal flow faces in grid
5	NSHQF	Integer	Number of horizontal flow faces in surface layer of grid
6	NL	Integer	Maximum number of layers in grid

This card specifies the dimensions of the water quality model grid. The user must take care that the values specified here are consistent with corresponding PARAMETER statements in the file WQM_COM.INC.

Example:

GEOM DEFINE	NB	NSB	NQF	NHQF	NSHQF	NL
	30	10	51	31	10	3

Time Control (TIME CON)

Field	Name	Value	Description
1	TMSTRT	Real	Starting simulation date (Julian day)
2	TMEND	Real	Ending simulation date (Julian day)

This card specifies the starting and ending dates of the simulation.

Example:

TIME CON	TMSTRT	TMEND
	0.0	365.0

Number of Time Step Intervals (# DLT)

Field	Name	Value	Description
1	NDLT	Integer	Number of time step intervals

The model provides the option to vary the time step. The time step may be varied through the autostepping option or at discrete, user-specified intervals. This card specifies the number of intervals in which the time step, maximum time step, and/or the fraction of the calculated time step vary.

This card group establishes a pattern followed by successive groups which govern output. The first card establishes that time steps are assigned three times. The second card names the times as days 0, 120, and 250. The third card assigns a time step of 3600 seconds at day 0, 1800 seconds at day 120, 3600 seconds at day 250.

Example:

```
# DLT    NDLT
           3
```

Time Step Date (DLT DAY)

Field	Name	Value	Description
1	DLTDAY	Real	Date of time step interval (Julian day)

This card specifies the intervals in which the time step, maximum time step, and/or fraction of the maximum calculated time step may vary.

Example:

```
DTL   DAY    DLTD     DLTD     DLTD     DLTD     DLTD     DLTD     DLTD     DLTD
              0.0     120.0     250.0
```

Time Step Value (DLT VAL)

Field	Name	Value	Description
1	DLTVAL	Real	Time step (sec)

This card specifies the initial time step if autostepping is used, or the values for the time step if autostepping is turned off.

Example:

```
DLT   VAL    DLTVAL   DLTVAL   DLTVAL   DLTVAL   DLTVAL   DLTVAL   DLTVAL   DLTVAL
             3600.0   1800.0    3600.0
```

Maximum Time Step (DLT MAX)

Field	Name	Value	Description
1	DLTMAX	Real	Maximum time step (sec)

This card specifies the maximum value for the time step if autostepping is turned on.

Example:

```
DLT   MAX    DLTMAX   DLTMAX   DLTMAX   DLTMAX   DLTMAX   DLTMAX   DLTMAX   DLTMAX
             7200.0   7200.0   7200.0
```

Time Step Fraction (DLT FTN)

Field	Name	Value	Description
1	DLTFTN	Real	Fraction of calculated time step used

This card specifies the fraction of the time step used if autostepping is turned on. The autostepping algorithm estimates the maximum time step in order to maintain stability. Since this is an estimate, DLTFN can be used to decrease the time step if the model becomes unstable when autostepping is turned on.

Example:

```
DLT   FTN    DLTFTN   DLTFTN   DLTFTN   DLTFTN   DLTFTN   DLTFTN   DLTFTN   DLTFTN
             0.95     0.80     0.95
```

Hydrodynamic Model Time Step (HM DLT)

Field	Name	Value	Description
1	AHMDLT	Real	Hydrodynamic update interval (sec)
2	HMEND	Real	Ending date of hydrodynamic file (Julian day)

This card specifies the interval at which hydrodynamic information is input to the model. This information is employed only if “BINARY” or “DEPTH-AVERAGE” options are specified (see card HYD MODEL). The present formulation requires a constant update interval. The ending date of the hydrodynamic input (HMEND) is necessary to synchronize elapsed time in the model, time determined from hydrodynamic updates, and update times specified on various input files. If multiple hydrodynamic files are employed, they should be as close to one year in length as possible.

Example:

```
HM    DLT    AHMDLT   HMEND
             0.0       364.52
```

Diagnostics Output Control (DIAGNSTCS)

Field	Name	Value	Description
1	DIAC	Character	Specifies if output is written to diagnostic file
2	NDIA	Integer	Number of diagnostic intervals

This card specifies if information is output to the diagnostics file and the number of intervals in which the frequency of the output varies.

Example:

```
DIAGNSTCS   DIAC     NDIA
            ON       4
```

Diagnostics Date (DIA DAY)

Field	Name	Value	Description
1-9	DIAD	Real	Date of diagnostics interval (Julian day)

This card specifies the intervals at which the diagnostics output varies. If there are more intervals than can be specified on one line, then they are continued on the next line.

Example:

```
DIA   DAY   DIAD    DIAD    DIAD    DIAD    DIAD    DIAD    DIAD    DIAD
            0.0     100.0   101.0   200.0
```

Diagnostics Frequency (DIA FREQ)

Field	Name	Value	Description
1-9	DIAF	Real	Diagnostics output frequency (Julian day)

This card specifies the frequency at which diagnostics information is output. If there are more values than can be specified on one line, then they are continued on the next line.

Example:

```
DIA   FREQ  DIAF    DIAF    DIAF    DIAF    DIAF    DIAF    DIAF    DIAF
            7.0     0.1     7.0     1.0
```

Hydrodynamic Model (HYD MODEL)

Field	Name	Value	Description
1	HYDC	Character	Specifies the type of hydrodynamic model input

This card specifies the nature of hydrodynamic inputs to the water quality model. There are three options: “BINARY”, “ASCII”, and “DEPTH_AV”. The BINARY option is usually specified for large files generated by a hydrodynamic model, e.g., CH3D-WES. The ASCII option is usually specified for smaller files which are created independent of a hydrodynamic model. The DEPTH_AV option indicates a two-dimensional (longitudinal-lateral) implementation of the water quality model with binary input format.

Example:

```
HYD MODEL   HYDC
            BINARY
```

Hydrodynamic Solution (HYD SOLTN)

Field	Name	Value	Description
1	SLC	Character	Specifies the type of transport solution
2	TH	Real	Crank-Nicholson weighting factor

This card allows the user to specify the type of longitudinal transport scheme that the water quality model uses, how mass is conserved, and the amount of implicit/explicit weighting used in vertical transport. The options for the longitudinal transport scheme are “UPWIND” and “QUICKEST”. The upwind option is numerically diffusive which can cause problems during calibration. The QUICKEST option reduces numerical diffusion but can cause overshoots and undershoots in regions of sharp gradients. Theta (“TH”) specifies the amount of explicit/implicit time weighting in the vertical transport scheme. A value of zero is fully explicit and a value of one is fully implicit.

Example:

```
HYD   SOLTN  SLC          TH
             QUICKEST     1.0
```

Miscellaneous Coefficients (MISC)

Miscellaneous Controls (CONTROLS)

Fields	Name	Value	Description
1	AUTOC	Character	Turns on/off autostepping
2	VBC	Character	Turns on/off volume balance calculations
3	ICIC	Character	Specifies format of initial conditions

The first five fields must be specified as “ON” or “OFF”. The third field may be “UNIFORM”, “VARIED”, or “BINARY”. Descriptions of these options are provided in the description of the Initial Conditions File.

Example:

```
CONTROLS   AUTOC   VBC     ICIC
           ON      ON      UNIFORM
```

Dead Sea Case (DEAD SEA)

Fields	Name	Value	Description
1	FLC	Character	Turns on/off hydrodynamic flows
2	XYDFC	Character	Turns on/off horizontal diffusion
3	ZDFC	Character	Turns on/off vertical diffusion

This card allows the user to turn off any or all transport processes in the model. All fields must be specified as “ON” or “OFF”.

Example:

```
DEAD SEA   FLC     XYDFC   ZDFC
           ON      ON      ON
```

Horizontal Diffusion (HDIFF)

Fields	Name	Value	Description
1	XYDF	Real	Value for horizontal diffusion ($m^2\ s^{-1}$)
2	ZDFMUL	Real	Multiplier for vertical diffusion
3	ZDFMAX	Real	Maximum vertical diffusion ($m^2\ s^{-1}$)

The first application of the water quality model was in a system with negligible horizontal dispersion. To minimize the size of the hydrodynamic file, dispersion was specified once in the control file rather than written out by the hydrodynamic model at every cell interface, at every time step. This feature is still in place when the BINARY or DEPTH_AV hydrodynamic options

are employed. Spatially variable dispersion is a feature of the model code but the user must perform revisions to read dispersion in binary format. When the ASCII option is employed, spatially- and temporally-varying horizontal dispersion is specified in the hydrodynamic file and the field on this card is ignored.

In the first application of the model, sensitivity runs were performed with varying vertical diffusion. To avoid running the hydrodynamic model repeatedly, a vertical diffusion multiplier was installed in the water quality model. For similar reasons, specification of maximum vertical diffusion was allowed.

Example:

```
HDIF        XYDF    ZDFMUL  ZDFMAX
            10.0    1.0     0.1
```

Constituent Input (CST INPUT)

Fields	Name	Value	Description
1	BCC	Character	Turns on/off boundary inputs

This card allows the user to selectively turn on/off constituent inputs. All fields must be specified as “ON” or “OFF”.

Example:

```
CST   INPUT   BCC
              ON
```

Boundary Concentration (BOUNDARY)

Fields	Name	Value	Description
1	BNDTC	Character	Specifies how boundary concentrations are specified

This card specifies if input boundary concentrations are updated when the next value is read in from the time varying data file or if the concentrations are interpolated for each time step. These inputs are "STEP" or "INTERP".

Example:

```
BOUNDARY    BNDTC
            STEP
```

Boundary Solution (BOUNDARY)

Fields	Name	Value	Description
1-22	BNDSC	Character	Specifies how boundary concentrations are calculated

This card allows the user to specify how boundary concentrations are handled by the transport scheme. The values can be "UPWIND", "EXTRAP", or "REFLECT".

Example:

```
BOUNDARY   BNDSC   BNDSC   BNDSC   BNDSC   BNDSC   BNDSC   BNDSC   BNDSC
           UPWIND  UPWIND  UPWIND  UPWIND  UPWIND  UPWIND  UPWIND  UPWIND
           UPWIND  UPWIND  UPWIND  UPWIND  UPWIND  UPWIND  UPWIND  UPWIND
           UPWIND  UPWIND  UPWIND
```

Active Constituents (ACT CST)

Field	Name	Value	Description
1-22	ACC	Character	Turns on/off constituent calculations

This card allows the user to turn on/off a state variable for a simulation. Care must be taken when exercising this option because of the interaction of the state variables in the kinetic formulations. Most commonly, this option is used when initially calibrating the water quality model transport to the hydrodynamic model transport using salinity. This option turns on/off computations only. All state variables must still be included in load files, boundary conditions, etc. The state variables and their order are:

1	Temperature
2	Salinity
3	Iron + Manganese (for suspended solids)
4	Cyanobacteria
5	Diatoms
6	Green algae
7	Dissolved organic carbon
8	Labile particulate carbon
9	Refractory particulate carbon
10	Ammonium
11	Nitrate-nitrite
12	Dissolved organic nitrogen
13	Labile particulate nitrogen
14	Refractory particulate nitrogen
15	Total phosphate
16	Dissolved organic phosphorus
17	Labile organic phosphorus
18	Refractory organic phosphorus
19	Chemical oxygen demand
20	Dissolved oxygen
21	Particulate silica
22	Dissolved silica

Example:

```
ACT   CST   ACC     ACC     ACC     ACC     ACC     ACC     ACC     ACC
            ON      ON      ON      ON      ON      ON      ON      ON
            ON      ON      ON      ON      ON      ON      ON      ON
            ON      ON      ON
```

Number of Files (# FILES)

Field	Name	Value	Description
1	NHYDF	Integer	Number of hydrodynamic files in simulation
2	NTVDF	Integer	Number of time varying data files in simulation

This card is used to specify the number of time varying input files which will be used during the simulation. When the end of a file is reached during the simulation, a new file will be opened. For example, if NTVDF is set to three, then a new file will be opened each time an end of file is reached until the third file is opened. When the end of the file is reached for the last file, the run terminates. Potential time-varying input files are HYD and CBC listed below.

Example:

```
# FILES      NHYDF   NTVDF
             2       2
```

Input/Output Filenames

Field	Name	Value	Description
1-9	"File"	Character	Input/output filename

The last cards in the control file specify the names of all input/out files used by the model. This method eliminates the need to know how to link FORTRAN files under a particular operating system. However, it does not preclude linking I/O files. If the user does decide to link files, they must be linked with the names specified by the user on the following cards. Under UNIX and VAX operating systems, full pathnames can be included in the filename specification. The names listed below are recommended from experience with the model. It is better to keep the names the same for all applications and put the I/O files in separate subdirectories with descriptive names. The number of hydrodynamic files listed under HYDFN must correspond to the number specified for NHYDF. The same holds true for the other time varying files. There must be the exact number specified by NTVDF.

Example:

```
MAP FILE.............................MAPFN.............................
        map.npt

GEO FILE............................GEOFN.............................
        geo.npt

ICI FILE............................ICIFN.............................
        ici.npt

HYD FILE............................HYDFN.............................
        hyd.npt

CBC FILE............................CBCFN.............................
        cbc.npt

DIA FILE............................DIAFN.............................
        dia.opt
```

The Map File

The MAP file specifies the linkage between faces and boxes, the number of vertical faces for each column in the water quality model, the vertical face numbers in each column, and the faces at the open-ocean boundary. Because of the nature of “box” models and the transport solution scheme employed in the model, a large amount of information must be specified in the MAP file.

In setting up the MAP file, the user must assign a face number to each face in a box. The only restriction in assigning face numbers is that all faces in the X and Y directions must be assigned a number before the vertical faces are assigned a number. Other than this restriction, there is no requirement that face numbers be assigned in any order.

Once the face numbers are assigned to the boxes, the user must specify the direction in which the faces are oriented and the surrounding box numbers. Faces in the X direction are assigned a flow direction of 1, faces in the Y direction are assigned a flow direction of 2, and faces in the Z direction are assigned a flow direction of 3.

The QUICKEST advection scheme uses a three-point interpolation scheme to estimate concentrations at a face. This requires that the solution scheme know the concentration in the two upstream boxes for a positive flow and the two downstream boxes for a negative flow. Thus for each face, four box numbers are required. They are specified in relation to positive flow. The variables ILB, IB, JB, and JRB designate which is the next upstream box number, the box immediately upstream, the box immediately downstream, and the next downstream box. If one of these boxes for a given face lies outside the grid, the box number is assigned a value of zero. Faces that exist on an inflow or outflow boundary are defined, but the IB and ILB for positive inflow and the JB and JRB for positive outflow are assigned a value of zero.

The user is also required to specify the number of vertical faces in each column even if the model is in depth integrated mode. In addition, the user must specify the bottom box number of each column and the vertical face numbers in the column starting from the bottom face.

Finally, the user must specify the face numbers of cells located along open-ocean boundaries. This specification is required for the mass balance boundary condition.

The first six lines of the MAP file are ignored and can be used to describe the file. The input formats are (6I8) for the faces and box associations, (11X,8I8) for the number of vertical faces in each column, and (9I8) for the vertical face numbers.

The following example is a 30-cell, two-dimensional grid. The grid contains 10 cells in the longitudinal direction, one cell in the lateral dimension, and three cells in the vertical. The grid contains one inflow boundary (Face 1) and three open-ocean boundaries (Faces 11, 21, 31).

```
Chesapeake Bay mock up for teaching purposes.
SET UP August 10,1992
dummy title
dummy title
dummy title
dummy title
```

F	QD	IL	IQ	JQ	JR
1	1	0	0	1	2
2	1	0	1	2	3
3	1	1	2	3	4
4	1	2	3	4	5
5	1	3	4	5	6
6	1	4	5	6	7
7	1	5	6	7	8
8	1	6	7	8	9
9	1	7	8	9	10
10	1	8	9	10	0
11	1	9	10	0	0
12	1	0	11	12	13
13	1	11	12	13	14
14	1	12	13	14	15
15	1	13	14	15	16
16	1	14	15	16	17
17	1	15	16	17	18
18	1	16	17	18	19
19	1	17	18	19	10
20	1	18	19	20	0
21	1	19	20	0	0
22	1	0	21	22	23
23	1	21	22	23	24
24	1	22	23	24	25
25	1	23	24	25	26
26	1	24	25	26	27
27	1	25	26	27	28
28	1	26	27	28	29
29	1	27	28	29	30
30	1	28	29	30	0
31	1	29	30	0	0
32	3	0	21	11	1
33	3	21	11	1	0
34	3	0	22	12	2
35	3	22	12	2	0
36	3	0	23	13	3
37	3	23	13	3	0
38	3	0	24	14	4
39	3	24	14	4	0
40	3	0	25	15	5
41	3	25	15	5	0
42	3	0	26	16	6
43	3	26	16	6	0
44	3	0	27	17	7
45	3	27	17	7	0
46	3	0	28	18	8
47	3	28	18	8	0
48	3	0	29	19	9
49	3	29	19	9	0
50	3	0	30	20	10
51	3	30	20	10	0

COLUMNS		NVF	NVF	NVF	NVF	NVF	NVF	NVF
1-9		2	2	2	2	2	2	2
BOX	BBX	VFN	VFN	VFN	VFN	VFN	VFN	VFN
21	32	33						
22	34	35						
23	36	37						

```
24      38      39
25      40      41
26      42      43
27      44      45
28      46      47
29      48      49
30      50      51

OCEAN BOUND  OBP     OBP     OBP     OBP     OBP     OBP     OBP
             11      21      31
```

The Geometry File

Cell geometry is invariant. The geometry file specifies only the cell number immediately above each cell and the surface and bottom cell numbers in each column of cells. The remaining geometric information is input in the hydrodynamics file. This format allows cell geometry to vary due to changes in surface elevation.

Two title cards and two header cards are available for descriptive purposes but are ignored by the program. The format to reach the upper cell is (8X,I8). The first field lists each cell in the grid in sequential order but is ignored by the computer. The format to read surface and bottom cells in each column of cells in (2I8).

```
Mock up Chesapeake Bay for teaching purposes.
BINARY Hydrodynamics.    March 17, 1994.

        B               BUP
        1               0
        2               0
        3               0
        4               0
        5               0
        6               0
        7               0
        8               0
        9               0
        10              0
        11              1
        12              2
        13              3
        14              4
        15              5
        16              6
        17              7
        18              8
        19              9
        20              10
        21              11
        22              12
        23              13
        24              14
        25              15
        26              16
        27              17
        28              18
        29              19
        30              20
```

sbox	bbox
1	21
2	22
3	23
4	24
5	25
6	26
7	27
8	28
9	29
10	30

The Initial Conditions Files

The Initial Conditions File has three alternate formats. The appropriate format depends on the initial conditions option specified in the Control File. The options are “UNIFORM”, “VARIED”, and “BINARY”. The UNIFORM option supplies uniform concentrations to each cell in the water column and sediments. (Modifications to the uniform conditions may be specified.) The VARIED option requires specification, in ASCII, of initial concentration in every water column and sediment cell. The BINARY option requires specification, in binary, of initial concentration in every water column and sediment cell. In a typical application of the model, uniform conditions are specified for the first model run.

The first example provided is for specification of uniform initial conditions for the 30-cell grid. The input format for all cards is (8X,9I8 or 9F8.0). Three title cards are available for identifying the file.

Example:

```
Initial conditions for thirty segment bay mockup
Includes modifications for initial sediment conditions
August 15, 1992
```

Initial Concentrations (INIT CONC)

Field	Name	Value	Description
1-22	CIC	Real	Initial concentrations

This card specifies initial water column concentrations for each of the water column state variables.

Example:

```
INIT CONC   CIC     CIC     CIC     CIC     CIC     CIC     CIC
            5.0     25.0    0.0     0.0     0.1     0.1     1.0
            1.0     1.0     1.0     1.0     1.0     1.0     1.0
            1.0     1.0     8.0     1.0.    1.0
```

Attachment 2

Additional Feature Development for the POM-IPXMT Linkage for the Lake Michigan Mass Balance Project – Task 1 Report

by

Raymond S. Chapman, Ph.D.
Ray Chapman and Associates
1725 Mac Arthur Place
Vicksburg, Mississippi 39180

Prepared for

U.S. Army Corps of Engineers
U.S. Army Waterways Experiment Station
Environmental Laboratory
3909 Halls Ferry Road
Vicksburg, Mississippi 39180

July 31, 2000

Introduction

The model design for the Lake Michigan Mass Balance Project (LMMBP) is based on a linked model approach, which in part includes hydrodynamics, sediments, transport, eutrophication, and contaminants. It is the desire of the United States Environmental Protection Agency (USEPA), Large Lakes Research Station (LLRS) that the hydrodynamic model (POM) and water quality model (IPX) linkage task follows the work performed by Water Quality and Contaminants Modeling Branch (WQCMB), Environmental Laboratory (EL) on the Chesapeake Bay Eutrophication Study. As of the present, the following tasks have been successfully completed (Chapman *et al*., 1997). 1. A processor subroutine has been implemented in the POM hydrodynamic model (HM) to provide input geometry, flow, and diffusion rate for the IPX water quality model (WQM). 2. The ICM type transport scheme has been tested within the IPX framework on both simplified and prototype grids. 3. The linkage methodology has been documented and a user's guide developed.

During the course of this project, USEPA requested the implementation of additional features in the linkage programs. The objectives of the present project is to: (1) implement ULTIMATE

QUICKEST (UQ) as the explicit portion of the water column vertical transport routine and include species variable settling; 2) assist USEPA in the development and implementation of flow and constituent tributary boundaries within POM and IPXMT; and 3) develop MAP and GEO files for the sediment layer and include logic within the transport routine to limit the sediment layer to vertical settling using UQ.

ULTIMATE QUICKEST and Vertical Settling Implementation

The following modifications were required in subroutine WTRANS and WINIT and the related include file WTRANSPT to implement ULTIMATE QUICKEST (UQ) and vertical settling. These modifications are described in order of appearance within each routine and are noted *via* comment statements.

WTRANSPT

All variables associated with the UQ multipliers were dimensioned to NQFP, which allows for vertical UQ multipliers.

The vertical settling velocity array WSET was defined and dimensioned to the number of vertical flow faces and constituents. This will allow species and spatial variability of the settling velocities.

A theta weighting array THA was defined and dimensioned to the number of constituents. This modification allows one select vertical transport that ranges from fully explicit to fully implicit for each constituent.

The ratio and curvature UQ arrays were dimensioned to NQFP.

WSET and THA were put in common blocks.

WINIT

The DO loop 10190 was changed from the number of horizontal flow faces NHQF to all flow faces NQF to allow for the initialization of the vertical UQ multipliers.

The UPWIND default loop was changed from the number of horizontal flow faces, NHQF, to all flow faces, NQF, to allow for the vertical flow face test option.

WSET is initialized to zero for all vertical flow faces and constituents.

The theta array THA is initialized to the constant theta input value TH.

WTRANS

Within the autostepping option loop, a variable NLOOP has been defined that allows for a vertical Courant Number check if explicit vertical UQ is used. When vertical, the vertical explicit option used (TH = 0.0) NLOOP is changed from NHQF to NQF. Note that this is an example

modification in that it assumes that all constituents are run fully explicit and has been implemented for testing purposes.

A switch parameter IUPDTV has been defined which enables loop 10600. When IUPDTV = 1, all of the vertical UQ multipliers are updated every time step to account for water surface elevation changes. When IUPDTV = 0, no action is taken. Note that IUPDTV is hardwired prior to the IF test.

The computation of the advection and diffusion UQ multipliers has been separated and repeated for the vertical flow faces. This has been done so that the constituent dependent settling velocity could be incorporated into the definition of the Courant Number. As a result, these computations are performed in a loop over the active constituents, which includes an IF statement that excludes the computations when a constituent uses fully implicit transport.

The DO loop 10690, which sets boundary conditions, applies the ULTIMATE limiter, updates the mass derivative, and computes the provisional concentration CSTAR has been modified as follows: first the range of the DO loop was increased to NQF to include vertical faces. IF statements checking for fully implicit transport (THA = 1) disables the computation of vertical faces. Within the loop, flows are modified to account for settling velocities, and theta weighting is applied to advection and diffusion in the vertical. In addition, the local settling flow is set to zero for horizontal faces and computed as the product of the settling velocity and face area for vertical flow faces. When the vertical flow faces are computed in a local array shift counter (IFW) is used for the vertical settling velocities. Within the vertical implicit loop 10770 the array shift counter is again used. In addition, the variables WFALL and WFALL1 are defined as one-half of the product of WSET and the time step DLT at the top and bottom of a cell, respectively.

Finally, the right hand side vector DT of the tridiagonal solver is simply set to CSTAR.

Mass Balance Tests

Simplified and prototype grid mass conservation tests were performed using a variety of settling velocities and theta weightings. Utilizing spot dumps, line dumps, and grid uniform concentrations, mass was found to be conserved within machine and POM volume accuracy.

Implementation of IPXMT Into the USEPA LLRS Water Quality Model

Subsequent to the completion of mass conservation testing within the UQ development code, the aforementioned modifications were implemented within LLRS' F90 version of IPXMT. In order to accommodate LLRS' need for an internal time step and the specification of column dependent settling velocities within the vertical UQ, the horizontal and vertical UQ routines were segregated. In addition, the loop structure for vertical UQ was changed from a NHQF to NQF to a 1 to NSB and 1 to NVF (SB), which is the same structure used in the vertically implicit routine. Again, mass balance testing confirmed that the implementation of UQ was complete and correct.

References

Chapman, R.S., T.M. Cole, and T.K. Gerald. 1997. Development of a Hydrodynamic/Water Quality (POM-IPX) Linkage for the Lake Michigan Mass Balance Project. Final Report. U.S. Environmental Protection Agency, Office of Research and Development, National Health and Environmental Effects Research Laboratory, Mid-Continent Ecology Division-Duluth, Large Lakes Research Station, Grosse Ile, Michigan. 63 pp.

Attachment 3

Additional Feature Development for the POM-IPXMT Linkage for the Lake Michigan Mass Balance Project – Task 2 Report

by

Raymond S. Chapman, Ph.D.
Ray Chapman and Associates
1725 Mac Arthur Place
Vicksburg, Mississippi 39180

Prepared for

U.S. Army Corps of Engineers
U.S. Army Waterways Experiment Station
Environmental Laboratory
3909 Halls Ferry Road
Vicksburg, Mississippi 39180

December 28, 2000

Introduction

The model design for the Lake Michigan Mass Balance Project (LMMBP) is based on a linked model approach, which, in part, includes hydrodynamics, sediments, transport, eutrophication, and contaminants. It is the desire of the United States Environmental Protection Agency (USEPA), Large Lakes Research Station (LLRS) that the hydrodynamic model (POM) and water quality model (IPX) linkage tasks follows the work performed by Water Quality and Contaminants Modeling Branch (WQCMB), Environmental Laboratory (EL) on the Chesapeake Bay Eutrophication Study. As of the present, the following tasks have been successfully completed (Chapman *et al*., 1997). (1) A processor subroutine has been implemented in the POM hydrodynamic model (HM) to provide input geometry, flow and diffusion data for the IPX water quality model (WQM); (2) the ICM type transport scheme has been tested within the IPX framework on both simplified and prototype grids; and (3) the linkage methodology has been documented and a user's guide developed. During the course of this project, USEPA requested the implementation of additional features in the linkage programs. The objective of the present project is to: (1) implement ULTIMATE QUICKEST (UQ) as the explicit portion of the water column vertical transport routine and include species variable settling; (2) assist USEPA in the development and implementation of flow and constituent tributary boundaries within POM and

IPXMT; and (3) develop MAP and GEO files for the sediment layer and include logic within the transport routine to limit the sediment layer to vertical settling using UQ.

Boundary Flow Subroutine BCOND Implementation

A boundary flow subroutine (BCOND) provided by Dr. David Schwab of National Oceanic and Atmospheric Administration (NOAA), Great Lakes Environmental Research Laboratory (GLERL) was modified to provide the MAPPER program with the I, J, K coordinates of each boundary flow and whether the boundary flow is considered a left or right flow boundary within IPXMT. These data are written to file TRIB.DAT which provides, Boundary Flow Number, Flow Direction, KP, KF, KL and a 0 or 1 to denote a left or right flow boundary, respectively. Initial POM simulations with the boundary flow subroutine enabled suggested that it was properly implemented in that it did not create catastrophic errors during the simulation. Additional testing was planned *via* a comparison of POM and IPXMT volume balances subsequent to the modification of the MAPPER program, which defines the location of the tributary boundaries within IPXMT.

MAP and GEO File Modifications

Subroutine ICM_BMF was modified to read tributary flow locations from the TRIB.DAT file. The addition of tributary boundaries did not require a change of format within the MAP file nor modification of the IPXMT hydrodynamic input processor WQMS. The additional occurrences of boundary flow faces are processed by WQMS the same as the internal POM flow faces. No changes were required within the GEO file.

Within POM, the location of the call to the mapping generation routines was moved to the bottom of the time-marching loop. This was done because all boundary information had to be properly initialized before generating the mapping information.

Local Volume Error Correction and Mass Balance Tests

Initial volume balance comparisons between POM and IPXMT yielded local continuity errors were much greater than that experienced during previous test application without tributary inflows. After a thorough examination of the linkage between POM and IPXMT, it was decided the origin of the volume imbalance resided within POM. A close examination of the BCOND subroutine revealed a number of problems. These problems and the corrections implemented are listed below.

The computation of the depth-averaged velocity used in the internal mode scaled each layer velocity by the cell height or DZ. This resulted in internal mode velocities flows that were but a fraction of that specified for that tributary. This problem was corrected by removing the DZ scaling, however, although the correction resulted in flows that were of the right order of magnitude, the integral of the internal velocities did not match the external depth-averaged velocities.

Further investigation led to an examination of the difference in how the depth-averaged velocity in the internal and external mode are computed. The depth-averaged velocity in the external mode was computed by dividing the input tributary flow by an average depth between the active cell receiving the tributary inflow and the depth default of a non-active cell (0.5 m). The internal

mode depth-averaged velocity was computed by dividing the tributary inflow by the number of layers required to achieve 5 m. In most cases, this resulted in internal flows that did not sum to the tributary inflow. Under these conditions, volume conservation is impossible. This problem was fixed by using consistent average of half-depth scaling, which was accomplished by removing the 5 m cutoff depth. Again, volume continuity was improved, but was still questionable.

An analysis of the interaction of the volume balance computation revealed that the three time level or leapfrog scheme used velocities in the external and internal continuity calculations that were off by one external time. As a result, even when the vertical distribution of internal velocities was adjusted to match and a external time step integrated depth-averaged velocity volume conservation within the internal mode was not achieved. In other words, the vertical velocities back calculated by continuity did not satisfy the boundary condition W = 0.0 at z = -h, or the lake bottom. The influence of the size of the time lag was investigated by decreasing the external time step by a factor of three. This increased the ratio of the internal to external time step from 10:1 to 30:1, the idea being that if the lag decreased the magnitude of the local volume balance error should decrease. This notion was proven true when the decrease in the external time step significantly improved the local volume balances throughout the entire simulation. Cell specific volume imbalances decreased from around 50000.0 m^3 to less than 10.0 m^3. Global volume differences were approximately 2 m^3 out of a total grid volume of 4.81475×10^{12} m^3 or less than 5×10^{-11}%.

Subsequent to implementation of the volume error corrections, prototype grid mass conservation tests were performed using a variety of settling velocities and theta weightings. Utilizing spot dumps, line dumps, and grid uniform concentrations, mass was found to be conserved within machine and POM volume accuracy. Additional POM and IPX simulations performed with double precision yielded total mass conservation to within 13 significant digits, or within machine accuracy.

References

Chapman, R.S., T.M. Cole, and T.K. Gerald. 1997. Development of Hydrodynamic/Water Quality (POM-IPXMT) Linkage for the Lake Michigan Mass Balance Project. Final Report. U.S. Environmental Protection Agency, Office of Research and Development, National Health and Environmental Effects Research Laboratory, Mid-Continent Ecology Division-Duluth, Large Lakes Research Station, Grosse Ile, Michigan. 63 pp.

Attachment 4

Additional Feature Development for the POM-IPXMT Linkage for the Lake Michigan Mass Balance Project – Final Report

by

Raymond S. Chapman, Ph.D.
Ray Chapman and Associates
1725 Mac Arthur Place
Vicksburg, Mississippi 39180

Thomas M. Cole
U.S. Army Corps of Engineers
U.S. Army Waterways Experiment Station
Environmental Laboratory
3909 Halls Ferry Road
Vicksburg, Mississippi 39180

Terry K. Gerald
AScI, Incorporated
U.S. Army Corps of Engineers
U.S. Army Waterways Experiment Station
Environmental Laboratory
3909 Halls Ferry Road
Vicksburg, Mississippi 39180

Prepared for

U.S. Environmental Protection Agency
Office of Research and Development
National Health and Environmental Effects Research Laboratory
Mid-Continent Ecology Division-Duluth
Large Lakes Research Station
Grosse Ile, Michigan 48138

September 27, 2001

Introduction

The model design for the Lake Michigan Mass Balance Project (LMMBP) is based on a linked model approach, which, in part, includes hydrodynamics, sediments, transport, eutrophication, and contaminants. The United States Environmental Protection Agency (USEPA), Large Lakes Research Station (LLRS) desires that the hydrodynamic model (POM) and water quality model (IPX) linkage task follows the work performed by Water Quality and Contaminants Modeling Branch (WQCMB), Environmental Laboratory (EL) on the Chesapeake Bay Eutrophication Study (Cerco and Code, 1995; Dortch *et al*. 1991). As of the present, the following tasks have been successfully completed (Chapman *et al*., 1997). (1) A processor subroutine has been implemented in the POM hydrodynamic model (HM) to provide input geometry, flow and diffusion data for the IPX water quality model (WQM); (2) the ICM-type transport scheme has been tested within the IPX framework on both simplified and prototype grids; and (3) the linkage methodology has been documented and a user's guide developed. The USEPA has requested additional features within the linkage programs. The objective of the present project is to: (1) implement ULTIMATE QUICKEST (UQ) (Leonard, 1979, 1991) as the explicit portion of the water column vertical transport routine and include species variable settling; (2) implementation of flow and constituent tributary boundaries within POM and IPXMT; and (3) develop MAP and GEO files for the sediment layer and implement a vertical transport routine using UQ.

ULTIMATE QUICKEST and Vertical Settling Implementation

The following modifications were required in subroutine WTRANS and WINIT and the related include file WTRANSPT to implement ULTIMATE QUICKEST (UQ) and vertical settling. These modifications are described in order of appearance within each routine and are noted *via* comment statements.

All variables associated with the UQ multipliers were dimensioned to NQFP, which allows for vertical UQ multipliers.

The vertical settling velocity array WSET was defined and dimensioned to the number of vertical flow faces and constituents. This will allow species and spatial variability of the settling velocities.

A theta weighting array THA was defined and dimensioned to the number of constituents. This modification allows one select vertical transport that ranges from fully explicit to fully implicit for each constituent.

The ratio and curvature UQ arrays were dimensioned to NQFP.

WSET and THA were put in common blocks.

The DO loop 10190 was changed from the number of horizontal flow faces NHQF to all flow faces NQF to allow for the initialization of the vertical UQ multipliers.

The UPWIND default loop was changed from the number of horizontal flow faces, NHQF, to all flow faces, NQF, to allow for the vertical flow face test option.

WSET is initialized to zero for all vertical flow faces and constituents.

The theta array THA is initialized to the constant value.

Within the autostepping option loop, a variable NLOOP has been defined that allows for a vertical Courant number check if explicit vertical UQ is used. When vertical, the vertical explicit option used (TH = 0.0) NLOOP is changed from NHQF to NQF. Note that this is an example modification in that it assumes that all constituents are run fully explicit and has been implemented for testing purposes.

A switch parameter IUPDTV has been defined which enables loop 10600. When IUPDTV = 1 all of the vertical UQ multipliers are updated every time step to account for water surface elevation changes. When IUPDTV = 0 no action is taken. Note that IUPDTV is hardwired prior to the IF test.

The computation of the advection and diffusion UQ multipliers has been separated and repeated for the vertical flow faces. This has been done so that the constituent dependent settling velocity could be incorporated into the definition of the Courant number. As a result, these computations are performed in a loop over the active constituents, which includes an IF statement that excludes the computations when a constituent uses fully implicit transport.

The DO loop 10690, which sets boundary conditions, applies the ULTIMATE limiter, updates the mass derivative, and computes the provisional concentration CSTAR has been modified as follows: first, the range of the DO loop was increased to NQF to include vertical faces. IF statements checking for fully implicit transport (THA = 1) disables the computation of vertical faces. Within the loop, flows are modified to account for settling velocities, and theta weighting is applied to advection and diffusion in the vertical. In addition, the local settling flow is set to zero for horizontal faces and computed as the product of the settling velocity and face area for vertical flow faces. When the vertical flow faces are computed, a local array shift counter (IFW) is used for the vertical settling velocities.

Two corrections were made to the UQ transport code. The first was to include the settling velocity in the determination of a positive or negative flow direction. The second was to correct an omission in the UQ limiter logic.

Within the vertical implicit loop 10770, the array shift counter is again used. In addition, the variables WFALL and WFALL1 are defined as one-half of the product of WSET and the time step DLT at the top and bottom of a cell, respectively.

Finally, the right hand side vector DT of the tridiagonal solver is simply set to CSTAR.

Mass Balance Tests

Simplified and prototype grid mass conservation tests were performed using a variety of settling velocities and theta weightings. Utilizing spot dumps, line dumps and grid uniform concentrations, mass was found to be conserved within machine and POM volume accuracy.

Boundary Flow Subroutine BCOND Implementation

A boundary flow subroutine (BCOND) provided by Dr. David Schwab of the National Oceanic and Atmospheric Administration (NOAA), Great Lakes Environmental Research Laboratory

(GLERL), was modified to provide the MAPPER program with the I, J, K coordinates of each boundary flow and whether the boundary flow is considered a left or right flow boundary within IPXMT. These data are written to file TRIB.DAT which provides Boundary Flow Number, Flow Direction, KP, KF, KL and a 0 or 1 to denote a left or right flow boundary, respectively. Initial POM simulations with the boundary flow subroutine enabled suggested that it was properly implemented in that it did not create catastrophic errors during the simulation. Additional testing was planned *via* a comparison of POM and IPXMT volume balances subsequent to the modification of the MAPPER program, which defines the location of the tributary boundaries within IPXMT.

MAP and GEO File Modifications

Subroutine ICM_BMF was modified to read tributary flow locations from the TRIB.DAT file. The addition of tributary boundaries did not require a change of format within the MAP file nor modification of the IPXMT hydrodynamic input processor WQMS. The additional occurrences of boundary flow faces are processed by WQMS the same as the internal POM flow faces. No changes were required within the GEO file.

Within POM, the location of the cell to the mapping generation routines was moved to the bottom of the time-marching loop. This was done because all boundary information had to be properly initialized before generating the mapping information.

Local Volume Error Correction and Mass Balance Tests

Initial volume balance comparisons between POM and IPXMT yielded local continuity errors that were much greater than that experienced during previous test application without tributary inflows. After a thorough examination of the linkage between POM and IPXMT, it was decided the origin of the volume imbalance resided within POM. A close examination of the BCOND subroutine revealed a number of problems. These problems and the corrections implemented are listed below.

The computation of the depth-averaged velocity used in the internal mode scaled each layer velocity by the cell height or DZ. This resulted in internal mode velocities flows that were but a fraction of that specified for that tributary. This problem was corrected by removing the DZ scaling. However, although the correction resulted in flows that were of the right order of magnitude, the integral of the internal velocities did not match the external depth-averaged velocities.

Further investigation led to an examination of the difference in how the depth-averaged velocities in the internal and external mode are computed. The depth-averaged velocity in the external mode was computed by dividing the input tributary flow by an average depth between the active cell receiving the tributary inflow and the depth default of a non-active cell (0.5 m). The internal mode depth-average velocity was computed by dividing the tributary inflow by the number of layers required to achieve 5 m. In most cases, this resulted in internal flows that did not sum to the tributary inflow. Under these conditions, volume conservation is impossible. This problem was fixed by using consistent average or half-depth scaling, which was accomplished by removing the 5 m cutoff depth. Again, volume continuity was improved, but was still questionable.

An analysis of the interaction of the volume balance computation revealed that the three time level or leapfrog scheme used velocities in the external and internal continuity calculations that were off by one external time. As a result, even when the vertical distribution of internal velocities was adjusted to match the external time step integrated depth-averaged velocity, volume conservation within the internal mode was not achieved. In other words, the vertical velocities back calculated by continuity did not satisfy the boundary condition W = 0.0 at z = -h, or the lake bottom. The influence of the size of the time lag was investigated by decreasing the external time step by a factor of three. This increased the ratio of the internal to external time step from 10:1 to 30:1, the idea being that if the lag decreased the magnitude of the local volume balance error should decrease. This notion was proven true when the decrease in the external time step significantly improved the local volume balances throughout the entire simulation. Cell specific volume imbalances decreased from around 50000.0 m^3 to less than 10.0 m^3. Global volume differences were approximately 2 m^3 out of a total grid volume of 4.81475×10^{12} m^3 or less than 5×10^{-11}%.

Subsequent to implementation of the volume error corrections, prototype grid mass conservation tests were performed using a variety of settling velocities and theta weightings. Utilizing spot dumps, line dumps, and grid uniform concentrations mass was conserved within machine and POM volume accuracy. Additional POM and IPX simulations performed with double precision yielded total mass conservation to within 13 significant digits, or within machine accuracy.

Sediment MAP and GEO Files

The MAP and GEO files have been developed for the sediment layer through modification of the original POM MAPPING routine (Chapman *et al*., 1997). These routines create several files that contain the grid linkage data. ICM_COOR2.OUT, which is generated by subroutine ICM_GCF, contains basic cell geometry data that are used to generate the remaining linkage data. The files VGEO.INP and VMAP.INP, which are generated by subroutine ICM_BVMF, contain data that are used directly by the IPXMT sediment compartment transport schemes. In addition, ICM_BVMF generates VHYD.INP, the hydrodynamic input file. VHYD.INP is similar in form to the water column file HYD, however, it only contains time invariant values of the vertical box lengths, BL, surface areas, A, and flows, Q. Default values for the diffusion and settling are defined in the input control file, SED_CON.NPT.

A diagnostic file named VMAP.DIA is also created that contains the values of constants and parameters required by IPX. The IPX code contains PARAMETER statements that set the size of data arrays. The user should note that if the POM grid geometry is altered, VMAP.DIA provides the information required to define PARAMETER statements declared in DIMSV.INC. The sediment MAPPING routine reads a free-field control file, SCN_CON.NPT, which is shown below:

```
Number of zone
                5
Depth Cutoff (m)
                10   20   50   80   120
Number of Layers
                10   8    6    5    5    5
```

Layer Thickness (m)

0.1 0.2 0.5 0.5 0.5 0.5

The general function of this modification is to allow the user a convenient means of specifying the number of layers of a given sediment column and the grid spacing of those layers depending on the depth of the water column above. Specifically, the “Number of Zones” defines the number of depth gradations where different grid refinements are defined. The “Depth Cutoff” is the range of depth that a particular “Number of Layers and Layer Thickness” are defined. For example, the “Depth Cutoff” of 10 m results in 10 layers with a layer thickness of 0.1 m for all cells where 0.0 < Depth < 10 m. In the case shown above, the same definition applies up to 80 m < Depth < 120 m. Finally, the last entries for “Number of Layers” and “Layer Thickness” correspond to all depths greater than 120 m. These data, in conjunction with the POM depth file, are used to generate the VMAP.INP, VGEO.INP, VMAP.DIA and VHYD.INP following the procedure outlined above.

Subsequent to compiling using CL_POM.SH, the following command will automatically generate both the water column and sediment grid mapping files:

```
POM MAPPWC MAPPSC  <  POMGL.IN  >  POMGL.OUT
```

The modification to the original MAPPING argument allows individual or combined mapping of the water column grid (MAPPWC) and sediment column grid (MAPPSC). Specification of MAPPWC or MAPPSC individually will result in the generation of either water column or sediment column grid linkages, respectively. This command also generates the POM standard output file POMGL.OUT, which can be compared to MAP.DIA in order to verify the integrity of the linkage. As in the original version of the mapping option, if MAPPWC and MAPPSC are omitted from the command line, a complete linked POM hydrodynamic simulation will result.

Sediment Layer Transport Routine Modification

The philosophy adopted during the development of the sediment layer transport routine was to have separate but functionally identical transport routines for the water column and sediment layers. Consequently, the F77 version of the vertical UQ water column transport routine was used as the basis for the development of the sediment layer routine. The routines VTRANS and VINIT and the related input and include files were modified so that:

All variables associated with horizontal transport, IO, kinetics and time step computations have been removed.

All variables associated with the UQ multipliers were dimensioned to NVFP, which limits the computation to vertical UQ multipliers. The labeled common names have been altered for uniqueness.

The series of read statements within VINIT has been reduced in that unnecessary input has been removed. As a consequence, the input decks have been appropriately modified.

Subroutines HYDRO and TVDS have been removed. In addition, the boundary interpolation associated with TVDS has been removed.

The autostepping option was eliminated.

The updates to volumes, box lengths, and box depths were removed to reflect the invariant nature of the grid.

The IUPDTV option has been removed to reflect the fact that the sediment grid is invariant.

The computation of the advection and diffusion UQ multipliers has been limited to the NVFQ vertical flow faces.

The fully implicit transport option (THA = 1) disables the computation of vertical UQ.

The fully explicit option (THA = 0.0) disables the diffusion contribution within the UQ multipliers and the vertical implicit transport.

The UQ conditional test on flow direction was modified so that the settling velocity was included. A correction of an omission in the UQ limiter logic was implemented.

A conservative exchange mechanism between the water column and sediment layers has been implemented. Subroutine EXCHANGE, which assigns source contributions or losses into the bottom layer of the water column grid and top layer of the sediment column, has been incorporated into the modeling system. The source information is supplied to the water and sediment columns *via* individual labeled common statements.

The control information required by the sediment transport routine is (1) read from the modified input file, SED_CON.NPT, and (2) the water column time step, SDLT, which is passed through the call to the sediment routine. At this time, it is assumed that the sediment layer time step is identical to the horizontal transport time step determined within the water column routine. An example of SED_CON.NPT is show below.

Control file for IPX Sediment Model

GEOM DEFINE	NB	NSC	NVQF	
	9849	2318	7531	
SOLTN	SLC	TH (1 if fully implicit, 0 if explicit)		
	QUICKEST	0.55		
Dispersion	ZDFC			
	ON			
DIFF	ZDF			
	.000001			
ACT CST	ACC	ACC	ACC	ACC
	ON	OFF	OFF	OFF
CST Set Vel	CSV	CSV	CSV	CSV
	1.0e-8	2.0e-9	0.0	0.0

In addition to removing unnecessary input data, a sediment layer record that defines a default settling velocity, CSV, for each constituent has been included.

Combined Water and Sediment Column Transport Routine Testing

Mass conservation testing was performed utilizing a simplified grid to test the integrity of the fully coupled system. The simplified grid tests comprised of: (1) A uniform concentration distribution test with advection and diffusion enabled. These tests used THETA equal to 0.0, 0.55 and 1.0. In all cases, mass and the uniform distribution were conserved to machine accuracy. (2) Settling was then introduced and the tests repeated. In all cases, mass was conserved and a stable vertical concentration was achieved. (3) The final set of simplified grid tests investigated pure settling where theta equal to 0.0 or fully explicit transport. An initial uniform concentration was specified with varying values of the settling velocity. In all cases, mass was conserved and the distribution monotonically settled into the bottom sediment layer. Next, the surface layer was loaded with the remaining layers set to 0.0. This test resulted in smooth transfer of mass from the surface layer to the bottom with mass conserved to within machine accuracy.

Prototype grid mass conservation tests were performed using POM hydrodynamics and a variety of settling velocities and theta weightings. The purpose of these tests was to ensure that mass could be transferred to and from the water column and sediment layers. Specifications of known fluxes within the subroutine EXCHANGE demonstrated the grid exchange mechanism was found to be conservative within machine and volume accuracy.

References

Cerco, C.F. and T. Cole. 1995. User's Guide to the CE-QUAL-ICM Three-Dimensional Eutrophication Model. U.S. Army Corps of Engineers, U.S. Army Engineer Waterways Experiment Station, Vicksburg, Mississippi. Technical Report EL-95-15, 2,420 pp.

Chapman, R.S., T.M. Cole, and T.K. Gerald. 1997. Development of Hydrodynamic/Water Quality (POM-IPXMT) Linkage for the Lake Michigan Mass Balance Project. Final Report. U.S. Environmental Protection Agency, Office of Research and Development, National Health and Environmental Effects Research Laboratory, Mid-Continent Ecology Division-Duluth, Large Lakes Research Station, Grosse Ile, Michigan. 63 pp.

Dortch, M.S. R.S. Chapman, and S.R. Abt. 1991. Application of Three-Dimensional Residual Transport. J. Hydraul. Engin., 118(6):831-848.

Leonard, B.P. 1979. A Stable and Accurate Convective Modelling Procedure Based on Upstream Interpolation. Comp. Methods App. Mechan. Engin., 19(1):59-63.

Leonard, B.P. 1991 The Ultimate Conservative Difference Scheme Applied to Unsteady One-Dimensional Advection. Comp. Methods. Appl. Mechan. Engin., 88(1):17-74.

APPENDIX B

THE FIREAD[1] PRIMER: FORMAT INDEPENDENT READING OF PROGRAM INPUT PARAMETERS

LLRS Technical Note

by

Michael Settles
Computer Sciences Corporation
Large Lakes Research Station
9311 Groh Road
Grosse Ile, Michigan 48138

Prepared for

U.S. Environmental Protection Agency
Office of Research and Development
National Health and Environmental Effects Research Laboratory
Mid-Continent Ecology Division-Duluth
Large Lakes and Rivers Forecasting Research Branch
Large Lakes Research Station
9311 Groh Road
Grosse Ile, Michigan 48138

December 16, 1996

[1]Based upon the CERN FFREAD Utility.

Contents

Abstract 271
Overview 272
User Manual 274
The FIREAD Utility Subroutines 274
Special FIREAD Keys 275
FIREAD Input Syntax 276
Examples 278
First Sample Program 278
Input Data for First Sample Program 279
Second Sample Program 280
Input Data for Second Sample Program 282
Third Sample Program 283
Input Data for Third Sample Program 285

Abstract

The FIREAD utility for passing unstructured data to batch programs is discussed. An overview of maximally unformatted reading is provided. A user manual for the operation of FIREAD is given, including example programs that make use of FIREAD input.

Appendix B

The FIREAD Primer: Format Independent Reading of Program Input Parameters

Overview

The FIREAD utility is provided to resolve an increasingly common problem in large computer programs: that of allowing programs to read input parameter data flexibly, unambiguously, and without a specific input format. Using rigid data structures to input program data is often unavoidable. However, use of fixed data structures for the input of parameter data has disadvantages; the most serious of which is the inability of the programmer to add new input parameter data not provided for in a program's original data format. This particular problem is very noticeable in modeling of complex systems in which many degrees of freedom must be parameterized. Rigid data formats also make inline documentation – i.e., documentation of the data within the file that contains it – problematic. Lastly, the data structures common read by programming languages such as FORTRAN do not provide for abbreviated input of redundant data; i.e., each data item must be represented in the input file by its individual value. When the number of redundant input data items is large, this practice becomes tedious, both to the person responsible for initially creating the input data set and to those who must review its content.

The FIREAD utility was created to address and resolve these specific problems. Instead of associating an input program variable with its location in an input data structure, the FIREAD utility uses predefined keys or tokens that may appear arbitrarily and without any predefined order with the input data stream. The flexibility of this input method allows the programmer to pass input data to the program without resorting to the use of unnecessarily complex data structures to describe the number of input data items associated with a class, the number of classes, and super-classes present, etc.

Consider as an example a simple FORTRAN program that reads in some number of similar items:

```
      PROGRAM GET_PARAMS
*
C- Read some program parameters.
      IMPLICIT NONE
*
C-The input program data.
      INTEGER        NSTRAWBERRIES        ! Number of strawberries
      INTEGER        NCHERRIES            ! Number of cherries
      INTEGER        NAPPLES              ! Number of apples
```

```
      INTEGER          NPEACHES          ! Number of peaches
      INTEGER          NBLUEBERRIES      ! Number of blueberries
*
C- Number items being read in.
      INTEGER          NFRUIT
*
C- A fruit loop variable.  (:-)
      INTEGER          IFRUIT
*
C- The maximum number of fruits to be read.
      INTEGER          MAXFRUITS
      PARAMETER        (MAXFRUITS=10)
*
C- A dummy array containing the input data items.
      INTEGER          JDUMMY (MAXFRUITS)
*
C- Get the number of fruits to be read.
      READ(5,*)NFRUIT
*
C- Read the data.
      READ(5,*) (JDUMMY(IFRUIT), IFRUIT=1, NFRUIT)
*
C- Assign the values to the appropriate variables.
      NSTRAWBERRIES   = JDUMMY (1)
      NCHERRIES       = JDUMMY (2)
      NAPPLES         = JDUMMY (3)
      NPEACHES        = JDUMMY (4)
      NBLUEBERRIES    = JDUMMY (5)
*
C- Process the fruit...
C     CALL EAT_FRUIT (NSTRAWBERRIES,NCHERRIES,NAPPLES,NPEACHES<
C  1  NBLUEBERRIES)
*
      END
```

The input data for such a program would typically look something like this:

```
5
2
1
3
7
9
```

Beginning with the number of elements in the list and followed by the values of those input data items. If a more general input format including, for instance, classes of animals and cities is desired, this input format may be inadequate. Further, the program created to read the new data format would then be unable to read the old data format without modifying the old input data to reflect the new classes of animals and cities.

The FIREAD format-independent approach is not to rely on data structures whose future we may not be able to anticipate, but to associate input data with program variables through the use of special keys. A sample of FIREAD data for the above program might look something like the following:

```
C  Begin with some documentation....
C  Example FIREAD input deck for the program GET_PARAMS
C  Here are the input parameters and their values.
NCHERRIES       12
NSTRAWBERRIES       20
NAPPLES       3
C This marks the end of the FIREAD input deck.
END
```

Here the FIREAD input values are associated with internal program variables using the keys NCHERRIES, NSTRAWBERRIES, and NAPPLES. No individual FIREAD key is required to appear in the input data. Program variables corresponding to missing FIREAD keys are assigned default values that must be specified in the application program. Since key names are uniquely associated with program variables, the keys need not appear in the input data in any particular order. The association between the keys and the program variables is made from within the application program *via* a FORTRAN callable subroutine that will be discussed later. The flexible format of the input data allows comment fields to be placed into the input data, allowing the programmer to document the input parameters.

User Manual

The FIREAD Utility Subroutines

As designed, user code calls FIREAD routines that associate user-defined variables with character keys to be read in on the input stream of the program. To use FIREAD, user routines must make use of three interface subroutines: FIINIT, FIKEY, and FIGO, which are responsible for initialization of the FIREAD utility, definitions of FIREAD keys, and reading of FIREAD input data, respectively. Each of these subroutines is discussed below.

FIINIT initializes the FIREAD utility and must be called before either FIKEY or FIGO. This routine initializes the number of defined keys and sets up predefined keys that are used by the utility. The subroutine has no arguments and the format for its use is simply:

```
CALL FIINIT()
```

FIKEY defines FIREAD keys that associate input data with variables in a calling program. The keys are short character tokens that appear in an input data file together with the data actually being read in to the user application. FIKEY must be called after FIINIT and before FIGO. The format of the subroutine is:

```
CALL FIKEY(KEYNAME,VARIABLE,N_ELEMENTS,TYPE)
*
      CHARACTER*15      KEYNAME          ! Key name
      INTEGER           N_ELEMENTS       ! Dimension of VARIABLE
      CHARACTER*(*)     TYPE             ! Data type of VARIABLE
*
```

Additional information:

- The name of the FIREAD key being defined KEYNAME can be chosen to be the same as the FORTRAN program variable VARIABLE; however, a different name may also be selected.

- The argument N_ELEMENTS specifies the dimension of the array VARIABLE[1]. If VARIABLE is not an array – i.e., a scalar – set N_ELEMENTS equal to one.

- The data type of the scalar or array reference VARIABLE is largely arbitrary. Available data types include 'INTEGER', 'REAL', 'DOUBLE', and 'CHARACTER'. Note that the data type for double precision, eight bytes, is not 'DOUBLE PRECISION', but has been abbreviated to 'DOUBLE'.

- Character string data types are provided for using the standard FORTRAN declaration syntax; i.e., CHARACTER*[n], where [n] is the length of the string. In the absence of this [n] syntax, the data type is assumed to be that of one byte, single character data.

FIGO reads in the FIREAD keys and data values from the standard input. FIGO should be called after FIINIT and FIKEY. FIGO has no arguments and the format for its use is simply:

CALL FIGO()

Special FIREAD Keys

The FIREAD utility is prepared with predefined keys that affect the way that FIREAD reports and reads data. These keys are defined as follows:

- LIST: Causes FIREAD to echo records being read in to standard output.

- NOLIST: Causes FIREAD to stop echoing records being read to standard output.

- END: Signals that reading of FIREAD cards is to be halted. Input of additional cards may be started again by calling FIGO in the user program.

---

[1]The FIKEY subroutine does not actually associate a key with the array itself, but with its starting address. Therefore, a code such as CALL FIKEY ('KEYNAME',VARIABLE(5),10, 'INTEGER') will associate the key KEYNAME not with the first element of the array VARIABLE, but with the fifth element. Note that in this case, the array VARIABLE is assumed to have been dimensioned to include a 14th element.

- STOP: Signals that reading of FIREAD cards is to be halted. The STOP key is similar to the END key, except that input of new FIREAD cards may not begin again unless FIREAD is re-initialized by calling FIINIT.

FIREAD Input Syntax

The input format used by FIREAD is chosen to allow and support documentation, scalar data, and vector data. The syntax used for specifying comment fields is borrowed from FORTRAN language conventions. Language elements used for specifying array indices and repetition of input values are taken from the CERN utility FFREAD upon which FIREAD is based. Detailed examples of the FIREAD language elements discussed below are appended to this note.

FIREAD comments are delimited by placing an uppercase "C" in the first column of the input field. The second column must contain either ASCII space or by null. Tail remarks are also supported in FIREAD and are delimited by the exclamation point "!" symbol, which may appear in any column of the input data.

FIREAD keys or data items may appear in any column of the input field. Keys that appear on the same input line as data items must appear before those data items.

Character string input data must be delimited by single quotes if the string is identical to an existing FIREAD key and appears before any key or data item on the line, or if the string contains either an ASII space, "=", or "*" symbol. Placing single quotes around all input character strings is always a safe practice.

Input FIREAD data can be provided as simple key-value pairs as shown in the example above. However, additional syntax was added to the FIREAD utility to provide for easier input of array data.

Data items to be assigned to an array may be input by placing them after the FIREAD key in the input data file. Data values to be associated with a given FIREAD key may appear on any line either on or after the line containing the key, but before the next FIREAD key.

Use of the "=" symbol specifies the array index associated with a specific input data item using the syntax:

[i]=VALUE

where [i] represents the index of the respective array and VALUE the assigned value. For example, in the following FIREAD input data:

```
C  Assign a value to an element of an array.
KEYNAME 3=5.0
```

the result is to assign the value 5.0 to the third element of the array with which KEYNAME has been associated. The next data item for this key is assumed to correspond to the next index.

For example,

```
C  Assign two values to the middle of an array.
KEYNAME 3=5.0  6.0
```

will assign the value 5.0 to the third element of the respective array, and the value 6.0 to the fourth element of the array.

Repeating data values can be assigned to elements of arrays by using the FIREAD "*" syntax:

[n]*VALUE

For example,

KEYNAME 100*-0.4

will assign the value of -0.4 to the first 100 elements of the array associated with the key KEYNAME. The "*" syntax can be combined with the "=" syntax to assign repeating values to specific array elements. The combined syntax is:

[i]=[n]*VALUE

where [i] specifies the starting index and [n] the number of times the indicated VALUE is to be repeated. This syntax is used to assign values to the elements ([i]) through ([i]+[n]-1) of the associated array. For example,

KEYNAME 5=4*1.0 2.0

which assigns the value of 1.0 to elements 5 through 8, and 2.0 to the element 9 of the associated array.

Some care must be taken when using the "=" and "*" syntax since the array elements implied in the input list can fall outside of the defined array dimensions. Given the key definition:

CALL FIKEY('KEYNAME',VARIABLE,N_ELEMENTS,'INTEGER')

FIREAD will check that array elements specified in the input data do not exceed the dimension N_ELEMENTS. However, FIREAD cannot check that the value of N_ELEMENTS corresponds to the actual array dimension. If the dimension of the array VARIABLE is less than N_ELEMENTS, FIREAD may assign input data values outside of the designated array bounds and overwrite memory. For example,

```
      PROGRAM BAD_BOUNDS
*
      IMPLICIT NONE
*
      INTEGER FLAGS (10)
```

```
*
C- Initialize FIREAD.
      CALL FIINIT()
*
C- Define a key.
      CALL FIKEY('FLAGS',FLAGS,11,'INTEGER')
*
C- Read in the FIREAD data.
      Call FIGO()
*
END
```

would allow FIREAD to assign data values outside of the memory reserved for the variable FLAGS.

Examples

First Sample Program

```
      PROGRAM FIREX1
*
C-----------------------------------------------------------------------------------------------------------------------
C-
C-   $Id;  firex1.F,v 1.2.1.1  1966/08/13  19:56:56  model  Exp $
C-   $Locker:  $
C-
C-   Purpose and Methods:    FIREAD example program 1.
                             Input data for this program is appended to this file and cpp'd out.
C-
C-   Inputs      :    None
C-   Outputs     :    None
C-   Controls    :    None
C-
C-   Calls:           FIINIT,FIKEY,FIGO
C-
C-   Created:         08/09/96  M. Settles
C-
C-   Revised:
C-
C-----------------------------------------------------------------------------------------------------------------------
*
      IMPLICIT NONE
*
C-   Some flags to be used with FIREAD.
      INTEGER  FLAG1
      REAL  FLAG2
      CHARACTER*10  FILENAME
*
C-   Need to put these in common blocks for FIREAD to work.
      COMMON / FIBLK / FLAG1,FLAG2,FILENAME
```

```
*
C-   A loop variable for printing.
          INTEGER  I
*
C-   Initialize FIREAD.
          CALL FIINIT()
*
C-   Define some keys.
          CALL FIKEY('FLAG1',FLAG1,1,'INTEGER')
          CALL FIKEY('FLAG2',FLAG2,1,'REAL')
          CALL FIKEY('FILENAME',FILENAME,1,'CHARACTER')
*
C-   Read in the FIREAD data.
          CALL FIGO()
*
C-   Show what was assigned to memory by FIREAD.
          PRINT  *
          PRINT  *,'FLAG1  =  ',FLAG1
          PRINT  *,'FLAG2  =  ',FLAG2
          PRINT  *,'FILENAME  =  ',FILENAME
          PRINT  *
*
END
```

Input Data for First Sample Program

```
C   This is a comment line.
    C   It can start after the first column,
    C   but cannot appear after a key or data item.
    C   The letter-C must be followed by a space:  "C "

C   Empty lines are ignored.
C   The 'LIST' card tells FIREAD to echo the input
C   data to standard output.
LIST

C   This is another user-defined key, this time
C   associated with a REAL data value.
C   This input line also has a "tail remark" on
C   the line.  Use tail remarks to place comments
C   on the same line, after keys and/or data values.
C   Notice that the key name or data item need not appear
C   in any particular column, so long as the key shows up
C   before the data item to be associated with it.
          FLAG2      1.2            ! This is a tail remark.

C   Here is an example of a user data card.
C   The first word 'FLAG1' denotes the name
```

```
C   of a FIREAD "key" previously defined by
C   a call to FIKEY().  This key is associated with
C   type INTEGER data.
C   Notice that the key FLAG1 appears after FLAG2
C   even though FLAG1 was declared first by a call
C   to FIKEY().  Rule:  FIREAD keys can appear in any order
C   in an input deck, so long as they are declared via
C   a call to FIKEY() before reading the input data with
C   a call to FIGO().  If a key appears more than once,
C   FIREAD will issue a warning and abort the program.
FLAG1  11

C   This next card tells FIREAD to stop echoing the input
C   data to standard output.
NOLIST

C   Here is a user-defined key associated with
C   CHARACTER string data.  Three things to notice
C   about this FIREAD card.  First, the data value associated
C   with the key does not need to be on the same input line
C   as the key.  Second, the "1=" notation means that FIREAD
C   will place the associated character data in what corresponds
C   to the first element of the variable's array.  (This key's
C   variable is a scalar, so that notation does not do much here.)
C   Third, cannot tell how much memory has been reserved for
C   the character data being provided on the input line unless it is
C   explicitly told.  Memory may be overwritten if the user provides
C   a character string that is too long for the variable intended to
C   hold it.
FILENAME
                    1,'ipx-a.dat'

C   The 'STOP' card tells FIREAD that there is no
C   more cards to read.  To begin reading cards again,
C   the user must first call FIINIT() to re-initialize
C   the FIREAD utility.
STOP
```

Second Sample Program

```
      PROGRAM FIREX2
*
C-----------------------------------------------------------------------------------------------------------------------
C-
C-     $Id:  firex2.F, v 1.2.1.1  1996/08/13  20:01:27  model  Exp  $
C-     $Locker: $
C-
```

```
C-     Purpose and Methods:  FIREAD example program 2.
C-                           Input data for this program is appended to this file and cpp'd out.
C-
C-     Inputs    :    None
C-     Outputs   :    None
C-     Controls  :    None
C-
C-     Calles:        FIINIT,FIKEY,FIGO
C-
C-     Created:       08/09/96  M. Settles
C-
C-     Revised:
C-
C-----------------------------------------------------------------------------------------------------------------------
*
       IMPLICIT  NONE
*
C-     Some flags to be used with FIREAD.
            INTEGER  FLAG1
            REAL  FLAG2
            CHARACTER*10  FILENAME
            CHARACTER*10  NAMES
            CHARACTER*15  NAMEVAR
*
C-     Need to put these in common blocks for FIREAD to work.
            COMMON / FIBLK / FLAG1,FLAG2(5),FLAG3(10),FILENAME,NAMES(5),
            NAMEVAR(10)
*
C-     A loop variable for printing.
            INTEGER  I
*
*-----------------------------------------------------------------------------------------------------------------------
*
C-     Initialize FIREAD.
            CALL FIINIT()
*
C-     Define some keys.
            CALL FIKEY('FLAG1',FLAG1,1,'INTEGER')
            CALL FIKEY('FLAG2',FLAG2,5,'REAL')
            CALL FIKEY('FLAG3',FLAG3(1),10,'INTEGER')
*
C-     The data type is just 'CHARACTER'; i.e., the character
C-     length associated with this variable is not given.  This
C-     is ok, so long as the data item does not represent an array.
            CALL FIKEY('FILENAME',FILENAME,1,'CHARACTER')
            CALL FIKEY('NAMES',NAMES(1),5,'CHARACTER*10')
            CALL FIKEY('NAMEKEY',NAMEVAR(1),10,'CHARACTER*15')
*
C-     Read in the FIREAD data.
            CALL FIGO()
```

```
*
C-     Show what was assigned to memory by FIREAD.
          PRINT   *
          PRINT   *,'FLAG1       =   ',FLAG1
          PRINT   *,'FLAG2       =   ',(FLAG2(I),I=1,5)
          PRINT   *,'FLAG3       =   ',(FLAG3(I),I=1,10)
          PRINT   *,FILENAME     =   ',FILENAME
          PRINT   *,'NAMES       =   ',(NAMES(I),I=1,5)
          PRINT   *,'NAMEKEY     =   ',(NAMEVAR(I),I-1,10)
          PRINT   *
*
END
```

Input Data for Second Sample Program

```
C   This is a comment line.
    C     It can start after the first column,
    C     but cannot appear after a key or data item.
    C     The letter-C must be followed by a space:  "C "

C   Empty lines are ignored.
C   The 'LIST' card tells FIREAD to echo the input
C   data to standard output.
LIST

C   Here is an example of a user data card.
C   The first word 'FLAG1' denotes the name
C   of an FIREAD "key" previously defined by
C   a call to FIKEY().
C   This key is associated with type INTEGER data.
FLAG1  11

C   This example shows how to assign items in an array.
C   "3=2.3" means that the next index is 3 and that the
C   associated data item for this element of the array
C   is 2.3.  There must be no spaces between the "="
C   assignment operator and neither the provided index
C   nor the data item.
FLAG2  2.1  3=2.3     !   This is a tail remark.
          2.4  2.5    !   This line dos not start with a key name,
                      !   so append the data to the variable specified
                      !   by the key on the previous line.

C   This line assigns several values to an integer array.
C   The "3*" notation means repeat the data value at the
C   implied or specified array index three (3) times.
C   There must be no spaces between the repetition operator "*"
C   and neither the repetition factor nor the data item.
C   Notice that the sixth element of the array is not assigned.
```

```
C   This element of the array will keep whatever default
C   value the user has assigned to it.
FLAG3  31  32  3*33  7=3*37  10=40
C   Here is a user-defined key associated with
C   CHARACTER string data.  Three things to notice
C   about this FIREAD card.  First, the data value associated
C   with the key does not need to be on the same input line
C   as the key.  Second, the "1=" notation means that FIREAD
C   will place the associated character data in what corresponds
C   to the first element of the variable's array.  (This key's
C   variable is a scalar, so the notation does not do much here.)
C   Third, cannot tell how much memory has been reserved for
C   the character data being provided on the input line unless it is
C   explicitly told.  Memory may be overwritten if the user provides
C   a character string that is too long for the variable intended to
C   hold it.

FILENAME
                 1='ipx-a.data'

C   This line assigns values to a character string array.
C   For this to work, the user must explicitly declare both
C   the number of elements in the array and the character size
C   of each element of the array when the key is defined by
C   the call to FIKEY.
NAMES  2*'Howdy'  'doodie'  'there'  'pardner!'

C   This assignment example only shows that the name of an
C   FIREAD key is not the same as a variable or array name
C   in a FORTRAN program.  Like the NAMES card, this card
C   also allows the user to input data for a character array;
C   however, the data goes into a variable called NAMEVAR,
C   which is different than the name of the FIREAD key.
NAMEKEY  1=3*'Big Dogs'
         4='Little Dogs'
         5=5*'Green Dogs'

END
```

Third Sample Program

```
      PROGRAM FIREX3
*
C-----------------------------------------------------------------------------------------------------------------------
C-
C-    $Id:  firex3.F,v 1.2.1.1  1996/08/13  20:06:02  model  Exp  $
C-    $Locker:  $
```

```
C-
C-     Purpose and Methods:     FIREAD example program 3.
C-                              Input data for this program is appended to this file and cpp'd out.
C-
C-     Inputs    :    None
C-     Outputs   :    None
C-     Controls  :    None
C-
C-     Calls:         FIINIT,FIKEY,FIGO
C-
C-     Created:       08/09/96  M. Settles
C-
C-     Revised:
C-
C-----------------------------------------------------------------------------------------------------------------------
*
       IMPLICIT NONE
*
C-     Some flags to be used with FIREAD.
              INTEGER  FLAG1
              REAL  FLAG2
              INTEGER  FLAG3
              CHARACTER*10  FILENAME
              CHARACTER*10  NAMES
              CHARACTER*15  NAMEVAR
*
C-     Need to put these in common blocks for FIREAD to work.
              COMMON / FIBLK / FLAG1,FLAG2(5),FLAG3(10),FILENAME,NAMES(5),
              1                          NAMEVAR(1)
*
C-     A loop variable for printing.
              INTEGER  I
*
*-----------------------------------------------------------------------------------------------------------------------
*
C-     Initialize FIREAD.
              CALL FIINIT()
*
C-     Define some keys.
              CALL FIKEY('FLAG1',FLAG1,1,'INTEGER')
              CALL FIKEY('FLAG2',FLAG2,5,'REAL')
              CALL FIKEY('FLAG3',FLAG3(1),10,'INTEGER')
*
C-     Read in the first set of FIREAD data.
              CALL FIGO()
*
C-     Show what was assigned to memory by the first FIREAD.
              PRINT   *,'FLAG1   =',FLAG1
              PRINT   *,'FLAG2   =',(FLAG2(I),I=1,5)
              PRINT   *,'FLAG3   =',(FLAG3(I),I=1,10)
```

```
*
C-     After FIGO() is called and the 'END' card has been read,
C-     the previously defined FIREAD cards become undefined.
C-     To use them again, the user must redefine them again
C-     by calling FIKEY() appropriately.
*
C-     The data type is just 'CHARACTER'; i.e., the character
C-     length associated with this variable is not given.  This
C-     is ok, so long as the data item does not represent an array.
            CALL FIKEY('NAMES',NAMES(1),5,'CHARACTER*10')
            CALL FIKEY('NAMEKEY',NAMEVAR(1),10,'CHARACTER*15')
*
C-     Read in the second set of FIREAD data.
            CALL FIGO()
*
C-     Show what was assigned to memory by the second FIREAD.
            PRINT   *,'FILENAME  =',FILENAME
            PRINT   *,'NAMES     =',(NAMES(I),I=1,5)
            PRINT   *,'NAMEKEY   =',(NAMEVAR(I),I=1,10)
*
END
```

Input Data for Third Sample Program

```
C   This is a comment line.
      C   It can start after the first column,
      C   but cannot appear after a key or data item.
      C   The letter-C must be followed by a space "C "

C   Empty lines are ignore.
C   The 'LIST' card tells FIREAD to echo the input
C   data to standard output.
LIST

C   Here is an example of a user data card.
C   The first word 'FLAG1' denotes the name
C   of a FIREAD "key" previously defined by
C   a call to FIKEY().
C   This key is associated with type INTEGER data.
FLAG1  11

C   This example shows how to assign items in an array.
C   "3=2.3" means that the next index is 3 and that the
C   associated data item for this element of the array
C   is 2.3.  There must be no spaces between the "="
C   assignment operator and the provided index
C   or the data item.
```

```
FLAG2  2.1  3=2.3   ! This is a tail remark.
                    ! This line does not start with a key name,
                    ! so append the data to the variable specified
                    ! by the key on the previous line.

C   This line assigns several values to an integer array.
C   The "3*" notation means repeat the data value at the
C   implied or specified array index three (3) times.
C   There must be no spaces between the repetition operator "*"
C   and neither the repetition factor nor the data item.
C   Notice that the sixth element of the array is not assigned
C   This element of the array will keep whatever default
C   value the user has assigned to it.
FLAG3  31  32  3*33  7=3*37  10=40

C   Let FIREAD know that this is the end of the first
C   set of cards by using the special card 'END'.  Do
C   not use the card 'STOP' unless all of the FIREAD
C   data to be interpreted by the calling program have
C   been processed.  After 'STOP' is issued, FIREAD will
C   not process any more cards unless it has been reinitialized
C   by a call to FIINIT()
END

C   Here is a user-defined key associated with
C   CHARACTER string data.  Three things to notice
C   about this FIREAD card.  First, the data values associated
C   with the key does not need to be one the same input line
C   as the key.  Second, the "1=" notation means that FIREAD
C   will place the associated character data in what corresponds
C   to the first element of the variable's array.  (This key's
C   variable is a scalar, so the notation does not do much here.)
C   Third, cannot tell how much memory has been reserved for
C   the character data being provided on the input line unless it is
C   explicitly told.  Memory may be overwritten if the user provides
C   a character string that is too long for the variable intended to
C   hold it.
FILENAME
                1='ipx-a.dat'

C   This line assigns values to a character string array.
C   For this to work, the user must explicitly declare both
C   the number of elements in the array and the character size
C   of each element of the array when the key is defined by
C   the call to FIKEY.
NAMES  2*'Howdy'  'doodie'  'there'  'pardner!'
```

```
C   This assignment example only shows that the name of an
C   FIREAD key is not the same as a variable or array name
C   in a FORTRAN program.  Like the NAMES card, this card
C   also allows the user to input data for a character array;
C   however, the data goes into a variable called NAMEVAR,
C   which is different than the name of the FIREAD key.
NAMEKEY     1=3*'Big Dogs'
            4='Little Dogs'
            5=5*'Green Dogs'
C   There are no more cards, so use of the STOP card
C   is ok here.  END will also work.
STOP
```

APPENDIX C

UT VERSION 1.3.1

LLRS Technical Note

by

Michael Settles
Computer Sciences Corporation
9311 Groh Road
Grosse Ile, Michigan 48138

Prepared for

U.S. Environmental Protection Agency
Office of Research and Development
National Health and Environmental Effects Research Laboratory
Mid-Continent Ecology Division-Duluth
Large Lakes and Rivers Forecasting Research Branch
Large Lakes Research Station
9311 Groh Road
Grosse Ile, Michigan 48138

April 1, 1998

Contents

Abstract 291
Array Manipulation Subroutines and Functions 292
Time Handling Utilities 292
Error and Message Handling: REPORT 292
Format Independent Reading: FIREAD 294

Tables

C1 FORTRAN Callable Array Manipulation Subroutines 293
C2 FORTRAN Callable Time Conversion Subroutines 293
C3 FORTRAN Callable Array Manipulation Functions 293
C4 C Callable Utility Functions 294
C5 REPORT Subroutines 294
C6 FIREAD Subroutines 294

Abstract

A number of utility routines were developed to assist with coding efforts involving several different computer models. These routines were assembled into a common object archive. Their names, functions, and the details of their uses are described.

Appendix C

UT Version 1.3.1

Array Manipulation Subroutines and Functions

Several routines were written to provide for consistent, simple array handling. These routines and functions are summarized in Tables C1 and C3. Most or perhaps all of these subroutines and functions have been superseded by FORTRAN 90 intrinsic functions and array or vector math capabilities. All of these routines were rewritten using these FORTRAN 90 language features. Users of FORTRAN 90 are encouraged to make direct use of the capabilities of that language as an alternative to these subprograms. A C callable function is provided to abbreviate the test for the presence or absence of a file. This function is briefly described in Table C4.

Time Handling Utilities

A selection of FOTRAN callable subroutines is provided. The subroutines provide for conversion between the Gregorian year/month/day, Julian day number, and Common Data Format (CDF) EPOCH representations of time. They are simple interface routines and user programs. To make use of them, the appropriate modules from Bob Stout's SNIPPETS library[1] as well as National Aeronautics and Space Administration's (NASA) CDF library and shareable object[2] must be linked to the executable file. These routines are summarized in Table C2.

Error and Message Handling: REPORT

An error and message handling utility was written. Three FORTRAN callable subroutines were written to initialize, execute, and finish error handling for user code. These routines are summarized in Table C5.

---

[1]See the site http://www.snippets.org
[2]See the site http://cdf.gsfc.nasa.gov

Table C1. FORTRAN Callable Array Manipulation Subroutines

Subroutine	Description
CVSET(N,CVAL,CARR)	Assign to a complete array
CZERO(N,.CARR)	Zero a complex array
DVSET(N,DVAL,DARR)	Assign to a double precision array
DZERO(N,DARR)	Zero a double precision array
JVSET(N,JVAL,JARR)	Assign to an integer array
JZERO(N,JARR)	Zero an integer array
SVSET(N,SVAL,SARR)	Assign to a single precision array
SZERO(N,SARR)	Zero a single precision array
ZVSET(N,ZVAL,ZARR)	Assign to a double complex array
ZZERO(N,ZARR)	Zero a double complex array

Table C2. FORTRAN Callable Time Conversion Subroutines

Subroutine	Description
EPOCH_TO_JULIAN(EP,JDN)	Convert CDF EPOCH to Julian day number
EPOCH_TO_YMD(EP,Y,MO,D,H,MI,S,MS)	Convert CDF EPOCH to Gregorian time
JULIAN_TO_EPOCH(JDN,EP)	Convert Julian day number to CDF EPOCH
JULIAN_TO_YMD(JDN,Y,MO,D)	Convert Julian day number to Gregorian date
YMD_TO_EPOCH(Y,MO,D,H,MI,S,MS,EP)	Convert Gregorian time to CDF EPOCH
YMD_TO_JULIAN(Y,MO,D,JDN)	Convert Gregorian date to Julian day number

Table C3. FORTRAN Callable Array Manipulation Functions

Function	Description
COMPLEX CVPROD(N,DN,CARR)	Product of complex array elements
COMPLEX CVSUM(N,DN,CARR)	Sum of complex array elements
LOGICAL DUNDER(DVAL)	Check for floating underflow of a double precision argument
DOUBLE PRECISION DVPROD(N,DN,DARR)	Product of double precision array elements
DOUBLE PRECISION DVSUM(N,DN,DARR)	Sum of double precision array elements
INTEGER JVPROD(N,DN,JARR)	Product of integer array elements
INTEGER JVSUM(N,DN,JARR)	Sum of vector array elements
REAL SUNDER(SVAL)	Check for floating underflow of a single precision argument
REAL SVPROD(N,DN,SARR)	Product of single precision array elements
REAL SVSUM(N,DN,SARR)	Sum of single precision array elements
INTEGER TRLEN	Length of a trimmed string
DOUBLE COMPLEX ZVPROD(N,DN,ZARR)	Product of double complex array elements
DOUBLE COMPLEX ZVSUM(N,DN,ZARR)	Sum of double complex array elements

Table C4. C Callable Utility Functions

Function	Description
int fexist(char qfile[])	Test if a file exists

Table C5. REPORT Subroutines

Subroutine	Description
REPINIT()	Initialize error handling
REPORT(ID,CSEVER,CNAME,CMESS)	Report a message
REPFIN()	Complete error handling

Format Independent Reading: FIREAD

A utility for providing input of parameter data in a format free manner is provided. The FIREAD utility routines are listed in Table C6. See Appendix B for the FIREAD user manual.

Table C6. FIREAD Subroutines

Subroutine	Description
FIINIT()	Initialize FIREAD keys
FIKEY(KEY,VAR,NELEMENTS,DTYPE)	Define a FIREAD key
FIGO()	Read FIFREAD data from standard input

APPENDIX D

AN RCS CHEAT SHEET

LLRS Technical Note

by

Michael Settles
and
Wilson Melendez
Computer Sciences Corporation
Large Lakes Research Station
9311 Groh Road
Grosse Ile, Michigan 48138

Prepared for

U.S. Environmental Protection Agency
Office of Research and Development
National Health and Environmental Effects Research Laboratory
Mid-Continent Ecology Division-Duluth
Large Lakes and Rivers Forecasting Research Branch
Large Lakes Research Station
9311 Groh Road
Grosse Ile, Michigan 48138

June 25, 2007

Contents

Introduction 297
Unix/Linux Directories 297
LLRS Implementation: File Names 298
The Cheat Sheet 298
    rlog Commands 298
    co Commands 299
    ci Commands 299
    rcsdiff Commands 299
An Example User Session 300
rcs Command 300
Conclusion 301

Appendix D

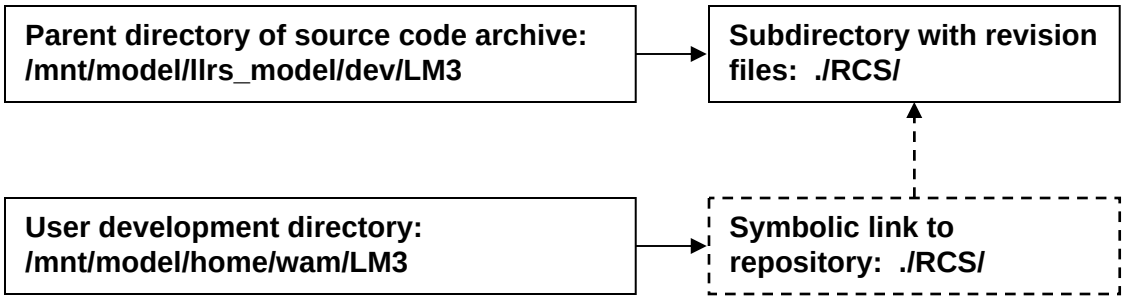
An RCS Cheat Sheet

Introduction

The following is a brief overview of the Unix/Linux commands needed for using the Revision Control System or RCS code management. Several example commands are provided below. However, this document is not intended as a comprehensive manual of RCS. More detailed discussions of RCS and source code revision control in general can be found elsewhere.[1,2]

Unix/Linux Directories

By convention, each application - i.e., each model or utility program - has a directory associated with it. The relationship between the user's development directory and the RCS repository for the respective application is shown in the figure below. The Unix/Linux command for creating a symbolic link to the repository directory is shown in the figure. The user has the username "wam" in the figure.



ln -s/mnt/model/llrs_model/dev/LM3/RCS

---

[1]Unix in a Nutshell: A Desktop Quick Reference for SVR4 and Solaris 7 - O'Reilly, 3rd Edition, August 1999.
[2]Applying RCS and SCCS - O'Reilly, 1st Edition, August 1995.

LLRS Implementation: File Names

The following conventions are being followed for application sources files.

- FORTRAN modules - i.e., subroutines and functions - are each given separate files. File names corresponding to FORTRAN modules are given the suffix .F.[1] FORTRAN INCLUDE files containing parameter definitions and common blocks are given the suffix .inc, while cpp header file names are given the suffix .h.

- All standard C function file names are given the suffix .c. C INCLUDE files are given the standard file suffix .h.

- Repository s-files - i.e., those contained in the subdirectory RCS have the additional file suffix ,v appended to their names.

The Cheat Sheet

rlog Commands

The following commands are used to show revision information.

RCS Command	Explanation
rlog RCS/*	Show detailed revision information for all source files
rlog -R RCS/*	Show all revision files
rlog RCS/filename	Show revision information for source file filename
rlog filename	Show revision information for source file filename
rlog -r1.1.1 filename	Show revision information for branch 1.1.1 of filename
rlog -R -L RCS/*	Show revision information for all files that do not have locks set
rlog -R -L -lusername RCS/*	Show files locked by the user username
rlog -d01-June-1996\<31-December-1996 filename	Show all revisions made to filename between 1-June-1996 and 31-December-1996. Note the backlash

[1]Earlier revisions of FDCHAIN, IPX, and UT were given the file suffix .FORTRAN. Please refer to http://idl340.grl.epa.gov:1024/modeling/dev.html for a discussion of the development of these applications.

co Commands

The following commands are used to check out or retrieve from the RCS repository.

RCS Command	Explanation
co filename	Check out source file filename from the default branch
co -q filename	Check out source file filename quietly (no diagnostics) from the default revision branch
co -r1.1.1 filename	Check out the latest revision of filename from branch 1.1.1
co -l1.1.1 filename	Check out and lock the latest revision of filename from branch 1.1.1
co -u1.1.1 filename	Check out and unlock the latest revision of filename from branch 1.1.1. Note: You must already have a lock on the corresponding revision

ci Commands

The following commands are used to add files to the RCS repository. All of these commands assume that the revision corresponding to the modified file has already been locked by the user.

RCS Command	Explanation
ci filename	Check in source file filename
ci -q filename	Check in source file filename quietly (no diagnostics)
ci -r1.1.1 filename	Check in source file to the revision branch 1.1.1. This is usually not necessary
ci -f filename	Force the check in of a source file filename. Check in is not normally done if no modifications were made
ci -l filename	Check in the source file filename, then check out and lock again
ci -u filename	Check in the source file filename, then check out (unlocked) again
ci -u1.1.1 filename	Check in the source file filename to the revision branch 1.1.1. Note: You must already have a lock on the corresponding revision

rcsdiff Commands

Because the RCS s-files contain both the source and its revision information, direct comparison of a revised source file with a RCS s-file is impractical. The RCS command rcsdiff allows the user to compare a checked out version of a file with any previous revision of that file, for the purpose of identifying recent modifications to the source.

RCS Command	Explanation
rcsdiff filename	Show differences between the user file filename and its most recent revision in the default branch
rcsdiff -r1.1.1 filename	Show differences between the user file filename and the latest revision in the branch 1.1.1
rcsdiff -r1.1.1.3 filename	Show differences between the user file filename and the specific revision 1.1.1.3
rcsdiff -r1.1.1.3 -r1.1.1.4 filename	Show differences between revisions 1.1.1.3 and 1.1.1.4 of the source file filename

An Example User Session

In the example below, the user makes modifications to a source file for the LM3 application. The symbolic link to the LM3 RCS repository only needs to be made if it is not already present. The user reviews the revision information for a specific file and checks out two files. The user intends to modify the files being checked out and locks them at check out time. Only one file is modified, but the user desires that both files be checked in as new revisions. The modified file can be checked in normally; however, the unmodified file must be checked in using the -f flag, otherwise, no new revision will be registered by the source code management. The user then reviews the revision information for the two files.

```
% cd                                  # Go to home directory.
% cd LM3                              # Go to LM3 development directory.
% ln -s ~model/dev/LM3/RCS            # Make symbolic link to repository.
% rlog diatoms.F                      # See revisions for a file.
% co -l3.2.13 diatoms.F               # Check out and lock first file.
% co -l3.2.13 greens.F                # Check out and lock second file.
% emacs diatoms.F &                   # Edit the file.
% rcsdiff diatoms.F                   # Compare with last checked in revision.
% ci diatoms.F                        # Check in first file.
% ci -f greens.F                      # Force check in of second file.
% rlog diatoms.F greens.F             # See revision information of both files.
```

rcs Commands

The RCS command rcs is used by the source code administrator to create s-files, assign access privileges for modification of source files, set default revision branches, and, in general, performs those tasks responsible for the creation and organization of source code repositories. Use of this command is restricted to the source code administrator.

RCS Command	Explanation
rcs -i filename	Create a s-file for filename
rcs -i -awam filename	Create a s-file for filename and give user wam write access to it
rcs -bversion filename	Set the default branch for source file filename to version. Versions checked in and out for this file will use this version, unless explicitly specified otherwise
rcs - lversion filename	Lock revision branch version. No further revisions to this branch will be allowed
rcs -uversion filename	Unlock revision branch version. Breaking the lock allows new revisions to be made within this branch
rcs -mversion:comment	Change the comment for version to comment
rcs -sstate:version	Change the development version to a given state. The state designates the stability of the particular revision and is determined by the administrator. States can be Rel for released, Stab for stable, or Exp for experimental. By default, revision branches are experimental. The label is not used internally by RCS, but is only intended to help organize the component revision branches for a given application

Conclusion

A brief review of the RCS directory structure and commands has been provided. Example commands for examining revision information, retrieving and checking in source code, and examining differences between source code revisions have been provided. Review of the RCS administrator's command rcs has been appended; however, this command is not generally accessible by source developers and should only be used by the source code administrator.

APPENDIX E

DERIVATION OF THE ULTIMATE QUICKEST METHOD IN ONE DIMENSION

by

Michael Settles
Computer Sciences Corporation
Large Lakes Research Station
9311 Groh Road
Grosse Ile, Michigan 48138

Prepared for

U.S. Environmental Protection Agency
Office of Research and Development
National Health and Environmental Effects Research Laboratory
Mid-Continent Ecology Division-Duluth
Large Lakes and Rivers Forecasting Research Branch
Large Lakes Research Station
9311 Groh Road
Grosse Ile, Michigan 48138

June 30, 2000

Contents

E.1 Introduction 306
E.2 Derivation of the ULTIMATE QUICKEST Method 306
E.3 Implementation of the ULTIMATE QUICKEST Method 315
E.3.1 ULTIMATE QUICKEST Advective Multipliers 315
E.3.1.1 Solutions for Advective Multipliers for the Case u > 0 316
E.3.1.2 Solutions for Advective Multipliers for the Case u < 0 320
E.3.2 ULTIMATE QUICKEST Dispersive Multipliers 323
E.3.2.1 Solutions for Dispersive Multipliers for the Case u > 0 324
E.3.2.2 Solutions for Dispersive Multipliers for the Case u < 0 326
E.3.3 Updating the Concentration 327

Figures

E.1 Cell geometry and notation used in this derivation 310

E.2 Interpolation of concentration used for the case u > 0 317

E.3 Interpolation of concentration used for the case u < 0 320

Tables

E.1 FORTRAN Variables Used in u > 0 Calculation of Advection and Diffusion Multipliers 307

E.2 FORTRAN Variables Used in u < 0 Calculation of Advection and Diffusion Multipliers 308

E.3 FORTRAN Variables Used in Both u > 0 and u < 0 Transport Calculations 309

Appendix E

Derivation of the ULTIMATE QUICKEST Method in One Dimension

E.1 Introduction

The ULTIMATE QUICKEST (UQ) transport scheme developed by Leonard (1979) was augmented for use with variable grid sizes by Chapman (Chapman *et al*., 1997). The resulting transport algorithm was coded in FORTRAN and applied to the Chesapeake Bay (CE-QUAL-ICM) model (Cerco and Cole, 1994; Cerco and Cole, 1995). The same transport scheme was adapted for use in the Lake Michigan Mass Balance Project (LMMBP) and the IPX pollution calculation (Velleux *et al*., 2001; Chapman *et al*., 1997).

The box model calculation performs numerical integration of spatially varying concentrations using quadratic interpolations of the concentration to infer its values at flow faces and analytic integration over the space- and time-variables to account for changes in the concentration at the cell wall during the course of the time step. This appendix provides a step-by-step derivation of the UQ method in one dimension (horizontal) and its implementation in the LM3 water quality model transport calculations. This work was previously performed by Leonard (1979) and by Chapman (Chapman *et al*., 1997).

Associations between key FORTRAN variables and the symbols used in the derivation are summarized in Tables E.1 through E.3.

E.2 Derivation of the ULTIMATE QUICKEST Method

Derivation of the UQ transport method is performed here in one dimension[1] and assumes the advection-diffusion equation

$$\partial_t \phi = -\partial_x \left(u\phi \right) + \partial_x \left(\Gamma \partial_x \phi \right) \tag{E.1}$$

[1]A more precise interpretation is that the transport was performed through one dimension of a two- or three-dimensional system. Use of the one-dimensional transport equation was assumed to be approximately correct throughout this derivation. A two-dimensional UQ calculation had been done by Hall and Chapman (1985).

Table E.1. FORTRAN Variables Used in u > 0 Calculation of Advection and Diffusion Multipliers

FORTRAN Variable	Expression
DEN1$^+$	$\frac{1}{4}(\Delta x_{i-1} + 2\Delta x_i + \Delta x_{i+1})(\Delta x_{i-1} + \Delta x_i)$
DEN2$^+$	$-\frac{1}{4}(\Delta x_{i-1} + \Delta x_i)(\Delta x_i + \Delta x_{i+1})$
DEN3$^+$	$\frac{1}{4}(\Delta x_i + \Delta x_{i+1})(\Delta x_{i-1} + 2\Delta x_i + \Delta x_{i+1})$
GRAD1$^+$	G^+_{i-1}
GRAD2$^+$	G^+_i
GRAD3$^+$	G^+_{i+1}
TERM1$^+$	T^+_{i-1}
TERM2$^+$	T^+_i
TERM3$^+$	T^+_{i+1}
T2$^+$	$\frac{\Delta x_{i+1}}{\Delta x_i + \Delta x_{i+1}}$
T3$^+$	$\frac{\Delta x_i}{\Delta x_i + \Delta x_{i+1}}$
TP1$^+$	$\frac{\delta x}{2}(\Delta x_i - \Delta x_{i+1})$
TP2$^+$	$\frac{\delta x}{2}(\Delta x_{i-1} + 2\Delta x_i - \Delta x_{i+1})$
TP3$^+$	$\frac{\delta x}{2}(\Delta x_{i-1} + 3\Delta x_i)$

Table E.2. FORTRAN Variables Used in u < 0 Calculation of Advection and Diffusion Multipliers

FORTRAN Variable	Expression
DEN1$^-$	$\frac{1}{4}(\Delta x_i + 2\Delta x_{i+1} + \Delta x_{i+2})(\Delta x_i + \Delta x_{i+1})$
DEN2$^-$	$-\frac{1}{4}(\Delta x_{i+1} + \Delta x_{i+2})(\Delta x_i + \Delta x_{i+1})$
DEN3$^-$	$\frac{1}{4}(\Delta x_{i+1} + \Delta x_{i+2})(\Delta x_i + 2\Delta x_{i+1} + \Delta x_{i+2})$
GRAD1$^-$	G_i^-
GRAD2$^-$	G_{i+1}^-
GRAD3$^-$	G_{i+2}^-
TERM1$^-$	T_i^-
TERM2$^-$	T_{i+1}^-
TERM3$^-$	T_{i+2}^-
T1$^-$	$\frac{\Delta x_{i+1}}{\Delta x_i + \Delta x_{i+1}}$
T2$^-$	$\frac{\Delta x_i}{\Delta x_i + \Delta x_{i+1}}$
TP1$^-$	$-\frac{\delta x}{2}(3\Delta x_{i+1} + \Delta x_{i+2})$
TP2$^-$	$\frac{\delta x}{2}(\Delta x_i - 2\Delta x_{i+1} - \Delta x_{i+2})$
TP3$^-$	$\frac{\delta x}{2}(\Delta x_i - \Delta x_{i+1})$

Table E.3. FORTRAN Variables Used in Both u > 0 and u < 0 Transport Calculations

FORTRAN Variable	Expression
A	$\mathbf{A}$
ALFA	$2\left[\frac{\Gamma\Delta t}{\delta x^2}-\frac{1}{6}(1-c^2)\right]$
COUR	$\mathbf{c}$
DIFF	$\mathbf{\Gamma}$
DLT	$\mathbf{\Delta t}$
DTM	$\partial_t\mathbf{m}$
SF1	$\mathbf{\delta x}$
SF2	$\mathbf{\Delta x_i^2}$
V1	$\mathbf{v_i^n}$
V2	$\mathbf{v_i^{n+1}}$

where ϕ is some concentration, u is the velocity and Γ is the dispersion coefficient. Source-sink terms are assumed to be negligible. The cell geometry and notation used in this derivation are shown in Figure E.1. Integrating (E.1) over time and the volume of the ith cell gives:

$$\int_{-\Delta x_i/2}^{+\Delta x_i/2} d\xi \int_{t^n}^{t^{n+1}} dt\,\partial_t\phi = \int_{-\Delta x_i/2}^{+\Delta x_i/2} d\xi \int_{t^n}^{t^{n+1}} dt\left[-\partial_\xi\left(u\phi\right)+\partial_\xi\left(\Gamma\partial_\xi\phi\right)\right] \tag{E.2}$$

where $\Delta x_i = x_{i+1} - x_i$ is the cell size, $\xi = x - x_i$ is the distance from the cell center, $\partial_\xi = \partial_x$, and t^n is the value of time at the beginning of the nth time step. Carrying out the integration gives the result:

$$\int_{-\Delta x/2}^{+\Delta x/2} d\xi\left(\phi^{n+1}-\phi^n\right) = \int_{t^n}^{t^{n+1}} dt\left[-\left(u_r\phi_r - u_l\phi_l\right)+\Gamma_r\partial_\xi\phi_r - \Gamma_l\partial_\xi\phi_l\right] \tag{E.3}$$

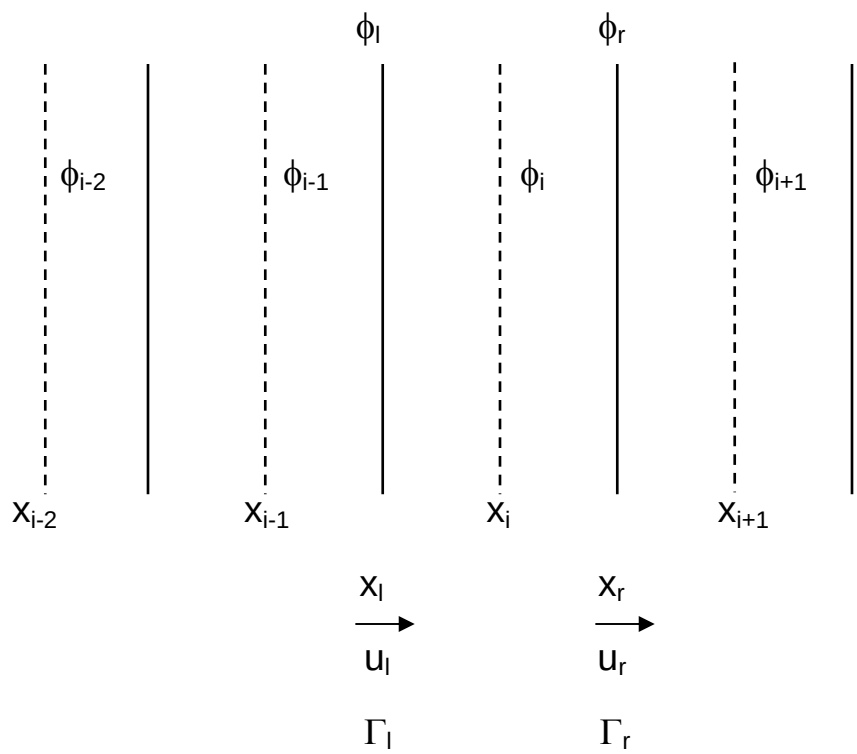


Figure E.1. Cell geometry and notation used in this derivation. ϕ_i represents constituent concentrations that have been volume-averaged within each cell. The variables ϕ_r and ϕ_l represent values of the concentration at the right and left cell walls. Space coordinates x_i represent the cell centers, and $x_{r,i}$ coordinates of the cell walls. Velocities of the water at the right and left cell faces $u_{r,l}$ are presumed to be positive to the right, as indicated by the arrows. $\Gamma_{r,l}$ are the respective dispersion coefficients.

where ϕ^n is the concentration at the nth time step, and the subscripts r, l refer to values at the right and left cell faces, respectively. Equation E.3 is broken into the contributing integrals:

$$I_1 = \int_{-\Delta x/2}^{+\Delta x/2} d\xi \left(\phi^{n+1} - \phi^n\right) \tag{E.4}$$

$$I_2 = u\int_{t^n}^{t^{n+1}} dt \left(\phi_l - \phi_r\right) \tag{E.5}$$

$$I_3 = \Gamma\int_{t^n}^{t^{n+1}} dt \left(\partial_\xi \phi_r - \partial_\xi \phi_l\right) \tag{E.6}$$

where we have assumed that the velocity u and dispersion coefficient Γ are constant over the domains of integration.

The integral I_1 is evaluated by first Taylor expanding the concentration about the point x_i, giving:

$$I_1 = \int_{-\Delta x/2}^{+\Delta x/2} d\xi \left(\phi_i^{n+1} + \xi \partial_\xi \phi_i^{n+1} \frac{1}{2} \xi^2 \partial_\xi^2 \phi_i^{n+1} + \cdots - \phi_i^n - \xi \partial_\xi \phi_i^n - \frac{1}{2} \xi^2 \partial_\xi^2 \phi_i^n - \cdots \right) \quad \text{(E.7)}$$

where the notation $\partial_\xi \phi_i$ has been used to represent the value of the derivative $\partial_\xi \phi(x)_{x=xi}$. Derivative factors are constant with respect to the integration over ξ. Terms antisymmetric in ξ integrate away, leaving:

$$I_1 \doteq \Delta x \phi_i^{n+1} + \frac{\Delta x^3}{24} \partial_\xi^2 \phi_i^{n+1} - \Delta x \phi_i^n - \frac{\Delta x^3}{24} \partial_\xi^2 \phi_i^n \quad \text{(E.8)}$$

The second and fourth terms can be combined:

$$\frac{\Delta x^3}{24} \partial_\xi^2 \left(\phi^{n+1} - \phi^n \right) \doteq \frac{\Delta x^3}{24} \partial_\xi^2 \Delta t \partial_t \phi = -\frac{\Delta x^3}{24} \Delta t \partial_\xi \partial_\xi^2 \left(u\phi \right)$$

$$= -\frac{\Delta x^3}{24} u \Delta t \frac{\partial_\xi^2 \phi_r - \partial_\xi^2 \phi_l}{\Delta x} = -\frac{\Delta x^2}{24} u \Delta t \left(CURV_r - CURV_l \right) \quad \text{(E.9)}$$

where $\Delta t = t^{n+1} - t^n$, the relation (E.1) is used to replace the time derivative, and higher order contributions in the time series expansion of ϕ and $\mathcal{O}(4)$ dispersion contributions are dropped. Throughout, the abbreviated notation $\Delta x^n \equiv (\Delta x)^n$ is used. Divided difference representations[1] of the second derivatives of the concentration at the right and left cell walls are:

$$CURV_r \equiv \frac{\phi_{i-1}^n + \phi_{i+1}^n - 2\phi_i^n}{\Delta x^2} \quad \text{(E.10)}$$

$$CURV_l \equiv \frac{\phi_{i-2}^n + \phi_i^n - 2\phi_{i-1}^n}{\Delta x^2} \quad \text{(E.11)}$$

Substituting (E.9) into (E.8) gives the result

$$I_1 = \Delta x \left(\phi_i^{n+1} - \phi_i^n \right) - \frac{\Delta x^3}{24} c \left(CURV_r - CURV_l \right) \quad \text{(E.12)}$$

---

[1]Leonard's (1991) choice for representing the curvature of the concentration profile is practical and applicable but is also specific to the case u > 0. In the more general case, the CURV expressions should be related with the second space derivatives of concentration.

where c is the Courant number

$$c \equiv \frac{u\Delta t}{\Delta x} \tag{E.13}$$

The second integral (E.5) is evaluated by first Taylor expanding the concentration about time t^n

$$I_2 = u\int_{t^n}^{t^{n+1}} dt\left(\phi_l^n + \left(t - t^n\right)\partial_t\phi_l^n + \frac{1}{2}\left(t - t^n\right)^2 \partial_t^2\phi_l^n + \cdots \right.$$
$$\left. - \phi_r^n - \left(t - t^n\right)\partial_t\phi_r^n - \frac{1}{2}\left(t - t^n\right)^2 \partial_t^2\phi_r^n - \cdots\right) \tag{E.14}$$

Next use (E.1) to replace the time derivatives, giving:

$$I_2 = u\left[\Delta t\phi_l^n + \frac{1}{2}\Delta t^2\left(-u\partial_x\phi_l^n + \Gamma\partial_x^2\phi_l^n\right) + \frac{1}{6}u^2\Delta t^3\partial_x^2\phi_l^n + \cdots \right.$$
$$\left. - \Delta t\phi_r^n - \frac{1}{2}\Delta t^2\left(-u\partial_x\phi_r^n + \Gamma\partial_x^2\phi_r^n\right) - \frac{1}{6}u^2\Delta t^3\partial_x^2\phi_r^n - \cdots\right] \tag{E.15}$$

$$\doteq \Delta x \bullet c\left(\phi_l^n - \frac{1}{2}\Delta x \bullet c \bullet GRAD_l + \frac{1}{2}\Delta x^2\alpha \bullet CURV_l + \frac{1}{6}\Delta x^2c^2CURV_l \right.$$
$$\left. - \phi_r^n + \frac{1}{2}\Delta x \bullet c \bullet GRAD_r - \frac{1}{2}\Delta x^2\alpha \bullet CURV_r - \frac{1}{6}\Delta x^2c^2CURV_r\right) \tag{E.16}$$

using the dispersion number $\alpha = \Gamma\,\Delta t/\Delta x^2$, and where

$$GRAD_r \equiv \frac{\phi_{i+1}^n - \phi_i^n}{\Delta x} \tag{E.17}$$

$$GRAD_l \equiv \frac{\phi_i^n - \phi_{i-1}^n}{\Delta x} \tag{E.18}$$

The third integral (E.6) is similarly evaluated by first expanding the concentration ϕ in time about t^n.

$$I_3 = \Gamma\partial_\xi \int_{t^n}^{t^{n+1}} dt \left(\phi_r^n + \left(t - t^n\right)\partial_t\phi_r^n + \cdots - \phi_l^n - \left(t - t^n\right)\partial_t\phi_l^n - \cdots \right)$$

$$\doteq \Gamma\partial_\xi \left(\Delta t\phi_r^n + \frac{1}{2}\Delta t^2\partial_t\phi_r^n - \Delta t\phi_l^n - \frac{1}{2}\Delta t^2\partial_t\phi_l^n \right) \tag{E.19}$$

Again, use (E.1) to replace the time derivative with a space derivative. The dispersion contribution to the time derivative $\Gamma\partial_x^2$ gives terms $\mathcal{O}(3)$ in space and can be ignored, giving:

$$I_3 = \Gamma\left(\Delta t\partial_\xi\phi_r^n - \frac{u}{2}\Delta t^2\partial_\xi^2\phi_r^n - \Delta t\partial_\xi\phi_l^n + \frac{u}{2}\Delta t^2\partial_\xi^2\phi_l^n \right)$$

$$= \Delta x^2\alpha\left(GRAD_r - \frac{c}{2}\Delta x \bullet CURV_r - GRAD_l + \frac{c}{2}\Delta x \bullet CURV_l \right) \tag{E.20}$$

Substituting for the terms in (E.3) gives an expression free of integrals, relating the concentration ϕ to its derivatives.

$$\Delta x\left(\phi_i^{n+1} - \phi_i^n\right) - \frac{\Delta x^3}{24}c\left(CURV_r - CURV_l\right)$$

$$= \Delta x \bullet c\left(\phi_l^n - \frac{\Delta x}{2}c \bullet GRAD_l + \frac{\Delta x^2}{2}\alpha \bullet CURV_l + \frac{\Delta x^2}{6}c^2 CURV_l \right)$$

$$- \Delta x \bullet c\left(\phi_r^n - \frac{\Delta x}{2}c \bullet GRAD_r + \frac{\Delta x^2}{2}\alpha \bullet CURV_r + \frac{\Delta x^2}{6}c^2 CURV_r \right)$$

$$+ \Delta x^2\alpha\left(GRAD_r - \frac{\Delta x}{2}cCURV_r - GRAD_l + \frac{\Delta x}{2}cCURV_l \right) \tag{E.21}$$

The value of the concentration at the right wall ϕ, is simply derived by Taylor expanding its value about the center of the cell which is represented by ϕ_i at the coordinate x_i.

$$\phi_r = \phi_i + \frac{\Delta x}{2}\partial_x\phi_i + \frac{1}{2}\left(\frac{\Delta x}{2}\right)^2\partial_x^2\phi_i = \phi_i + \frac{\Delta x}{2}\left(\partial_x\phi_r - \frac{\Delta x}{2}\partial_x^2\phi_r\right) + \frac{1}{2}\left(\frac{\Delta x}{2}\right)^2\partial_x^2\phi_i$$

$$= \phi_i + \frac{\Delta x}{2}\left(\frac{\phi_{i+1} - \phi_i}{\Delta x} - \frac{\Delta x}{2}\partial_x^2\phi_r\right) + \frac{1}{2}\left(\frac{\Delta x}{2}\right)^2\partial_x^2\phi_i = \frac{1}{2}\left(\phi_{i+1} + \phi_i\right) - \frac{\Delta x^2}{8}CURV_r \tag{E.22}$$

The values of the curvature at the cell center $\partial_x^2\phi_i$ and right cell face $\partial_x^2\phi_r$ are equal and determined by the assumed parabolic interpolation. The similar case for the left wall concentration is derived by substituting $\phi_{i-1} \rightarrow \phi_{i-2}$, $\phi_i \rightarrow \phi_{i-1}$, and $\phi_{i+1} \rightarrow \phi_i$, giving:

$$\phi_l = \frac{1}{2}\left(\phi_{i-1} + \phi_i\right) - \frac{\Delta x^2}{8} CURV_l \tag{E.23}$$

Substituting for ϕ_r and ϕ_l and collecting terms gives expressions for the concentration at time t_{n+1} as a function of known quantities at the previous time step.

$$\begin{aligned}
\phi_i^{n+1} = \phi_i^n & \\
& + c\left[\frac{1}{2}\left(\phi_{i-1}^n + \phi_i^n\right) - \frac{\Delta x^2}{8} CURV_l - \frac{\Delta x}{2} c \bullet GRAD_l\right] \\
& + c\left[\frac{\Delta x^2}{2} \alpha \bullet CURV_l + \frac{\Delta x 2}{6} c^2 \bullet CURV_l\right] \\
& - c\left[\frac{1}{2}\left(\phi_{i+1}^n + \phi_i^n\right) - \frac{\Delta x^2}{8} CURV_r - \frac{\Delta x}{2} c \bullet GRAD_r \frac{\Delta x^2}{2} \alpha \bullet CURV_r\right] \\
& - c\left[\frac{\Delta x^2}{6} c^2 \bullet CURV_r\right] \\
& + \Delta x \bullet \alpha\left(GRAD_r - \frac{\Delta x}{2} c \bullet CURV_r - GRAD_l + \frac{\Delta x}{2} c \bullet CURV_l\right) \\
& + \frac{\Delta x^2}{24} c\left(CURV_r - CURV_l\right)
\end{aligned} \tag{E.24}$$

$$\begin{aligned}
\phi_i^{n+1} = \phi_i^n & \\
& + c\left[\frac{1}{2}\left(\phi_{i-1}^n + \phi_i^n\right) - \frac{\Delta x^2}{6}\left(1 - c^2 - 3\alpha\right) CURV_l - \frac{\Delta x}{2} c \bullet GRAD_l\right] \\
& - c\left[\frac{1}{2}\left(\phi_{i+1}^n + \phi_i^n\right) - \frac{\Delta x^2}{6}\left(1 - c^2 - 3\alpha\right) CURV_r - \frac{\Delta x}{2} c \bullet GRAD_r\right] \\
& + \Delta x \bullet \alpha\left(GRAD_r - \frac{\Delta x}{2} c \bullet CURV_r - GRAD_l + \frac{\Delta x}{2} c \bullet CURV_l\right)
\end{aligned} \tag{E.25}$$

This is the result offered by Leonard (1991).

E.3 Implementation of the ULTIMATE QUICKEST Method

Implementation of (E.25) in the ICM and more recently LM3 water quality transport code has been accomplished by Chapman (Cerco and Cole, 1994; Chapman *et al.*, 1997) and allows for variable grid spacing. Integration of the advection-diffusion differential equation (E.1) over the time step Δt gives the result

$$\phi^{n+1} = \phi^{n} - c\left(\bar{\phi}_r - \bar{\phi}_l\right) + \Delta x \bullet \alpha\left(GRAD_r - GRAD_l\right) \tag{E.26}$$

where $\bar{\phi}_r$ and $\bar{\phi}_l$ are the time averaged concentrations at the right and left cell walls with UQ corrections, accounting for quadratic interpolation of the concentration, and change in the concentration at the cell face over the course of the time step Δt.

The solution for ϕ^{n+1} is found by calculating the time averaged concentrations and gradients or, equivalently, the diffusive fluxes at the cell faces in (E.26) and integrating them over one time step. In the FORTRAN code, this final step makes use of the volume integrated version of (E.26) by associating the time rate of change of mass with flow ($\bar{\phi}_r$) and dispersion flux. The transactional symmetry of the cells (the left face of cell i is the same as the right cell face of cell i-1) is exploited so that time averaged concentrations and diffusive fluxes only need to be calculated at either the left or the right face of each cell. Following Chapman's (Chapman *et al*., 1997) convention, estimates for the right cell face are used in this derivation.

E.3.1 ULTIMATE QUICKEST Advective Multipliers

Rearranging and combining terms in the result (E.25) gives:

$$\begin{aligned}\phi_i^{n+1} = \phi_i^{n} \\ &+ c\left[\frac{1}{2}\left(\phi_{i-1}^{n} + \phi_i^{n}\right) - \frac{\Delta x}{2} c \bullet GRAD_l - \frac{\Delta x^2}{6}\left(1 - c^2 - 6\alpha\right) CURV_l\right] \\ &- c\left[\frac{1}{2}\left(\phi_{i+1}^{n} + \phi_i^{n}\right) - \frac{\Delta x}{6} c \bullet GRAD_r - \frac{\Delta x^2}{6}\left(1 - c^2 - 6\alpha\right) CURV_r\right] \\ &+ \Delta x \bullet \alpha\left(GRAD_r - GRAD_l\right)\end{aligned} \tag{E.27}$$

Using $\phi_i^{n+1} + \phi_i^{n} - c(\bar{\phi}_r + \bar{\phi}_l) + \ldots$ from (E.26), the time averaged concentration at the right cell wall is as follows:

$$\bar{\phi}_r = \frac{1}{2}\left(\phi_{i+1}^n + \phi_i^n\right) - \frac{\Delta x}{2} c \bullet GRAD_r - \frac{\Delta x^2}{6}\left(1 - c^2 - 6\alpha\right) CURV_r$$

$$= \frac{1}{2}\left(\phi_{i+1}^n + \phi_i^n\right) - \frac{u\Delta t}{2} GRAD_r + \left[\Gamma\Delta t - \frac{1}{6}\left(\Delta x^2 - u^2\Delta t^2\right)\right] CURV_r \qquad \text{(E.28)}$$

The results (E.25) and (E.28) are generally applicable to flow moving either from left to right – as has been assumed up to this point – or from right to left across a cell face if the Courant number c changes sign with u and if representations of the gradient and curvature terms are modified to reflect a concentration profile that is interpolated upstream of the cell wall.

E.3.1.1 Solutions for Advective Multipliers for the Case u > 0

Solution for the case u > 0 makes use of a quadratic interpolation of the concentration ϕ that is upstream of the cell wall x_r (Figure E.2). Next, polynomial representations of the concentration, gradient, and curvature terms such that the solution for $\bar{\phi}_r^+$ – where the superscript "+" is used to denote results specific to the case u > 0 – can be written:

$$\bar{\phi}_r^+ = P_1(x) - \frac{u\Delta t}{2} P_2'(x) + \left[\Gamma\Delta t - \frac{1}{6}\left(\Delta x^2 - u^2\Delta t^2\right)\right] P_2''(x) \qquad \text{(E.29)}$$

where $P_1(x)$ and $P_2(x)$ are first- and second-order Lagrange interpolation polynomials.

$$P_1(x) = \frac{x_{i+1} - x}{x_{i+1} - x_i}\phi_i^n + \frac{x - x_i}{x_{i+1} - x_i}\phi_{i+1}^n \qquad \text{(E.30)}$$

$$P_2(x) = \frac{\left(x - x_{i+1}\right)\left(x - x_i\right)}{\left(x_{i-1} - x_{i+1}\right)\left(x_{i-1} - x_i\right)}\phi_{i-1}^n + \frac{\left(x - x_{i+1}\right)\left(x - x_{i-1}\right)}{\left(x_i - x_{i+1}\right)\left(x_i - x_{i-1}\right)}\phi_i^n$$

$$+ \frac{\left(x - x_i\right)\left(x - x_{i-1}\right)}{\left(x_{i+1} - x_i\right)\left(x_{i+1} - x_{i-1}\right)}\phi_{i+1}^n \qquad \text{(E.31)}$$

As before, ϕ_i^n are the uncorrected concentrations averaged over the respective cell volumes at time t^n. Further, the UQ corrected concentration at the right cell face can be expressed as a weighted sum of the concentration being interpolated.

$$\bar{\phi}_r^+ = T_{i-1}^+\phi_{i-1}^n + T_i^+\phi_i^n + T_{i+1}^+\phi_{i+1}^n \qquad \text{(E.32)}$$

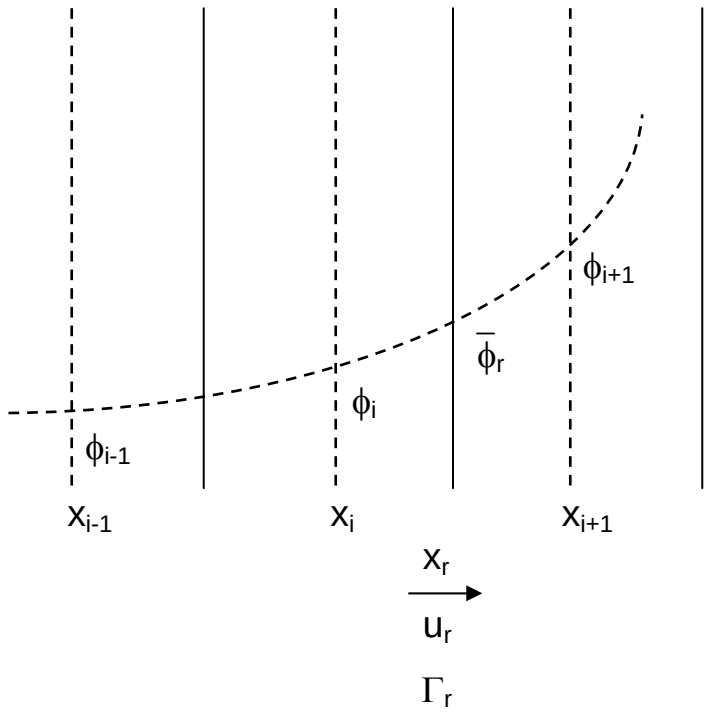


Figure E.2. Interpolation of concentration used for the case u > 0.

Evaluating the derivatives of $P_2(x)$ and substituting into (E.29) gives:

$$\bar{\phi}_r^{+} = \left[\frac{x_{i+1} - x}{x_{i+1} - x_i} \phi_i^n + \frac{x - x_i}{x_{i+1} - x_i} \phi_{i+1}^n \right]_{x=x_r}$$

$$- \frac{u\Delta t}{2} \left[\frac{2x - x_i - x_{i+1}}{\left(x_{i-1} - x_{i+1}\right)\left(x_{i-1} - x_i\right)} \phi_{i-1}^n \right.$$

$$\left. + \frac{2x - x_{i+1} - x_{i-1}}{\left(x_i - x_{i+1}\right)\left(x_i - x_{i-1}\right)} \phi_i^n + \frac{2x - x_i - x_{i-1}}{\left(x_{i+1} - x_i\right)\left(x_{i+1} - x_{i-1}\right)} \phi_{i+1}^n \right]_{x=x_r}$$

$$+ \left[\Gamma\Delta t - \frac{1}{6}\left(\Delta x_i^2 - u^2\Delta t^2\right) \right] \left[\frac{2}{\left(x_{i-1} - x_{i+1}\right)\left(x_{i-1} - x_i\right)} \phi_{i-1}^n \right.$$

$$\left. + \frac{2}{\left(x_i - x_{i+1}\right)\left(x_i - x_{i-1}\right)} \phi_i^n + \frac{2}{\left(x_{i+1} - x_i\right)\left(x_{i+1} - x_{i-1}\right)} \phi_{i+1}^n \right]_{x=x_r} \qquad \text{(E.33)}$$

The coefficients of ϕ_i^n in this expression are simply evaluated. Referring to Figure E.2, the numerator of the coefficient of $-\frac{u\Delta t}{2}\phi_{i+1}$ in Equation E.33 is:

$$2x - x_i - x_{i-1}\Big|_{x=x_r} = \left(x - x_i\right)_{x=x_r} + \left(x - x_{i-1}\right)_{x=x_r} = \frac{1}{2}\Delta x_i + \left(\Delta x_i + \frac{1}{2}\Delta x_{i-1}\right) = \frac{1}{2}\left(3\Delta x_i + \Delta x_{i-1}\right) \tag{E.34}$$

The denominator of the respective term is similarly calculated.

$$\left(x_{i+1} - x_i\right)\left(x_{i+1} - x_{i-1}\right) = \frac{1}{4}\left(\Delta x_{i+1} + \Delta x_i\right)\left(\Delta x_{i+1} + 2\Delta x_i + \Delta x_{i-1}\right) \tag{E.35}$$

The expression, for $\bar{\phi}_r^+$ (E.33) can then be written as:

$$\begin{aligned}
\bar{\phi}_r^+ &= \frac{\Delta x_{i+1}}{\Delta x_i + \Delta x_{i+1}}\phi_i^n + \frac{\Delta x_i}{\Delta x_i + \Delta x_{i+1}}\phi_{i+1}^n \\
&- \frac{u\Delta t}{2}\left[\frac{\frac{1}{2}\left(\Delta x_i - \Delta x_{i+1}\right)}{\frac{1}{4}\left(\Delta x_{i-1} + 2\Delta x_i + \Delta x_{i+1}\right)\left(\Delta x_{i-1} + \Delta x_i\right)}\phi_{i-1}^n + \frac{\frac{1}{2}\left(\Delta x_{i-1} + 2\Delta x_i - \Delta x_{i+1}\right)}{-\frac{1}{4}\left(\Delta x_i + \Delta x_{i+1}\right)\left(\Delta x_{i-1} + \Delta x_i\right)}\phi_i^n\right. \\
&\left. + \frac{\frac{1}{2}\left(\Delta x_{i-1} + 3\Delta x_i\right)}{\frac{1}{4}\left(\Delta x_i + \Delta x_{i+1}\right)\left(\Delta x_{i-1} + 2\Delta x_i + \Delta x_{i+1}\right)}\phi_{i+1}^n\right] + 2\Delta x_i^2\left[\left(\frac{\delta x}{\Delta x_i}\right)^2\frac{\Gamma\Delta t}{\delta x^2} - \frac{1}{6}\left(1 - c^2\right)\right] \\
&\times\left[\frac{1}{\frac{1}{4}\left(\Delta x_{i-1} + 2\Delta x_i + \Delta x_{i+1}\right)\left(\Delta x_{i-1} + \Delta x_i\right)}\phi_{i-1}^n + \frac{1}{-\frac{1}{4}\left(\Delta x_i + \Delta x_{i+1}\right)\left(\Delta x_{i-1} + \Delta x_i\right)}\phi_i^n\right. \\
&\left. + \frac{1}{\frac{1}{4}\left(\Delta x_i + \Delta x_{i+1}\right)\left(\Delta x_{i-1} + 2\Delta x_i + \Delta x_{i+1}\right)}\phi_{i+1}^n\right]
\end{aligned} \tag{E.36}$$

where δx is some characteristic length estimated as the minimum of the upstream and downstream cell sizes.

Values of the UQ advective multipliers T_i^+ can be found by equating the coefficients of ϕ_i^n in Equations E.32 and E.36. Approximating $\delta x/\Delta x_i \doteq 1$ in the physical dispersion terms (Chapman, private communication) gives:

$$T_i^+ = \frac{\Delta x_{i+1}}{\Delta x_i + \Delta x_{i+1}} + \frac{2\Delta x_i^2\left[\frac{\Gamma\Delta t}{\delta x^2} - \frac{1}{6}\left(1-c^2\right)\right] - \frac{u\Delta t}{2}\left[\frac{1}{2}\left(\Delta x_{i-1} + 2\Delta x_i - \Delta x_{i+1}\right)\right]}{-\frac{1}{4}\left(\Delta x_i + \Delta x_{i+1}\right)\left(\Delta x_{i-1} + \Delta x_i\right)} \tag{E.37}$$

$$T_{i-1}^+ = \frac{2\Delta x_i^2\left[\frac{\Gamma\Delta t}{\delta x^2} - \frac{1}{6}\left(1-c^2\right)\right] - \frac{u\Delta t}{2}\left[\frac{1}{2}\left(\Delta x_i - \Delta x_{i+1}\right)\right]}{\frac{1}{4}\left(\Delta x_{i-1} + 2\Delta x_i + \Delta x_{i+1}\right)\left(\Delta x_{i-1} + \Delta x_i\right)} \tag{E.38}$$

$$T_{i+1}^+ = \frac{\Delta x_i}{\Delta x_i + \Delta x_{i+1}} + \frac{2\Delta x_i^2\left[\frac{\Gamma\Delta t}{\delta x^2} - \frac{1}{6}\left(1-c^2\right)\right] - \frac{u\Delta t}{2}\left[\frac{1}{2}\left(\Delta x_{i-1} + 3\Delta x_i\right)\right]}{\frac{1}{4}\left(\Delta x_i + \Delta x_{i+1}\right)\left(\Delta x_{i-1} + 2\Delta x_i + \Delta x_{i+1}\right)} \tag{E.39}$$

Numerators and denominators of these coefficients are represented by the FORTRAN variables shown in Table E.1. As before, the additional factor of δx present in the definitions of the numerator terms represents some typical grid size and is presented for purposes of computational convenience. Scale factors representing δx and Δx_i^2 are assigned FORTRAN variables as shown in Table E.1. Using the FORTRAN variables shown in Tables E.1 and E.3, the UQ advective multipliers can be written as:

$$\mathrm{TERM1}^+ = \frac{\mathrm{SF2} \bullet \mathrm{ALFA} - \frac{1}{2}\mathrm{COUR} \bullet \mathrm{TP1}^+}{\mathrm{DEN1}^+} \tag{E.40}$$

$$\mathrm{TERM2}^+ = \mathrm{T2}^+ + \frac{\mathrm{SF2} \bullet \mathrm{ALFA} - \frac{1}{2}\mathrm{COUR} \bullet \mathrm{TP2}^+}{\mathrm{DEN2}^+} \tag{E.41}$$

$$TERM3^{+} = T3^{+} + \frac{SFA \bullet ALFA - \frac{1}{2} COUR \bullet TP3^{+}}{DEN3^{+}} \tag{E.42}$$

which are directly identifiable in Chapman's implementation of the UQ code (Chapman *et al*., 1997).

E.3.1.2 Solutions for Advective Multipliers for the Case u < 0

Solution for the case u < 0 makes use of a quadratic interpolation of the concentration ϕ that is downstream of the cell wall x_r (Figure E.3). Making use of the superscript "-" to denote results that are specific to the u < 0 case, we have the following:

$$\bar{\phi}_r^{-} = P_1(x) = \frac{u\Delta t}{2}(P_3'(x) + \left[\Gamma\Delta t - \frac{1}{6}\left(\Delta x^2 - u^2\Delta t^2\right)\right]P_3''(x) \tag{E.43}$$

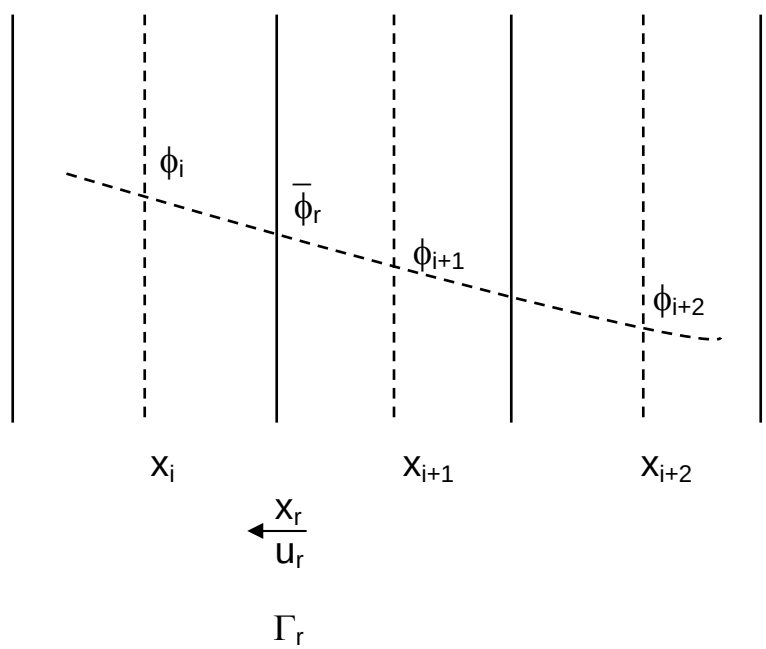


Figure E.3. Interpolation of concentration used for the case u < 0.

where $P_1(x)$ is given in (E.30) and $P_3(x)$ is, by analogy to $P_2(x)$, a second-order Lagrange interpolation polynomial $P_3(x;i) = P_2(x;i+1)$.

$$P_3(x) = \frac{(x - x_{i+2})(x - x_{i+1})}{(x_i - x_{i+2})(x_i - x_{i+1})}\phi_{i+1}^n + \frac{(x - x_{i+2})(x - x_i)}{(x_{i+1} - x_{i+2})(x_{i+1} - x_i)}\phi_{i+1}^n$$

$$+ \frac{(x - x_{i+1})(x - x_i)}{(x_{i+2} - x_{i+1})(x_{i+2} - x_i)}\phi_{i+2}^n \tag{E.44}$$

As before, substitution of $P_1(x)$ and $P_3(x)$ into (E.43) gives an expression for the concentration at the right cell face averaged over the time step Δt.

$$\bar{\phi}_r^- = \left[\frac{x_{i+1} - x}{x_{i+1} - x_i}\phi_i^n + \frac{x - x_i}{x_{i+1} - x_i}\phi_{i+1}^n\right]_{x=x_r} - \frac{u\Delta t}{2}\left[\frac{2x - x_{i+1} - x_{i+2}}{(x_i - x_{i+2})(x_i - x_{i+1})}\phi_i^n\right.$$

$$\left. + \frac{2x - x_{i+2} - x_i}{(x_{i+1} - x_{i+2})(x_{i+1} - x_i)}\phi_{i+1}^n + \frac{2x - x_{i+1} - x_i}{(x_{i+2} - x_{i+1})(x_{i+2} - x_i)}\phi_{i+2}^n\right]_{x=x_r}$$

$$+ \left[\Gamma\Delta t - \frac{1}{6}\left(\Delta x_i^2 - u^2\Delta t^2\right)\right]\left[\frac{2}{(x_i - x_{i+2})(x_i - x_{i+1})}\phi_i^n\right.$$

$$\left. + \frac{2}{(x_{i+1} - x_{i+2})(x_{i+1} - x_i)}\phi_{i+1}^n + \frac{2}{(x_{i+2} - x_{i+1})(x_{i+2} - x_i)}\phi_{i+2}^n\right]_{x=x_r} \tag{E.45}$$

$$\bar{\phi}_r^- = \frac{\Delta x_{i+1}}{\Delta x_i + x_{i+1}}\phi_i^n + \frac{\Delta x_i}{\Delta x_i + \Delta x_{i+1}}\phi_{i+1}^n - \frac{u\Delta t}{2}\left[\frac{-\frac{1}{2}(3\Delta x_{i+1} + \Delta x_{i+2})}{\frac{1}{4}(\Delta x_i + 2\Delta x_{i+1} + \Delta x_{i+2})(\Delta x_i + \Delta x_{i+1})}\phi_i^n\right.$$

$$\left. + \frac{\frac{1}{2}(\Delta x_i - 2\Delta x_{i+1} - \Delta x_{i+2})}{-\frac{1}{4}(\Delta x_{i+1} + \Delta x_{i+2})(\Delta x_i + \Delta x_{i+1})}\phi_{i+1}^n + \frac{\frac{1}{2}(\Delta x_i - \Delta_{i+1})}{\frac{1}{4}(\Delta x_{i+1} + \Delta x_{i+2})(\Delta x_i + 2\Delta x_{i+1} + \Delta x_{i+2})}\phi_{i+2}^n\right]$$

$$+ 2\Delta x_i^2\left[\left(\frac{\delta x}{\Delta x_i}\right)^2\frac{\Gamma\Delta t}{\delta x^2} - \frac{1}{6}\left(1 - c^2\right)\right] x \left[\frac{1}{\frac{1}{4}(\Delta x_i + 2\Delta x_{i+1} + \Delta x_{i+2})(\Delta x_i + \Delta x_{i+1})}\phi_i^n\right.$$

$$\left. + \frac{1}{-\frac{1}{4}(\Delta x_{i+1} + \Delta x_{i+2})(\Delta x_i + \Delta x_{i+1})}\phi_{i+1}^n + \frac{1}{\frac{1}{4}(\Delta x_{i+1} + \Delta x_{i+2})(\Delta x_i + 2\Delta x_{i+1} + \Delta x_{i+2})}\phi_{i+2}^n\right]$$

(E.46)

This solution is expressed as a weighted-sum of the concentration in the interpolated cells.

$$\bar{\phi}_r^- = T_i^- \phi_i^n + T_{i+1}^- \phi_{i+1}^n + T_{i+2}^- \phi_{i+2}^n \tag{E.47}$$

Comparison of (E.46) and (E.47) provide the solutions for the UQ advective multipliers for the case where u < 0 across the cell face. Again, approximating $\delta x/\Delta x_i \doteq 1$ in the physical dispersion terms gives:

$$T_i^- = \frac{\Delta x_{i+1}}{\Delta x_i + \Delta x_{i+1}} + \frac{2\Delta x_i^2 \left[\frac{\Gamma \Delta t}{\delta x^2} - \frac{1}{6}\left(1 - c^2\right)\right] - \frac{u\Delta t}{2}\left[-\frac{1}{2}\left(3\Delta x_{i+1} + \Delta x_{i+2}\right)\right]}{\frac{1}{4}\left(\Delta x_i + 2\Delta x_{i+1} + \Delta x_{i+2}\right)\left(\Delta x_i + \Delta x_{i+1}\right)} \tag{E.48}$$

$$T_{i+1}^- = \frac{\Delta x_i}{\Delta x_i + \Delta x_{i+1}} + \frac{2\Delta x_i^2 \left[\frac{\Gamma \Delta t}{\delta x^2} - \frac{1}{6}\left(1 - c^2\right)\right] - \frac{u\Delta t}{2}\left[-\frac{1}{2}\left(3\Delta x_{i+1} + \Delta x_{i+2}\right)\right]}{-\frac{1}{4}\left(\Delta x_{i+1} + \Delta x_{i+2}\right)\left(\Delta x_i + \Delta x_{i+1}\right)} \tag{E.49}$$

$$T_{i+2}^- = \frac{2\Delta x_i^2 \left[\frac{\Gamma \Delta t}{\delta x^2} - \frac{1}{6}\left(1 - c^2\right)\right] - \frac{u\Delta t}{2}\left[\frac{1}{2}\left(\Delta x_i - \Delta x_{i+1}\right)\right]}{\frac{1}{4}\left(\Delta x_{i+1} + \Delta x_{i+2}\right)\left(\Delta x_i + 2\Delta x_{i+1} + \Delta x_{i+2}\right)} \tag{E.50}$$

The FORTRAN variable representations of these expressions are given in Table E.2 with the results:

$$\mathrm{TERM1}^- = \mathrm{T1}^- + \frac{\mathrm{SF2} \bullet \mathrm{ALFA} - \frac{1}{2}\mathrm{COUR} \bullet \mathrm{TP1}^-}{\mathrm{DEN1}^-} \tag{E.51}$$

$$\mathrm{TERM2}^- = \mathrm{T2}^- + \frac{\mathrm{SF2} \bullet \mathrm{ALFA} - \frac{1}{2}\mathrm{COUR} \bullet \mathrm{TP2}^-}{\mathrm{DEN2}^-} \tag{E.52}$$

$$\mathrm{TERM3}^- = \frac{\mathrm{SF2} \bullet \mathrm{ALFA} - \frac{1}{2}\mathrm{COUR} \bullet \mathrm{TP3}^-}{\mathrm{DEN3}^-} \tag{E.53}$$

which are, again, directly identifiable in Chapman's implementation of the UQ code (Chapman *et al*., 1997).

E.3.2 ULTIMATE QUICKEST Dispersive Multipliers

Chapman's implementation of UQ (Chapman *et al*., 1997) for dispersion employs a method similar to that already discussed for advection; however, UQ dispersion multipliers are derived not for a time-averaged concentration at the right cell face, but for the mass flux. To the first order, we have that the [negative of the] time rate of change of mass is:

$$-\partial_t m = -\partial_t \int_v dv\phi = -\int_v dv\partial_t\phi \tag{E.54}$$

where m is the mass present within the cell and the integral $\int_v dv$ is over volume. The basic equation of continuity relating total flux density, mass/area/time to concentration in three dimensions is simply:

$$\vec{\nabla} \bullet \vec{j_T} = -\partial_t\phi \tag{E.55}$$

where $\vec{\nabla}$ is the divergence operator, $\vec{j_T}$ is the total (advective and dispersive) flux density, and ϕ is concentration. Using this result in (E.54) gives:

$$-\partial_t m = -\int_v dv\vec{\nabla} \bullet \vec{j_T} \tag{E.56}$$

By Gauss' divergence theorem, this is:

$$-\partial_t m = -\int_s \vec{ds} \bullet \vec{j_T} \tag{E.57}$$

It relates change in mass over time to the surface integral S over the cell face of the total flux density j_T. For a one-dimensional flux in the x direction, Equation E.57 gives:

$$-\partial_t m = -\left(A_r j_{T,r} - A_l j_{T,l}\right) \tag{E.58}$$

where A_r is the area of the right cell face, and $j_{T,r}$ is the total flux density flowing through the right cell face. Next, the advective j_a and dispersive j_d contributions to the total flux density are examined.

$$j_T = j_a + j_d \tag{E.59}$$

Breaking up the flux density into its advective and dispersive components gives:

$$-\partial_t m = -\left[A_r\left(j_{a,r} + j_{d,r}\right) - A_l\left(j_{a,l} + j_{d,l}\right)\right] \tag{E.60}$$

Comparisons of (E.1), (E.55), and (E.59) reveals the dispersive flux contribution to be

$$j_d = \Gamma \partial_x \phi^n \tag{E.61}$$

The dispersive contribution to the change in mass within a cell is given by:

$$\begin{aligned}\left(m^{n+1} - m^n\right)_{Dispersive} &= \int_{t^n}^{t^{n+1}} dt \left(A_r \Gamma_r \partial_x \phi_r^n - A_l \Gamma_l \partial_x \phi_l^n\right) \\ &\doteq \Delta t \left(A_r \Gamma_r \partial_x \phi_r - A_l \Gamma_l \partial_x \phi_l\right)\end{aligned} \tag{E.62}$$

which can be written in terms of the left and right dispersive mass fluxes

$$\left(m^{n+1} - m^n\right)_{Dispersive} = \Delta t \left(G_r - G_l\right) \tag{E.63}$$

where:

$$G_r \equiv A_r \Gamma_r \partial_x \phi_r^n \tag{E.64}$$

$$G_l \equiv A_l \Gamma_l \partial_x \phi_l^n \tag{E.65}$$

E.3.2.1 Solutions for Dispersive Multipliers for the Case u > 0

As before, the flux at the right cell face is evaluated by assuming a second-order Lagrange polynomial representation of the concentration $\phi(x) = P_2(x)$ given in (E.31) and shown in Figure E.1. Substitution into (E.64) gives one solution for the flux at the right cell face.

$$\begin{aligned}G_r^+ = A_r \Gamma_r \Bigg[&\frac{\left(2x - x_i - x_{i+1}\right)}{\left(x_{i-1} - x_{i+1}\right)\left(x_{i-1} - x_i\right)} \phi_{i-1}^n + \frac{2x - x_{i+1} - x_{i-1}}{\left(x_i - x_{i+1}\right)\left(x_i - x_{i-1}\right)} \phi_i^n \\ &+ \frac{2x - x_i - x_{i-1}}{\left(x_{i+1} - x_i\right)\left(x_{i+1} - x_{i-1}\right)} \phi_{i+1}^n \Bigg]_{x=x_r}\end{aligned} \tag{E.66}$$

In analogy to (E.32), a second solution for G_r^+ is proposed which is a linear combination of concentrations used in the interpolation $P_2(x)$.

$$G_r^+ = G_{i-1}^+\phi_{i-1}^n + G_i^+\phi_i^n + G_{i+1}^+\phi_{i+1}^n \tag{E.67}$$

Comparing the coefficients of ϕ_i^n in (E.66) and (E.67) gives expressions for the UQ dispersive multipliers

$$G_{i-1}^+ = A_r\Gamma_r\left[\frac{2x - x_i - x_{i+1}}{\left(x_{i-1} - x_{i+1}\right)\left(x_{i-1} - x_i\right)}\right]_{x=x_r} \tag{E.68}$$

$$G_i^+ = A_r\Gamma_r\left[\frac{2x - x_{i+1} - x_{i-1}}{\left(x_i - x_{i+1}\right)\left(x_i - x_{i-1}\right)}\right]_{x=x_r} \tag{E.69}$$

$$G_{i+1}^+ = A_r\Gamma_r\left[\frac{2x - x_i - x_{i-1}}{\left(x_{i+1} - x_i\right)\left(x_{i+1} - x_{i-1}\right)}\right]_{x=x_r} \tag{E.70}$$

As for the case of the UQ advective multipliers, evaluation at the right cell face gives:

$$G_{i-1}^+ = A_r\Gamma_r\frac{\frac{1}{2}\left(\Delta x_i - \Delta x_{i+1}\right)}{\frac{1}{4}\left(\Delta x_{i-1} + 2\Delta x_i + \Delta x_{i+1}\right)\left(\Delta x_{i-1} + \Delta x_i\right)} \tag{E.71}$$

$$G_i^+ = A_r\Gamma_r\frac{\frac{1}{2}\left(\Delta x_{i-1} + 2\Delta x_i - \Delta x_{i+1}\right)}{-\frac{1}{4}\left(\Delta x_i + \Delta x_{i+1}\right)\left(\Delta x_{i-1} + \Delta x_i\right)} \tag{E.72}$$

$$G_{i+1}^+ = A_r\Gamma_r\frac{\frac{1}{2}\left(3\Delta x_i + \Delta x_{i-1}\right)}{\frac{1}{4}\left(\Delta x_i + \Delta x_{i+1}\right)\left(\Delta x_{i-1} + 2\Delta x_i + \Delta x_{i+1}\right)} \tag{E.73}$$

Associations with FORTRAN variables are provided in Table E.1, giving:

$$GRAD1^+ = \frac{DIFF \bullet A \bullet TP1^+}{DEN1^+ \bullet SF1} \tag{E.74}$$

$$GRAD2^{+} = \frac{DIFF \bullet A \bullet TP2^{+}}{DEN2^{+} \bullet SF1} \tag{E.75}$$

$$GRAD3^{+} = \frac{DIFF \bullet A \bullet TP3^{+}}{DEN3^{+} \bullet SF1} \tag{E.76}$$

which are, once again, readily identifiable in the FORTRAN transport code.

E.3.2.2 Solutions for Dispersive Multipliers for the Case u < 0

In analogy to (E.66), a solution for G_r^- is proposed:

$$G_r^- = A_r\Gamma_r \left[\frac{\left(2x - x_{i+1} - x_{i+2}\right)}{\left(x_i - x_{i+2}\right)\left(x_i - x_{i+1}\right)} \phi_i^n + \frac{2x - x_i - x_{i+2}}{\left(x_{i+1} - x_{i+2}\right)\left(x_{i+1} - x_i\right)} \phi_{i+1}^n \right.$$
$$\left. + \frac{2x - x_i - x_{i+1}}{\left(x_{i+2} - x_{i+1}\right)\left(x_{i+2} - x_i\right)} \phi_{i+2}^n \right]_{x=x_r} \tag{E.77}$$

Again, a second solution for G_r^- is proposed:

$$G_r^- = G_i^-\phi_i + G_{i+1}^-\phi_{i+1} + G_{i+2}^-\phi_{i+2} \tag{E.78}$$

Evaluation of (E.77) at the right cell face gives the associated values for the UQ dispersive multipliers for the case u < 0.

$$G_i^- = A_r\Gamma_r \frac{-\frac{1}{2}\left(3\Delta x_{i+1} + \Delta x_{i+2}\right)}{\frac{1}{4}\left(\Delta x_i + 2\Delta x_{i+1} + \Delta x_{i+2}\right)\left(\Delta x_i + \Delta x_{i+1}\right)} \tag{E.79}$$

$$G_{i+1}^- = A_r\Gamma_r \frac{\frac{1}{2}\left(\Delta x_i - 2\Delta x_{i+1} - \Delta x_{i+2}\right)}{-\frac{1}{4}\left(\Delta x_{i+1} + \Delta x_{i+2}\right)\left(\Delta x_i + \Delta x_{i+1}\right)} \tag{E.80}$$

$$G_{i+2}^- = A_r\Gamma_r \frac{\frac{1}{2}\left(\Delta x_i - \Delta x_{i+1}\right)}{\frac{1}{4}\left(\Delta x_{i+1} + \Delta x_{i+2}\right)\left(\Delta x_i + 2\Delta x_{i+1} + \Delta x_{i+2}\right)} \tag{E.81}$$

Substituting with the FORTRAN variables shown in Tables E.2 and E.3 gives:

$$GRAD1^{-} = \frac{DIFF \bullet A \bullet TP1^{-}}{DEN1^{-} \bullet SF1} \tag{E.82}$$

$$GRAD2^{-} = \frac{DIFF \bullet A \bullet TP2^{-}}{DEN2^{-} \bullet SF1} \tag{E.83}$$

$$GRAD3^{-} = \frac{DIFF \bullet A \bullet TP3^{-}}{DEN3^{-} \bullet SF1} \tag{E.84}$$

E.3.3 Updating the Concentration

Solution for the updated concentration ϕ_i^{n+1} is straightforward. Equation E.26 is consistent with upwind difference advection of the time-averaged, UQ corrected concentrations $\bar{\phi}_r$ and $\bar{\phi}_l$. Advection and dispersion contributions to the mass derivative for each cell can, therefore, be calculated:

$$\partial_t m = Q_l \bar{\phi}_l - Q_r \bar{\phi}_r + G_r - G_l \tag{E.85}$$

where Q_l and Q_r are the flows through the left and right faces of the cell.

Implementation of this in FORTRAN code is performed separately at the upstream and downstream cells of each flow face. In LM3, the following calculation is performed

$$DTM(IB) = DTM(IB) - Q \bullet CADV + CDIFF \tag{E.86}$$

$$DTM(JB) = DTM(JB) + Q \bullet CADV - CDIFF \tag{E.87}$$

where IB is the upstream cell index, JB is the downstream cell index, and Q is the flow through the respective cell face. The advected concentration CADV and dispersion mass flux CDIFF are calculated using the relations (E.32), (E.47), (E.67), and (E.78) which are readily identified in the FORTRAN code.[1]

---

[1]The advection calculation actually makes use of an intermediate FORTRAN variable FLUXC used to perform additional numerical corrections to the UQ calculation.

$$CADV = TERM1 \bullet CONC1 + TERM2 \bullet CONC2 + TERM3 \bullet CONC3 \tag{E.88}$$

$$CDIFF = GRAD1 \bullet CONC1 + GRAD2 \bullet CONC2 + GRAD3 \bullet CONC3 \tag{E.89}$$

where CONC1, CONC2, and CONC3 represent the concentrations of the cells about which the UQ interpolation is performed. The ICM method is most similar but combines those advection and diffusion factors that multiply values of the concentration unique to each cell before performing that multiplication.

Update of the concentration in each cell is then calculated as:

$$\phi_i^{n+1} = \frac{\phi_i^n v_i^n + \Delta t \partial_t m}{v_i^{n+1}} \tag{E.90}$$

where v_i^n represents the volume of cell i at time t^n. The FORTRAN implementation of this is:

$$CSTAR = \frac{C \bullet V1 + DLT \bullet DTM}{V2} \tag{E.91}$$

Associations between the FORTRAN variables and the symbolic notation are provided in Table E.3.

References

Cerco, C. and T. Cole. 1994. Three-Dimensional Eutrophication Model of Chesapeake Bay, Volume I: Main Report. U.S. Army Corps of Engineers, U.S. Army Engineer Waterways Experiment Station, Baltimore, Maryland. Technical Report EL-94-4, 658 pp.

Cerco, C. and T. Cole. 1995. User's Guide to the CE-QUAL-ICM Three-Dimensional Eutrophication Model. U.S. Army Corps of Engineers, U.S. Army Engineer Waterways Experiment Station, Vicksburg, Mississippi. Technical Report EL-95-15, 2,420 pp.

Chapman, R.S., T.M. Cole, and T.K. Gerald. 1997. Development of a Hydrodynamic/Water Quality (POM-IPX) Linkage for the Lake Michigan Mass Balance Project. Final Report. U.S. Environmental Protection Agency, Office of Research and Development, National Health and Environmental Effects Research Laboratory, Mid-Continent Ecology Division-Duluth, Large Lakes Research Station, Grosse Ile, Michigan. 63 pp.

Hall, R. and R. Chapman. 1985. Two-Dimensional QUICKEST: Solution of the Depth-Averaged Transport Dispersion. U.S. Army Corps of Engineers, U.S. Army Engineer Waterways Experiment Station, Vicksburg, Mississippi. Technical Report EL-85-3, 33 pp.

Leonard, B. 1979. A Stable and Accurate Convective Modelling Procedure Based on Quadratic Upstream Interpolation. Comp. Methods Appl. Mechan. Engin., 19(1):59-98.

Leonard, B. 1991. The ULTIMATE Conservative Difference Scheme Applied to Unsteady One-Dimensional Advection. Comp. Methods Appl. Mechan. Engin., 88(1):17-74.

Velleux, M., S. Westenbroek, J. Ruppel, M. Settles, and D. Endicott. 2001. A User's Guide to IPX, The In-Place Pollutant Export Water Quality Modeling Framework. U.S. Environmental Protection Agency, Office of Research and Development, National Health and Environmental Effects Research Laboratory, Mid-Continent Ecology Division-Duluth, Large Lakes Research Station, Grosse Ile, Michigan. 179 pp.