Guidance Document for PMF Applications with the Multilinear Engine

EPA 600/R-09/032
April 2009
www.epa.gov

Guidance Document for PMF Applications with the Multilinear Engine

Gary Norris, Ram Vedantham
U.S. Environmental Protection Agency
National Exposure Research Laboratory
Research Triangle Park, NC 27711

Katie Wade, Patrick Zahn, Steve Brown
Sonoma Technology Inc.
Petaluma, CA 94954

Pentti Paatero
University of Helsinki
Helsinki, Finland

Shelly Eberly
Geometric Tools, Inc
Phoenix, AZ

Chuck Foley
Lockheed Martin
Systems Engineering Center
Arlington, VA 22201

Disclaimer

EPA through its Office of Research and Development funded and managed the research and development described here under contract 68-W-04-005 to Lockheed Martin. The document has been subjected to Agency review and is cleared for official distribution by the EPA. Although this work was reviewed by EPA and approved for publication, it may not necessarily reflect official Agency policy. Mention of trade names or commercial products does not constitute endorsement or recommendation for use.

TABLE OF CONTENTS

1.0 INTRODUCTION

This document serves as a guide for users of the Multilinear Engine version 2 (ME-2) for source apportionment applications utilizing positive matrix factorization (PMF). It aims to educate experienced source apportionment analysts on available ME rotational tools and provides guidance sensitivity analyses. Prior to using ME-2 for PMF, users should be familiar with the PMF model and its applications. Users should also be familiar with the wide body of literature describing PMF and ME-2. In particular, the End User's Guide (Paatero, 2004) and me2scrip.txt (Paatero, 2002) are useful companions to this document.

This document covers technical details and examples using two versions of ME-2. An individual ME-2 license (IL) can be purchased from Pentti Paatero or ME-2 is provided with the EPA distributed PMF 3.0 public license (PL). Both versions of ME-2 have been developed by Pentti Paatero and the version provided with the IL has a more flexible programming format and the PL version has a more restricted structure since it has been developed for EPA PMF software. Presenting two versions of the ME-2 and the associated input files increases the complexity of the document, however, it provides comprehensive explanation and examples of available ME-2 rotational tools.

1.1 ME-2

ME-2 is a least squares program for solving multilinear and quasi-multilinear problems (Paatero, 2000a). Specifically, it solves models where the data values are fitted by sums of products of unknown factor elements (Paatero, 2000a). For problems consisting of two groups of factor elements, such as those addressed in this document, the model is called a *bilinear factor analytic model* (Paatero, 2000a) and takes the form

$$\mathbf{X = GF + E} \qquad \textbf{(1-1)}$$

where **X** is a matrix of measured data, **G** and **F** are the factor matrices to be determined by ME-2, and **E** is the matrix of residuals or error terms (Paatero et al., 2003). In component form, the equation becomes

$$X_{ij} = \sum_{k=1}^{p} g_{ik} f_{kj} + e_{ij} \qquad \textbf{(1-2)}$$

For source apportionment applications like PMF:

x_{ij} represents the concentration of measured ambient species j in sample i;

p is the number of factors contributing to the measured sample and is provided to the model by the user;

f_{kj} is the concentration of species j in factor profile k;

g_{ik} is the relative contribution of factor k to sample i; and

e_{ij} is the residual for the species j in sample i.

The elements of **G** and **F** are constrained to non-negative values only, since neither a source contribution (**G**) nor its composition (**F**) can be negative; i.e., a source cannot emit negative mass or have a significant negative contribution for a species.

PMF incorporates estimates for sample-specific uncertainties. These uncertainties include both measured uncertainties and model uncertainties, and they can be provided by the analyst or generated by the model. Uncertainties allow each data point to be individually weighted in the PMF solution. The influence of each data point can be adjusted depending on the confidence in the measurement, retaining data that might otherwise be screened out and minimizing the impact of less certain data on the final solution (EPA, 2008). ME-2 finds a solution to PMF by iteratively minimizing the sum-of-squares object function, Q, based on these uncertainties:

$$Q = \sum_{i=1}^{n} \sum_{j=1}^{m} \left[\frac{e_{ij}}{s_{ij}} \right]^2 \quad \textbf{(1-3)}$$

where e_{ij} is defined as (from Eqn 1-2):

$$e_{ij} = x_{ij} - \sum_{k=1}^{p} g_{ik} f_{kj} \quad \textbf{(1-4)}$$

Since ME-2 operates by fitting data values to sums of products of unknown factor elements using a least squares method (Paatero, 2000a).

1.2 EPA PMF

The program EPA PMF was developed as a user interface for solving PMF equations, using the underlying program ME-2 as the factor analytic problem solver (EPA, 2008). The user provides data and specifications to the EPA PMF interface, which uses ME-2 to solve the PMF equations. EPA PMF also has a suite of tools for analyzing input data, viewing the resulting factor contributions and compositions, and analyzing the precision of the model solution. The initial version, 1.0, contained no tools for processing input data or performing rotations. Version 2.0 added a suite of tools for processing input data and model results. Version 3.0 includes additional tools and the capability to perform Fpeak and constrain elements of the contribution matrix. While EPA PMF provides a convenient interface for PMF, using ME-2 directly offers added control over the PMF model by allowing the user to constrain it to a greater degree than EPA PMF alone (Table 1). Additional constraining of the model is usually done by incorporating *a priori* information about factor species or the relationships between species. For example, knowledge regarding the source profiles or the contributions of specific sources can be used to constrain the PMF solution. The following table summarizes the features described in this document and their availability in EPA PMF and ME-2. Future versions of EPA PMF will include additional tools for rotational control.

Table 1. *The features described in this document and their availability in EPA PMF and ME-2.*

Feature	EPA PMF v3.0	ME-2
Control of main input parameters	✓	✓
Pre-processing of data (graphical and tabular)	✓	
Tabular output	✓	✓
Graphical output	✓	
Fpeak	✓	✓
AA.fkey (contribution matrix)		✓
BB.fkey (profile matrix)		✓
Auxiliary Equations		✓
Autopull		✓

1.3 Comparison of ME -2 with PMF2

Prior to the development of ME-2, PMF2 and PMF3 were developed to solve the PMF equations. These two programs solve a well defined problem (the bilinear and trilinear factor analytic models), which the

user can alter in small ways, for example by using Fpeak and fkey (see Section 3), but can not change the equations of the model. On the other hand, ME-2 is a general equation solver. The user defines all aspects of the problem to be solved, making ME-2 a much more flexible program than PMF2 or PMF3. ME-2 and PMF2 are similar in some details; however, the underlying process of the two programs is different. The differences in ME-2 and PMF2 have been examined in several studies by applying each model to the same data set comparing the results. Overall, the studies showed similar results for the major components, but a greater uncertainty in PMF2 results (Ramadan et al., 2003) and better source separation with ME-2 (Kim et al., 2007).

1.4 Definitions, Acronyms, and Abbreviations

Table 2. Terms and acronyms are used in this document.

Acronym	Definition
.ini file	Script file defining tasks for ME-2 to perform
Autopull	Auxiliary equation that allows the user to pull elements while specifying a limit to the change in Q
BDL	Below detection limit
CM	Configuration Management
CTM	Contract Task Manager
Element	A row or column of the **G** or **F** matrix, for example a factor contribution in a specific sample or the amount of a specific species in a given factor profile
EPA	U.S. Environmental Protection Agency
EPA PMF	Graphical user interface for PMF applications utilizing ME-2, developed by EPA
Factor	Group of species and their relative contributions
Fkey	PMF2 control code to constrain elements of **F** matrix
Fpeak	Tool for exploring rotational ambiguity in PMF solution
Gkey	PMF2 control code to constrain elements of **G** matrix
GUI	Graphical User Interface; in this document generally refers to EPA PMF
ITS-ESE	Information Technology Solutions − Environmental Systems Engineering
LM	Lockheed Martin
ME-2	Multilinear Engine version 2.0
PMF2, PMF3	Positive matrix factorization for 2-way (PMF2) and 3-way (PMF3) parafac models
Pull	Influence a factor element or group of elements towards a given value

Acronym	Definition
Q	The sum-of-squares object function
QA	Quality Assurance
Q(aux)	Contribution to sum-of-squares object function from auxiliary equations
Q(main)	Contribution to sum-of-squares object function from main equations
Residual	Difference between measured values and modeled value
Rotation	Linear transformation of a PMF solution that produces the same Q-value as the original solution
TNMOC	Total nonmethane organic compounds

1.5 References

Henry R.C. (1987) Current Factor Analysis Models are Ill-Posed. *Atmos. Environ.* **21**, 1815-1820.

Henry R.C. (2003) Multivariate receptor modeling by N-dimensional edge detection. *Chemometrics and Intelligent Laboratory Systems* **65**, 179- 189.

Kim E., Hopke P.K., and Edgerton E.S. (2004) Improving source identification of Atlanta aerosol using temperature resolved carbon fractions in positive matrix factorization. *Atmos. Environ.* **38**, 3349-3362.

Lanz V.A., Alfarra M.R., Baltensperger U., Buchmann B., Hueglin C., Szidat S., Wehrli M.N., Wacker L., Weimer S., Caseiro A., Puxbaum H., and Prevot A.S.H. (2008) Source attribution of submicron organic aerosols during wintertime inversions by advanced factor analysis of aerosol mass spectra. *Environ. Sci.Technol.* **42** (1), 214-220 (doi: 10.1021/es0707207).

EPA (2008) EPA Positive matrix factorization (EPA PMF) v3.0 user guide. June.

Paatero P. (1999) The multilinear engine - A table-driven, least squares program for solving multilinear problems, including the n-way parallel factor analysis model. *Journal of Graphical Statistics* **8**, 854-888.

Paatero P. (2000a) User's guide for the multilinear engine program "ME2" for fitting multilinear and quasi-multilinear models. February.

Paatero P. (2000b) User's guide for positive matrix factorization programs PMF2 and PMF3, part 1: tutorial. Prepared by University of Helsinki, Finland, February.

Paatero P. (2002) The multilinear engine (ME-2) script language (v. 1.05). December.

Paatero P. and Hopke P.K. (2002) Utilizing wind direction and wind speed as independent variables in multilinear receptor modeling studies. *Chemometrics and Intelligent Laboratory Systems* **60**, 25-41.

Paatero P., Hopke P.K., Song X.H., and Ramadan Z. (2002) Understanding and controlling rotations in factor analytic models. *Chemometrics and Intelligent Laboratory Systems* **60**, 253-264.

Paatero P., Hopke P.K., and Philip K. (2003) Discarding or downweighting high-noise variables in factor analytic models. *Anal. Chim. Acta* **490**, 277-289.

Paatero P. (2004) End user's guide to multilinear engine applications. August.

Paatero P., Hopke P.K., Begum B.A., and Biswas S.W. (2005) A graphical diagnostic method for assessing the rotation in factor analytical models of atmospheric pollution. *Atmos. Environ.* **39**, 193-201.

Paatero P. (2007a) Description of the C3 variable in ME-2. Personal communication with Katie Wade, December 4.

Paatero P. (2007b) Assigning mass to factor contributions in PMF. Personal communication with Katie Wade, December 28.

Paatero P. (2008) The multilinear engine (ME-2) script language (v. 1.210). February.

Paatero P. and Hopke P.K. (2008) Rotational tools for factor analytic models implemented by using the multilinear engine.

Reff A., Eberly S.I., and Bhave P.V. (2007) Receptor modeling of ambient particulate matter data using positive matrix factorization: review of existing methods. *J. Air & Waste Manag. Assoc.* **57**, 146-154.

Rizzo M.J. and Scheff P.A. (2007) Utilizing the chemical mass balance and positive matrix factorization models to determine influential species and examine possible rotations in receptor modeling results. *Atmos. Environ.* **41** (33), 6986-6998.

Wade K.S., Brown S.G., Turner J.R., and Garlock J.L. (2008) Concentration value uncertainty estimates for source apportionment modeling of chemical speciation network data. Manuscript prepared for *A&WMA's 101st Annual Conference & Exhibition, Portland, OR, June 24-27* (STI-3271; Paper No. 632).

Zhao W. and Hopke P.K. (2004) Source apportionment for ambient particles in the San Gorgonio wilderness. *Atmos. Environ.* **38**, 5901-5910.

1.6 Acknowledgments

The authors gratefully acknowledge the testing and review of staff at EPA Office of Research and Development. The authors also appreciate the researchers referenced in this document for their explorations with ME-2 and other models.

2.0 ME-2 PARAMETERS

2.1 ME-2 Versions

A free version of ME-2 can be obtained as part of the EPA PMF 3.0 software download (http://www.epa.gov/heasd/products/pmf/pmf.htm). Since ME-2 is the underlying engine that is used by the EPA PMF program, ME-2 is available with the EPA PMF 3.0 installation package. When EPA PMF 3.0 is installed (typically in the C:\Program Files folder), ME-2 executable is also installed in the same folder under the title of **me2wopt.exe**. This version of ME-2 is distributed under a public license and hence will be called the Public License (PL) version throughout this document.

The user can also run ME-2 (me2wopt.exe) independently using the MS-DOS command prompt. However, it is important to note that the ME-2 will function properly only if the certain other files are present in the same folder. This includes the script file (PMF_bs2.ini), the public key file (me2key.key), parameter file(s) iniparams.txt and moreparams.txt (needed only for rotating and pulling, explained below). In addition, the input data file must be present in the same folder as the executable. In addition, to run the rotational tools, the base run output file should also be present in the same folder. The names of the parameter files should not altered since they are hard-coded into the ME-2 executable. It is advisable to make copies of the parameter files before any alterations are made so that a fresh start can always be made. Instructions on running ME-2 from a DOS command prompt are given in section 2.2. The command to run ME-2 is the same irrespective of the desired outcome. The user should type "me2wopt.exe PMF_bs2.ini" at the command prompt. To obtain various outcomes, changes should be made in the parameter files only. The script file (PMF_bs2.ini) should never be altered. More specific instructions are provided in Appendix A.

If the user wishes to use an Individual Version (IL) version obtained directly from the author Dr. Pentti Paatero (Pentti.Paatero@helsinki.fi), the command line should be altered slightly. Instead of me2wopt.exe, the user will have to type the appropriate name of the executable provided by Dr. Paatero. This version will be referred as the IL version throughout this document. In this case, supporting information for ME-2 can be downloaded from http://www.helsinki.fi/~paatero/PMF. Any user can also request a CD-ROM with supplementary literature from Dr. Paatero.

2.2 Running ME-2 from a Script File

In addition to the three files provided when obtaining ME-2, the user will need to generate a script file specific to the task to be accomplished. In the script file, the user tells ME-2 the location and format of the data to be modeled, the definition of the model to be fitted, and the location and format of the modeling results. The user also specifies algorithmic details such as the maximum number of iterations to use in the minimization process and the model convergence criteria. This script file is sometimes referred to as an "ini" file because the file extension on scripts is "ini" In this document, for any references to a specific script refer to 2way.ini, which is included with this document. This basic script, as well as other sample script files, can be downloaded from www.helsinki.fi/~paatero/PMF/me2_scripts.zip. The basic files needed are the ME-2 exe (e.g., me2wG17.exe), the ME-2 library (ME2libr.txt), and the key (me2key.key), plus the script (ini file) and data to be used.

The general structure of the script and specific aspects of the script that are necessary for PMF modeling of environmental data are described in the following subsections. Example scripts are included with the example data sets as separate zip files. For a full discussion of scripting details, see Paatero (2000b; 2002; 2004; Paatero, 2008).

ME-2 is run through a MS-DOS command prompt window by navigating to the proper directory and entering the name of the executable followed by the name of the script. The user must provide a input data matrix with samples in rows and species in columns. This matrix should be saved as a text file with no headers. If an uncertainty matrix is also provided, it should follow directly below the data matrix with no blank lines in between. The reference scripts included with this document assume the input data file is in the same folder as the executable and script files.

2.2.1 Notes on Script Syntax

To ensure that scripts are easily read by any user, they should be written in a text editor in a fixed-pitch font (such as Ultra-Edit). Spaces can be used (but are not required anywhere in the script); tabs should be avoided. Comments must also be included to document scripts (especially when changes are made to existing scripts). There are two ways to include a comment: 1) shorter comments can be included by starting each comment line with a %; and 2) longer comments can be included between the commands '`$skiplines`' and '`$endskip`'. The user should also be certain to remove obsolete comments from a script. All variable names are defined by the user in the .ini file; variable names in this document are commonly used but may be changed by the user. The **F** matrix is referred to as the **BB** matrix when referenced in the script and the **G** matrix is referred to as the **AA** matrix.

2.2.2 Script Structure

Specific script language and structure is discussed in detail in following sections, and available in section 5.0 and the example scripts which accompany this document. The first line of each ME-2 script must have the location of the key file (example: `##ME-2 script for PMF. Licence: C:\Program Files\me2\me2key.key`). The rest of the script is divided into five sections, as detailed in Table 3. All sections must be present and in the same order as they are presented in Table 3. Each section is initiated by the command '`section>`' and ended by the command '`section!`' Multiple commands can be used on one line, ending each command with a semicolon.

Table 3. *Description of required sections of an ME-2 script.*

Section	Purpose
defines	Assign values to special variables, declare arrays and subroutines
equations	Generate model equations, initialize variables
preproc	Set initial data values, any preprocessing
postproc	Post-processing such as writing results to file
callback	Can be empty or can be used to monitor/influence iterations

2.2.3 Variables of Interest in the .ini Files

Table 4 contains the most common variables in an ME-2 script. All of these variables are included in the reference script 2way.ini. The user should, at minimum, verify that these variables are set to appropriate values. The sections below provide more information on specific variables

Table 4. *Description of common variables in an ME-2 script for PMF*

Variable Name	Description	Typical Value
version	Version of ME-2 executable that script works with	Current version is 1.203
monitor	Controls the amount of data written to the file me2.log; has no influence on results. Smaller numbers result in more information written to the file.	5-20
robust	Defines outliers and how they are handled. See discussion below, Robust Mode	
posoutdist		
negoutdist		

Variable Name	Description	Typical Value
missdatlim	Code used to indicate missing data values	Must be numeric, set to very small value, such as -7.7E17, if there are no missing data values
bdlneg	Set to 1 if negative values represent below detection limit data; otherwise set to 0	0 or 1
convtests	Criteria defining convergence for minimization algorithm. See discussion below, Convergence Criteria	
cgresets		
numtasks	Number of random starts to be computed. See below for more detail.	20+ for initial analysis
contrun	If results from a previous run are to be used as a starting point, contrun=1 (see discussion below); if a random starting point is to be used, contrun =0.	0,1
numoldsol	If results from a previous run are to be used as a starting point (contrun=1), this value should be set to the number of the solution (in the input file) that is to be used.	
alowlim	Lower limit for **G** matrix	-0.1 (allowed
blowlim	Lower limit for **F** matrix	0
seed1	Used to initiate random number generator for initial factor values	Any positive value
##d=' '	Name of file containing input data matrix in quotes: concentration matrix immediately followed by uncertainty matrix (if provided)	
##p=' '	Name of file in quotes containing previous ME-2 solution (if starting from previous run)	Must be in .dat format
##m=' '	Prefix for output files in quotes	
n1	Number of rows in input data matrix	
n2	Number of columns in input data matrix	
np	Number of factors	
c1	Variables that determine the total uncertainty for each observation. See discussion below, Uncertainty Estimates, Error Model Codes, and Global Uncertainty Parameters	
c2		
c3		
em		

Robust Mode

The robust mode is used by specifying "robust=1" in the script. In PMF, outliers (points where the difference between the measured value and the modeled value is large) can have a considerable impact on the modeling process. Outliers can be addressed by using robust mode, which allows for re-weighting of data points between iterations, reducing the weights for points where the model fit is poor. In this manner, the influence of poorly fit measurements on PMF solutions is diminished (Reff et al., 2007). Robust mode is selected by setting `robust = 1` in the '`defines`' section of the script (setting `robust = 0` indicates the non-robust mode, where outliers are not down-weighted). The user specifies a scaled

residual (e_{ij}/s_{ij}) threshold beyond which data are considered outliers and are down-weighted. This threshold is set using the variables `posoutdist` (above which scaled residuals are considered outliers) and `negoutdist` (below which scaled residuals are considered outliers). A typical value for the scaled residual threshold is +/- 4 (`posoutdist = 4`, `negoutdist = 4`) (Reff et al., 2007). The use of robust mode affects the calculation of Q, the sum of squares object function: for scaled residuals above or below the positive and negative thresholds respectively, the scaled residual used in calculating the Q-value simply becomes the threshold value (+/- 4 for the case above). The Q-value using these weights is referred to as Q-robust and is defined similarly to Q (see Equation 1-3):

$$Q_{robust} = \sum_{i=1}^{n}\sum_{j=1}^{m}\left[E_{ij}\right]^2 \qquad \textbf{(1-5)}$$

where

$$E_{ij} = \frac{e_{ij}}{s_{ij}} \text{ for } \texttt{negoutdist} \le \frac{e_{ij}}{s_{ij}} \le \texttt{posoutdist}$$

E_{ij} takes the value of `negoutdist` or `posoutdist` for $\frac{e_{ij}}{s_{ij}}$ below or above the respective thresholds (Reff et al., 2007).

The use of robust mode is recommended as the default in environmental cases. Non-robust mode may be used if the data are known to be normally distributed and there are no outliers and no non-representative values in the data (Paatero, 2000a). On occasion, the use of robust mode can be detrimental, which may happen when (1) two sources differ only in very few variables (possibly in only one variable) and (2) one of the two sources is only present in a small fraction of all samples (Paatero, 2007a). The user should run the model in robust and non-robust modes and compare the resulting Q-values. If Q is diminished (by 10s or more) by implementing robust mode, then the model fitting is likely sensitive to outliers, and robust mode should be used.

Calculation of Total Uncertainty, *S*, using Error Model Codes (`em`), and Global Uncertainty Parameters (`C1`, `C2`, and `C3`)

These parameters are specified at the end of the first section “defines”, as follows:

% std-dev coefficients and errormodel code for the main equations

c1=0.0; c2=0.0; c3=0.00; em=-14;

The total uncertainty estimate (*S* in Equation 1-3) should encompass both measurement uncertainty, such as analytical or sampling errors, and model uncertainty, such as variations in source profiles over time. ME-2 calculates the total uncertainty based on a user-specified error model, which defines the relative contributions of the measurement uncertainty and the model uncertainty to the total uncertainty. The error model defines these contributions using three global uncertainty parameters—`C1`, `C2`, and `C3` (See **Table 5** for more details.). The error model also defines whether the total uncertainty is calculated once before the iterations begin, or if it is re-calculated between model iterations.

The global uncertainty parameters are defined as follows:

C1 Measurement uncertainty. If the user supplies no measurement uncertainties, the model will set them equal to the value of C1. Any user-provided uncertainties over-write the C1 matrix.

C2 Applies only to Poisson-distributed data and is set to zero for the majority of environmental applications.

C3 Model uncertainty coefficient. Describes such expected residuals that are not caused by measurement (or laboratory) errors, such as variation of source profiles with time. In most error models C3 is multiplied by the species concentration x_{ij}, as it is assumed to vary with the magnitude of the observed value.

The user can provide an uncertainty matrix that will override the `C1` matrix calculated by ME-2 ($C1_{ij}$ in **Table 5**). If the user-specified uncertainty does not include modeling uncertainty, the `C3` parameter can be used. The user specified uncertainty file should be the same dimensions as the concentration file and should follow the concentration file directly in the input file (##d).

Table 5. *Description of parameters used in uncertainty estimates*

Parameter	Variable Name in .ini File	Typical Values/Range
Sample specific uncertainty estimate	Provided in separate file	Dependent on data set
Global uncertainty parameters	C1	Dependent on data set
	C2	0 except for Poisson-distributed data
	C3	0–0.25 for environmental problems
Error model code	em	Usually -12 or -14 for environmental data

Table 6 summarizes some of the available error model codes and how each is calculated using the global uncertainty parameters `C1`, `C2`, and `C3`. The global uncertainty parameters may be constant for all observations and species, they may vary by row or by column, or they may be observation-specific. To specify parameter values for individual observations, the command `XX.C1[i,j]=b` or `XX.C3[i,j]=c` can be included in the 'equations' section of the script. These values will override the existing C1 or C3 variables that are set using the global uncertainty parameters. Likewise, the error model code may be constant or may vary by row or by column.

Table 6. *Available error model codes (Paatero, 2000a).*

Error Model Code	Equation	Notes
-5	$S_{ij} = C1_{ij} + C2_{ij}\sqrt{sumabs} + C3_{ij}\,sumabs$	*sumabs* is the sum of absolute values of contributions to the fitted value. Used in equations where the difference (or sum) of two values is pulled toward zero.
-12	$S_{ij} = C1_{ij} + C2_{ij}\sqrt{\lvert s\rvert} + C3_{ij}\lvert s\rvert$	$s = \lvert x_{ij}\rvert$, where x_{ij} is the measured data. Commonly used in environmental applications.
-13	$S_{ij} = C1_{ij} + C2_{ij}\sqrt{\lvert s\rvert} + C3_{ij}\lvert s\rvert$	$s = \lvert y_{ij}\rvert$, where y_{ij} is the model-fitted value.
-14	$S_{ij} = C1_{ij} + C2_{ij}\sqrt{\lvert s\rvert} + C3_{ij}\lvert s\rvert$	$s = \max(\lvert x_{ij}\rvert, \lvert y_{ij}\rvert)$ Commonly used in environmental applications.

Error Model Code	Equation	Notes
-15		Indicates missing value, not fitted.
-16		If fitted value is below data value (i.e., if the residual is negative), the model pulls the fitted value toward the data value. However, if the residual is positive, the point is not weighted. Used when the detection limit has been substituted for values below the detection limit (BDL).
-18		*S* is "frozen". Can be used during the final iterations, e.g., to avoid extremely slow convergence that might result from the minute changes of weights when `em` = -14 or `em` = -12.
-20		Pull factor element or expressions of factor element up as far as possible, keeping Qmain limited.
-21		Pull factor element or expressions of factor element down as far as possible, keeping Qmain limited.
-22		Pull factor element or expressions of factor element up or down towards a target value, keeping Qmain limited.

Note: *S* is total uncertainty; *s* is user-provided.

Error model codes -20, -21, and -22 and used only with auxiliary pulling equations (see Sections 3 and 4). For these three codes, the user must also initialize the auxiliary variables

`NN.aux1[ii]` = dQ, the limit of increase of Qmain

`NN.aux3[ii]` = the absolute value of the expected change in the element of expression of elements

`NN.aux4[ii]` = the target value for the pulled quantity (for error model code -22 only)

Where NN is a defined array. The user may also enter a value for `NN.aux2[ii]`, which is the initial value of the pulled quantity. Use of this variable is optional and requires that dQ be given with a minus sign.

The most common error model codes for environmental applications are -12 and -14. As additional model equations are included, each one needs specific uncertainty parameters and an error model code.

In the case that the user provides an uncertainty matrix that includes measurement and model uncertainty, the model need not calculate the uncertainties. The global uncertainty parameters should be set to zero, and the user-provided uncertainties should be used as the total uncertainty. This function is performed by setting the error model code to -12 (with `C1 = C2 = C3 = 0`).

If the user does not provide an uncertainty file, or if the uncertainty file input does not contain both measurement and model uncertainty, different combinations of the global uncertainty parameters and error models can be used to calculate the total uncertainty *S*. For example, if the user-provided uncertainty does not include model uncertainty, the `C3` (the model uncertainty coefficient) value should be adjusted to an appropriate value, *a*. Then the user can use error model code -12 (with `C1 = C2 = 0,`

C3 = *a*) to compute the total uncertainty as defined in Table 6. To adjust the total uncertainty with each iteration, error model -14 can be used. In that case, the uncertainty is recalculated after each iteration using the max of the input value or the modeled value as s.

Typically in environmental work, each column of data represents a different chemical species. In this case, the error structures of all of the species cannot be assumed to be the same; thus, different values of C1 are needed for each column. These values can be provided through either a sample-specific uncertainty file, as discussed above, or an input file containing a matrix of C1 values. The C1 matrix generally has the same value for each row in a column. To avoid repeating the same value, the array should be transposed and written in repeat notation, where 100*5 represents 100 entries all equal to 5.

The user may have to explore appropriate values for C3 within the suggested range. The case studies in Section 5 demonstrate several examples. For example, in the Craig dataset, the uncertainties provided by the user were presumably conservative estimates and no further uncertainty was needed (C3=0). For the St. Louis dataset, an additional 10% modeling uncertainty was included (C3=0.1). The global uncertainty parameter C3 is not used by error model codes -20, -21, and -22.

Convergence Criteria (deltaQ test, consec steps, and max cumul step count**)**
As previously noted, PMF is solved iteratively, minimizing the sum-of-squares object function, Q. When Q has "converged," a stable solution has been reached. The user can adjust the convergence criteria in the 'defines' section of the script under the command convtest.

The three-row table immediately following the convtest command contains numerical values specifying convergence. Convergence is met if the change in Q is less than deltaQ test, over a given number of iterations, consecut. steps, within a given total number of iterations, max cumul. step count. The user provides the convergence criteria (these three variables) at three levels (one level on each row, indicated in the script below by a red box).

```
% Convergence tests and other parameters for the three
% iteration levels.
convtests
    0.100,   20,          300,           0,      0,      0.0001,     %level 1
    0.010,   40,          800,           0,      0,      0.0001,     %level 2
    0.0002,  60,          2000,          0,      0,      0.00002;    %level 3
% deltaQ     consecut.  max cumul.    not     not    gg2 norm
%  test      steps       step count     used    used   test
```

For example, in the script above, the change in Q must be less than 0.0002 over 60 consecutive steps by the 2,000th iteration. In general, the first level finds the correct region in space; the second level converges close to the final solution; the third stage reaches the best possible Q-values (Paatero, 2000a). Whereas a deltaQ test of 0.01 may be a good convergence limit for small models, larger models consisting of thousands of data points might converge well with limits in the range of 1 to 10 (Paatero, 2000a). The user is encouraged to experiment with different convergence limits. If the resulting Q does not change although the program is forced to run several hundred extra iterations, a good convergence has probably been achieved. ME-2 typically requires a few hundred iterations for simpler problems and up to 2,000 for more complex ones (Paatero, 2000a). If the deltaQ test is not met when the number of iterations exceeds max cumul. step count, the iterations are terminated at the level in question, although no convergence may be assumed (Paatero, 2000a).

Conjugate Gradient Resets (cgresets**)**
Immediately following the convtest section in the 'defines' part of the .ini file, there is an option for changing the variable cgresets. This variable determines the lengths of the conjugate gradient (CG) restart intervals. Altering it can influence the rate of Q convergence (Paatero, 2000a). For most environmental applications, the default values work well, and it is not recommended that the user change

them. However, it should be noted that if the upper limit of steps allowed in the CG sequence (the second term in the list, indicated below with a red box) is too small, difficult tasks may not converge properly (Paatero, 2000a).

```
cgresets 10, 80, 1, 1, 1, 1;
```

The first parameter of `cgresets` specifies the lower limit of steps allowed in a CG sequence. The third through sixth parameters are coefficients that allow the user to modify the allowed restart interval lengths cyclically. For no modification, they should be left at 1. However, if the user wishes to double the length of every fourth interval, for instance, one of the coefficients could be switched to 2 (Paatero, 2000a). For more details on `cgresets`, refer to Paatero (2000a).

Normalization of Factor Contributions

While ME-2 solves for the profiles of species concentrations for each factor (**BB**) and the relative contribution of each factor to a measured sample (**AA**), it does not control the relative contributions of **BB** and **AA** to the matrix **AABB**. As a result, it is possible that one of the factor elements (**BB** or **AA**) could be driven toward infinity while the other is driven to zero. For example, in PMF, one risks having a factor whose contribution (g_{ik}) is excessively large but whose factor species fractions (f_{ij}) are excessively small. This factor scaling indeterminacy should be removed to keep factor contributions of the same order of magnitude between factors (Paatero, 1999). The standard practice of ME-2 scripts for environmental applications is to normalize the **G** columns to an average value of 1 (Paatero and Hopke, 2008). The product **AABB** does not change when normalization is performed. In addition, normalizing factors does not influence factorization computations in any way, and Q-values should be minimally affected (User's Guide, part 2, Paatero, 2000). For more information on normalizing factor contributions, refer to Paatero (1999) and (Paatero and Hopke, 2008).

Most users will find that factor contributions in mass units are more useful. There are two common methods for obtaining mass contributions of each factor after normalization. The first method is linear regression using total mass as a dependent variable. After conducting the factor analysis, the contributions of each factor are regressed against the total mass. The coefficient for each factor is then used to scale the normalized factor contributions to mass units. The second approach is to include total mass as a species in PMF with very high uncertainty. The user can then use the mass apportioned to each factor as a scaling factor to determine mass contributions from normalized contributions. The inclusion of total mass with high uncertainty in the factor analysis does not heavily impact the final solution. Furthermore, including mass as a variable in ME-2 carries the benefit of guiding the solution towards a more realistic rotation where mass remains non-negative (Paatero, 2007b).

2.3 Evaluating ME-2 Output

For each model run, ME-2 generates three output files: *.dat, *.rsd, and *.txt, where * is the user-defined file label (variable `##m` in the '`defines`' section of the script). The .dat file contains the **BB** and **AA** matrices with no additional information. The **AA** matrix is the first set of values, followed by one blank line, followed by the **BB** matrix. If multiple runs are performed, the **AA** and **BB** matrixes for each one are provided in order with a line between each. The file .txt also contains the **BB** and **AA** matrices, along with the task number, seed value, Q-value, Qmain, Qaux, and the contribution to Qaux from the normalization equations. The .txt file also contains row and column headers (the default is a count of samples/species and "Fact01", etc.). Either the .dat or the .txt file can be used in a spreadsheet program, such as MS Excel, to calculate statistics or graph the output. Graphing the matrices is helpful in determining what sources the factors represent. The Q-value should be obtained from the .txt value. Q-values from multiple random starts should be compared to look for a global minimum value.

The final file, *.rsd, contains the scaled residuals (residual over the uncertainty) for each sample/species. The scaled residuals should be examined to determine if they are normally distributed and to look for outliers (typically greater than 4 or less than -4).

Section 5 contains detailed analysis of ME-2 output from a variety of example data sets. Additional details on evaluating PMF output can be found in EPA (2008).

3.0 ROTATIONS

In general, the non-negativity constraint alone in ME is not sufficient to produce a unique solution. This produces a multitude of plausible solutions none of which can be eliminated using mathematical algorithms. To reduce the number of solutions, additional information such as known source contributions and/or source compositions can be used and the use of this information is discussed in section 3.2.

Another approach to reduce the number of solutions is to rotate a given solution. The idea behind rotating a solution is to arrive at physically interpretable source composition and contribution. Often, these sources do not present themselves in a clear fashion. As mentioned above, additional information or the model results can be used to guide the selection of an appropriate rotation. For instance if only one source impacts a receptor site due to wind direction or operations, only one source would have a substantial contribution while others would have a zero contribution. This ideal situation may preclude the need for rotation. However, in reality most solutions are based on sources that may be present at the same and may muddle the interpretation of the sources profiles.

A method that helps indentify interpretable results uses a scatter plot of a pair of source contributions and they are called "G-space Plots." **Figure 1** shows G-space plots from the St. Louis example data set presented in Section 5.

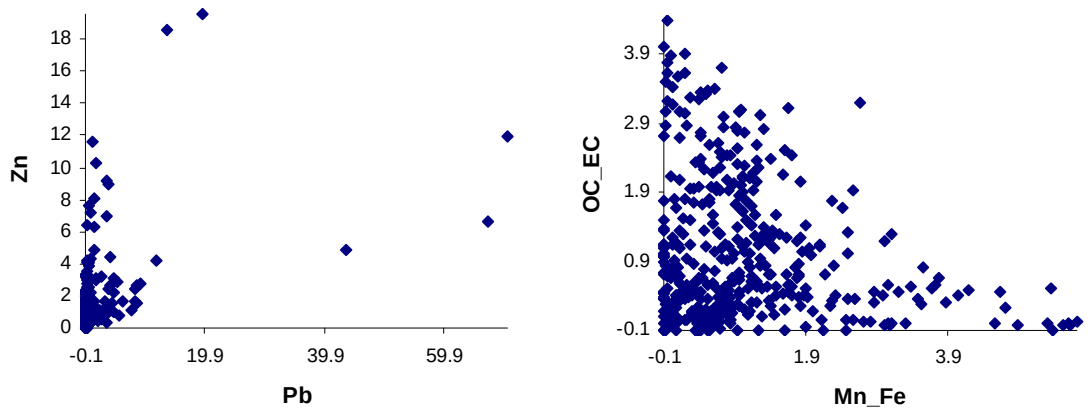


Figure 1. *G-space plots*

Mathematically, a pair of factor matrices (**G** and **F**) that can be transformed to another pair of matrices (***G****** and ***F****) with the same Q-value is said to be "rotated". The transformation takes place as follows:

$$G^* = GT \text{ and } F^* = T^{-1}F \qquad \textbf{(3-1)}$$

The *T* matrix is a *p* x *p*, non-singular matrix. If all the off diagonal elements are set to the same value, then this rotation is similar to the Fpeak rotation in PMF2. However, in ME-2, the non-diagonal values do not have to be the same. The range of all possible T values constitutes the rotational ambiguity in the solution. In PMF, this is not strictly a rotation but rather a linear transformation of the **G** and **F** matrices. Equation 3-1 represents a pure rotation where the Q-value will not change. Due to the non-negativity constraints in PMF, a pure rotation (i.e., a specific T matrix) is only possible if none of the elements of the new matrices is less than zero. Therefore, approximate rotations, which allow some increase in the Q-value and prevent any elements in the solution from becoming negative, are useful in PMF. There is no rule about how much the Q-value can increase, but in general an increase of a few tens of units is always acceptable, increases of hundreds of units could be acceptable depending on the data set, and increases of thousands of units are likely not acceptable (Paatero and Hopke, 2008; Paatero et al., 2002). If the Q-value of the rotated solution is less than the Q-value of the original solution, the original solution was a

local minimum. More base runs starting at pseudorandom points should be done to look for a true global minimum. Because of the subjective nature of determining if an increase in Q-values is acceptable, the changes in Q-value should always be reported. If no rotation is possible, the solution is unique.

The **T** matrix in equation 3-1can be calculated as (Paatero and Hopke, 2008)

$$\mathbf{T}=(\mathbf{G}^{0T}\mathbf{G}^{0})^{-1}\mathbf{G}^{0T}\mathbf{G} \qquad \textbf{(3-2)}$$

Where $\mathbf{G}^0$ is the original solution, **G** is the rotated solution, and $\mathbf{G}^{0T}$ is the transpose of the original solution. Each element in **T** and $\mathbf{T}^{-1}$ gives the strength and direction of rotation. These matrices can be used to determine which factors had strong rotations (Paatero and Hopke, 2008).

3.1 Evaluating Solutions for Rotation

The range of possible solutions should be examined to see if one solution is more physically realistic than the others. This determination can be made by comparing the solution to *a priori* knowledge about the area/data set being modeled. For example, if a source is known to be inactive for a certain period, the contributions from the factor that represents that source should be zero for the inactive time period. If they are not, the user may use the rotational tools to bring the solution in line with their expectations. To control rotations, external information may be used in the model to constrain the solution. These methods are described in Section 3.2. Without *a priori* information, the extent of the rotation that can be explored is limited (Section 3.2.1), but it is more difficult to determine which solution best fits the expectations. (Paatero et al., 2002). One simple way of graphically evaluating solutions for rotation, even without *a priori* information, is to look at g-space plots (Paatero et al., 2005). This method involves examining scatter plots of factor versus factor and looking for edges (Henry, 2003). An edge is evident when at least one side of the group of points in a plot of one source contribution vs. another does not align with the axis, suggesting additional rotations may be considered. However, it should be noted that the presence of edges is not always unwelcome and in fact, an edge may be appropriate for sources in the same wind sector and evaluation of source locations should be considered when evaluating the g-space plots. In such cases, the presence of the edge can used as a validating tool.

3.2 Tools for Exploring and Controlling Rotations

The following methodology for examining rotations was recommended by Paatero, et al. (2002). The user should first run the model with multiple pseudorandom start points and chose one (or a few) of these runs to serve as the starting point for future runs. Then, using the initial factor values as a starting point each time, the model should be run with the various rotational parameters of interest. By using the same starting point each time, variations between the initial run and each additional run can be directly compared as they will not be due to any differences in starting points. Applying the same rotational parameters to a different initial solution can also be useful.

3.2.1 fkey

Fkey should be specified in the '`preprocessing`' section. If fkey is used, the variable `contrun` should be set to 1 and a previous solution must be used as a starting point for iterations. The user can influence specific factor elements with controlled "pulling", allowing the incorporation of a variety of types of *a priori* information into PMF, through several tools. The simplest is fkey, which allows the user to control the solution in several ways. Starting with a known solution, fkey can be used to force a specific factor element in either the **AA** or **BB** matrix to zero, give a lower and upper limit to a factor element, or fix an element to its original value. Because fkey constraints are imposed without regard to the change in the Q-value, they are considered "hard" constraints or pulls. **Table 7** presents the various types of fkey values, their numeric name (for use in scripting), and their interpretation. The name in column 1 of Table 7 can be used in normal scripting; the numeric name in column 2 is used for clarity in some situations (for example, if a matrix of values is provided).

Table 7. *Types of Fkey controls, numerical value in script/Fkey matrix, and interpretation.*

Name/Type of Control	Numeric Name	Interpretation
nolimits	1	Unconstrained
lolimit	0	lolimit constraint (non-negative constraint in PMF)
lohilimits	-1	Lower and upper limit
locked	-5	Fixed to original (input) value
Masked	-6	Fixed to zero

Fkey is useful when a solution does not agree with *a priori* information. The user might notice, for example, that a factor that corresponds to a certain industrial facility has a non-zero contribution on days when it is known that the facility did not operate. Then fkey could be used to force the contributions of that factor to equal zero on the identified days. The user would then base the type of Fkey used on the confidence in the a priori information. If the facility is definitely the only contributor to the factor in question, and the user is certain the facility was not operating at all on given days, a masked fkey would be useful. However, if there are other facilities that could be contributing to the factor, or the days when the facility was not operational are only approximately known, a lohilimit constraint might be more useful.

The fkey method is appropriate when individual elements are being pulled based on very specific *a priori* information. Because there is no limit in the change in Q, the user should have a high level of confidence in the a priori information used.

In ME-2, fkey is included in the script as `AA.fkey` (to refer to the contribution matrix) or `BB.fkey` (to refer to the profile matrix). To specify specific Fkey constraints within the script, the following syntax is used:

```
ZZ.fkey[x,y]=t; ZZ.flow[x,y]=u; ZZ.fhigh[x,y]=v;
```

where ZZ is either the AA or BB matrix, x is the column, y is the row, t is a value from Table 7, u is the lower limit, and v is the upper limit. The flow and fhigh commands are only necessary if lower and/or upper limits are being specified. For example, if it is known that the profile value of species 3 in factor 5 should be around 4, the user might constrain that element to be greater than 2 but less than 6 (conservative limits should be used). The code for this situation would look like:

```
BB.fkey[5,3]=-1; BB.flow[5,3]=2; BB.fhigh[5,3]=6;
```

Fkey should be specified in the '`preprocessing`' section. If fkey is used, the variable `contrun` should be set to 1 and a previous solution must be used as a starting point for iterations. Multiple fkey commands can be used; however, if multiple fkeys are used, it is more difficult to determine the individual impacts on the Q-value of each fkey. For example, if fkey is applied to four factor elements and the Q-value increases by 1000, it is impossible to tell if each fkey is increasing the Q-value by 250 or if one fkey is increasing the Q-value by the entire 1000, or any other combination of increases, without additional analyses.

An fkey matrix can also be used to specify fkey values for all factor elements. In this case, the `contrun` variable should be set to 2 and the fkey matrix is provided in a separate file. The file name of the fkey matrix is included in the script as ##c in the '`defines`' section. In this file, three matrices are included: fkey, flow, and fhigh (in order). If the **AA** matrix is to be used, the values should be in column order (which is the transpose of the **AA** matrix). In this format, the first row of the fkey matrix contains values for the

first column of the **AA** matrix. This is done to simplify the creation of fkey matrices by allowing the use of repeat notation; for example, if there are 100 entries in a column of the **AA** matrix, and each entry is to be unconstrained, the corresponding row in the fkey matrix could be written 100*1, instead of including 100 separate entries, each with a value of 1. The fkey matrix for the **BB** matrix should also be in transposed form. For the **BB** matrix, the use of repeat notation is generally not as important, but the transposed matrix is easier to read. User comments can be included at the end of each line of the fkey matrix file if preceded by a forward slash (/).

In example data set 1 (St. Louis, Section 5.1), fkey is used to constrain elements of the **G** matrix by forcing contributions of the lead source to zero when winds are not from the direction of known sources. When a masked fkey was used, setting all of the contributions from the given wind direction to zero, the Q-value increased by over 100 units, indicating that some or all of the constraints were not reasonable. In this case, more advanced tools (**Section 3.2.3**) may be useful in determining which, if any, of the pulls are reasonable.

Fkey has been used in several studies to obtain more physically realistic results from PMF (Paatero and Hopke, 2008). It is important to always report when fkey is used in a PMF solution as well as the justification for using it.

3.2.2 Auxiliary Equations

Because fkey specifies exact values, or exact limits of values, without regard to the change in the Q-value, it is considered “hard pulling”. If there are factor elements that are only approximately known, techniques that implement “soft pulling” should be used. With soft pulling techniques, appropriate uncertainties can be specified for each pull, defined in auxiliary equations. A smaller value of s indicates a stronger pull. An additional Q term, Q^a is the contribution to the Q-value from the auxiliary equations, defined as (Paatero and Hopke, 2008):

$$Q^a = \sum_{v=1}^{V} Q_v^a = \sum_{v=1}^{V} \left(\frac{r_v}{s_v} \right)^2 \qquad \textbf{(3-3)}$$

where *v* is all auxiliary equations, *r* is the residual values from the auxiliary equations, and *s* is the uncertainty associated with the equation. The r_v is adjusted based on the type of equation. For example, to pull a factor element to a given value, the r_v is equal to $(a_v - f_{pj})^2$, where a_v is the target pull-to value, f_{pj} is the factor element to be pulled, and s_v is the “softness” or uncertainty of the pull, defined by the associated error model and C1 and C3 values provided by the user. Other equations for specific types of auxiliary equations can be found in Paatero and Hopke (2007).

With auxiliary equations, an element or expression of elements is either pulled up maximally, pulled down maximally, or pulled to a target value. Similar to fkey, auxiliary equations should be based on *a priori* information. The simplest type of auxiliary equation is similar to an fkey pull, where a single element is constrained to a given value. The same type of *a priori* information as used to determine an fkey pull would be applicable to an auxiliary equation in this case. The certainty in the *a priori* information should influence the uncertainty parameters that control s_v. Auxiliary equations can also be used to define relationships of elements. For example, a factor may contain both ammonium and sulfate, indicating that it represents secondary sulfate. An auxiliary equation could be used to define the ratio of sulfate to ammonium when the sulfate is fully neutralized. Auxiliary equations could also be used to define relationships between species when source profiles are known.

Because the user can include a measure of their confidence in the *a priori* information by adjusting the uncertainty (or “softness”) of the equation, auxiliary equations are appropriate when the user has uncertain or approximate *a priori* information. Auxiliary equations are also able to pull expressions of elements, which fkey cannot do. However, like fkey, if multiple constraints are applied at once, it is not immediately obvious to the user how much of an increase in the Q-value each constraint is responsible for without additional analyses.

3.2.3 Implementing Auxiliary Equations

Soft pulling using auxiliary equations is included in the script in the '`equations`' section. There are multiple ways to implement auxiliary equations in the script. In general, an equation consists of an array, a target value, C1 and C3 variables, an error model code, and the elements or expression of elements to be used. For example, the code shown below defines an auxiliary equation that pulls the fifth species in the first factor profile toward zero. The species and factor are specified in `BB[1,5]`, the pull to value is the '`Data`' variable, and the error model (`errmod`), `C1`, and `C3` variables are defined as described above. The term '`equ>`' signifies the beginning of an equation and '`equ!`' signifies the end of an equation. The '`term`' command defines the parts of the equation, with a '`pos`' or '`neg`' indicator that either adds or subtracts the value, respectively. Multiple terms can be included in an equation. When an expression is defined, it is generally simplest for the model to arrange the relationship of terms so that the sum is always pulled to 0. The 'AUXAR' array is an aux data array for auxiliary equations, defined by the user in the 'defines' section of the script ('`defarr auxdata AUXAR[2,np]`').

```
equ>
   AUXAR[2,5], Data=0, C1=normc1, C3=0.05, errmod=-16;
      term>
         pos; @BB[1,5];
      term!;
equ!;
```

A special type of auxiliary equation, referred to as autopull, has been developed to allow the user to put limits on the change in the Q-value associated with each pull. Using autopull equations, the user can implement many pulls at once and provide a Q-limit for each pull. Using this method, the user does not have to adjust the constraints and re-run the model multiple times to find the ideal solution. Autopull equations can be used for any of the constraints described above. The auxiliary Q value is adjusted to account for the limit to the change in Q. For example, to pull a factor element up maximally, *a* and *s* are defined as (Paatero and Hopke, 2008):

$$a = f^{initial} + f^{\exp ectedstep}$$

$$s = \frac{f^{\exp ectedstep}}{\sqrt{dQ}} = \frac{a - f^{initial}}{\sqrt{dQ}}$$

where *dQ* is the user defined limit on the change in Q. After each iteration, if convergence is not achieved, *a* is adjusted as:

$$a = f^{initial} + \max[2(f - f^{initial}), 0.1(f^{\exp ectedstep}), 0.5(a - f^{initial})]$$

Similar equations for pulling a factor element down maximally and pulling to a given value are defined in Paatero and Hopke (2008).

Autopull equations are specified in the '`equations`' section like other auxiliary equations. The construction of the equation is also similar to auxiliary equations. Additionally, three different error models can be specified in an autopull equation: -20 indicates pull the element up maximally, -21 indicates pull the element down maximally, and -22 indicates pull the element to a target value. The basic format for this type of equation is

```
equ>
   AUTO[ii], errmod=-22;
   term>
      pos; @BB[1,5];
   term!
```

```
equ!

%Q-main limit      Expected change    Expected or target value
Auto.aux1[ii]=10;  AUTO.aux2[ii]=1.0;      AUTO.aux3[ii]=0;
```

In this example, the fifth element (in this case, the fifth species) of the first factor is being pulled to zero, as in the previous example. The element is again defined by the '`term`' command, and multiple terms can be included in an autopull equation. The error model of -22 indicates to pull up or down to a target, which is defined by `AUTO.aux3[ii]` as zero. An error model of -21 could also have been used to pull the element down as far as possible. The expected change in the factor element is 1 (`AUTO.aux2[ii]`), to indicate the user expects to pull the element with only a small change (ones of Q) in Q, and the Q-main limit (`AUTO.aux1[ii]`) indicates the total change in the Q-value that is allowed to achieve the constraint.

In the second example data set for Craig (**Section 5.2**), autopull equations were used to pull given elements of the contributions of the steel factor to zero on days when the steel facility was known not to be operating. Autopull was used with an error model of -21 (pull down maximally) with change in Q limits of 10, 50, and 100 per pull (22 pulls total). Five of the elements were pulled to zero using the limit of 10, with an increase in Q of less than 100. With either the Q limit of 50 or the Q limit of 100, 19 elements were pulled to zero and the Q increase was around 100 units. Additional analyses varying the Q-limit on individual pulls could be used to determine if some pulls were more compatible with the solution than others. In this example, the three factor elements that the model was not able to pull to zero are incompatible with the solution.

Auxiliary equations have been used in several studies to incorporate information from source profiles into ME-2. For example, Rizzo and Scheff (2008) used the "target shape method", which uses auxiliary equations to incorporate profile information from CMB results by pulling specific factor profile elements to both zero and non-zero values. Lanz et al., 2008 and (Paatero et al., 2005) used a similar "hybrid method" approach to incorporate entire factor profiles based on *a priori* information. In both cases, the results provided more information than was provided by a PMF analysis alone.

3.2.4 Fpeak

Fpeak is used to explore rotations in ME. The Fpeak matrix is a *p* x *p* matrix with all off-diagonals set to a non-zero Fpeak value. A positive value of Fpeak corresponds to adding **AA** columns together and subtracting **BB** rows from each other. A negative value corresponds to subtracting **AA** columns and adding **BB** rows. By trying a range of Fpeak values, resulting solutions can for evaluated for the change in the factor profiles, contributions, and change in the Q-value. In addition, evaluate G-space plots to determine if the pair-wise relationship between the contributions is reasonable based on knowledge of the airshed. In ME-2, Fpeak is implemented by using an enhanced object function. Two additional terms, Q^n and Q^p, are used to normalize the rows of the **BB** matrix and pull all the elements of the G matrix to zero, respectively Paatero and Hopke (2008). These additional terms are implemented in the program as (Paatero and Hopke, 2008a):

$$\bar{g}_{ip} + d_{ip} = g_{ip} + r_v$$

$$Q_v^a = \frac{(\bar{g}_{ip} + d_{ip} - g_{ip})^2}{s_v^2} \qquad \textbf{(3-4)}$$

where a bar over a factor element indicates the value of that element in the previous rotation. **The strength of Fpeak is not comparable in PMF2 and ME-2. Paatero and Hopke (2004) showed that in order to use a comparable Fpeak in ME-2, the value used in PMF2 should be multiplied by 5.**

Global Fpeak is easily implemented through EPA PMF; free-standing Fpeak scripts are also available. It is recommended that Fpeak is applied through the EPA PMF GUI or script. To use Fpeak outside of EPA PMF, the iniparams file described in section 4.1 should be used. The parameter `dofpeak` should be set to 1, indicating a global Fpeak parameter will be applied, which should be included by setting `fp` equal to the desired strength of Fpeak. In this case, all of the non-diagonal values in the Fpeak matrix are set to `fp`. In ME-2, setting `fp` to zero does not have any meaning. If `dofpeak` is set to a value of 2, an Fpeak matrix with non-diagonal values set to the desired Fpeak strengths must be provided.

Fpeak has been used in several studies to explore a PMF solution. For example, Zhao and Hopke (2004) used Fpeak to explore the solution space and reported a final solution using an Fpeak of zero (because they were using PMF2, an Fpeak of zero does have some impact on the rotation of the solution). Kim et al., 2006 and (EPA, 2008) also used an Fpeak of zero in their reported PMF solution. It is recommended that Fpeak be used as an exploratory tool, and any value of Fpeak used in a final solution, as well as the justification for using that value, should be reported with the results.

3.2.5 Summary of Rotational Tools

Table 8 summarizes the various rotational tools described above. Example syntax is provided for reference; the description of each tool above should be used to determine the appropriate type of constraint for a particular task.

Table 8. *Summary of rotational tools available in ME-2*

	Command	Abilities	Syntax
Hard Pulling	BB.fkey	Masked	`BB.Fkey=-6`
		Locked	`BB.Fkey=-5`
		Set limits	`BB.Fkey=-1`
	AA.fkey	Masked	`AA.Fkey=-6`
		Locked	`AA.Fkey=-5`
		Set limits	`AA.Fkey=-1`
Soft Pulling	Autopull	Pull up maximally	`Errmod=-20` and code from above
		Pull down maximally	`Errmod=-21` and code from above
		Pull to a target value	`Errmod=-22` and code from above
		Pull ratios/equations of elements	Appropriate `errmod` and code from above
	Auxiliary Equations	Pull up maximally	See code above for examples
		Pull down maximally	
		Pull to a target value	
		Pull ratios/equations of elements	

4.0 USING CONTROL FILES FROM EPA PMF TO PERFORM TASKS WITH ME-2

If the user already has a copy of EPA PMF 3.0 or later and does not wish to purchase a fully editable version of ME-2, they can perform any of the tasks described in this document using the script, key, library file and ME-2 executable that are part of EPA PMF. When EPA PMF is installed, the default location for these files is C:\Program Files\EPA PMF. Before running ME-2, the user should copy these files (e.g, bs_4s2.ini, me2key.key, and me2wopt.exe) to a separate folder (e.g., C:\EPA_ME2) to prevent any interaction with EPA PMF. A description of specific steps the user can follow to employ ME-2 in this manner is provided in Appendix A.

The user may not directly alter the EPA PMF script; all changes and additions to the script are performed via two control files: iniparams.txt, which assigns values to the variables described in Section 2, and moreparams.txt, which generates the rotational code and equations described in Section 3. The names of these files must not be altered as they are referenced specifically by the EPA PMF script. The iniparams file is automatically deleted by ME-2 after running, so users should routinely make a copy of this file under a more detailed name, e.g., iniparams_StL_AAautopull10.

Once the user has written the two control files, and saved them to the same location as the other ME-2 files, ME-2 is run through the command prompt as described in section 2.2. At the command prompt, the user should enter 'me2wopt.exe PMFbs_2.ini' to run the program.

4.1 Iniparams

The iniparams.txt file allows the user to change many of the variables associated with performing PMF. A default version of the iniparams file is provided with EPA PMF (copied below). If the user wishes to use the EPA PMF default as a template to generate their own files, it should be copied to the ME-2 folder as described above. Case study iniparams are provided with this document and can also be used as a template for new iniparams files.

In the iniparams file, comments, which are ignored by the program, are preceded by a forward slash (/). Any text from the slash to the end of the line will not affect how the program runs. Comments are provided above each entry to indicate which variable is referenced. The variable names are the same as they are in the script; most were described in Section 2. Additional variables included in the iniparams file are:

- `iniparamsv`: version of iniparams used. This will not change unless a new version of EPA PMF script is used.
- `bsinitfact, bsmode, dobspull, pullc1, readbscnts`: used to control bootstrapping runs; refer to the EPA PMF User's Guide (2002) for more information.
- `numpf, numynpf, maxpfdim, modelc1, modelc3, pfpullc1, pfsmooc1`: used to control the parametric model, see Paatero and Hopke (2008) for more information.
- `n1, n2, np`: used to provide the number of samples, number of species, and number of factors, respectively.

```
/ iniparams.txt for PMF_bs_4s2.ini
/ Parameters for the script PMF_bs_4s2.ini
/ (in same order as parameters appear in the script)
/ Update iniparamv only when the format of file is updated,
/ e.g. when new parameters are added.
/ Copy or rename this file to file iniparams.txt

/ iniparamv,
  261107
```

```
/ robust, posoutdist, negoutdist, precmode, numtasks, numoldsol,
    1         4           4           20        5           0
/ bsinitfact, bsmode, simu,
    1         11      0
/  contrun, dobspull, pullc1, readbscnts, samplevari,
     0     0       1.5       0             1
/ alowlim, normc1, acbmodel,
   -0.20     0.0       0
/ seed1, seed2, seed3, seed4, seed5,
    78     15     24     89     49
/  n1,   n2,   np,   c1,   c3,    em,
  186,   23,   7,   0,   0,   -12
/  numpf  numynpf  maxpfdim  modelc1  modelc3  pfpullc1  pfsmooc1
    0        0         20      1.2      0.35      1.1     0.50
/ naming for input files:
/ main data file,  previous results  (type here full names),
'PMFData.txt'    'PMF_ab_base.dat'
/ naming for output (many files, type here main part of file name only)
'PMF_ab_base'

/ doresort,  dofpeak,  fp
    1         1        0
```

PMF_bs_4s2.ini is the ini file that is used in EPA PMF v3.0. For the examples, this ini file has been renamed to PMF_bs2 to make the file name consistent for EPA PMF programming. Reference PMF_bs_4s2.ini and this EPA report in applications that use the ME program provided with EPA PMF 3.0.

4.2 Moreparams

All of the rotational tools described in Section 3 are controlled by the moreparams.txt file. The version of this file used for the first example data set is provided with this document and can be used as a template for generating new moreparams files. As with the iniparams file, comments are signified with a forward slash (/); any text after the forward slash is ignored by the program. It is recommended that comments be included to define the pulling done on each line of the moreparams file.

Before writing the moreparams file, the user should go through the logic described in Section 3 to decide which type of pulling should be used and, if equations are to be used, how they should be constructed. Only after the user has been through this process and decided exactly how they want to perform the pulling operations should the moreparams file be generated.

The first row in the moreparams file defines how many and what type of pulls are going to be included as well as the maximum number of constants that can be used in the equation. There are five entries in this row (labeled with a commented line above in the example below):

- `numAApulls-` number of pulling equations using elements of the AA (contributions) matrix
- `numBBpulls-` number of pulling equations using elements of the BB (profile) matrix
- `AAfkeyinput-` number of AA.fkey pulls that will be defined
- `BBfkeyinput-` number of BB.fkey pulls that will be defined
- `numconstweights-` the total number of constants (across all equations) that can be defined: generally 1000 is an appropriate number for a small number of equations, hundreds of equations or more would require an increase in this value

The next set of rows defines the pulling equations specified in `numAApulls` and `numBBpulls`. If no pulling equations are specified (`numAApulls` = 0 and `numBBpulls` = 0), this set of rows is omitted. As an example, if `numAApulls` is set to 4 and `numBBpulls` is set to 5, the next nine rows will define each of these equations. The equations using elements the **AA** matrix must be first, followed by the equations using elements of the **BB** matrix. The user should first decide on the type of equation and the elements of the equation as discussed in section 3.2.2. Each row will have at least 8 entries, as defined in **Table 9**. The first entry is always the error model code (see Table 5). The next 6 entries are dependent on what the error model code is. The last entry is always a 0 to signify the end of the equation.

Table 9. *Description of entries in moreparams file.*

Error Model Code	Entry 2	Entry 3	Entry 4	Entry 5	Entry 6	Entry 7	Entry 8
-20	Expected change in value of the element (e.g., species or contribution) being pulled	0 (place-holder)	Limit in the change in the Q-value (dQ)	Row index for element • if *AA matrix, sample number/date;* • *if BB matrix, factor number*	Column index for element • *if AA matrix, factor number;* • *if BB matrix, species*	Weighting coefficient for element	0 (end of equa-tion)
-21	Expected change in value of the element being pulled	0 (place-holder)	Limit in the change in the Q-value (dQ)	Row index for element • *if AA matrix, sample number/date;* • *if BB matrix, factor number*	Column index for element • *if AA matrix, factor number;* • *if BB matrix, species*	Weighting coefficient for element	0 (end of equa-tion)
-22	Expected change in value of the element being pulled	Target value of element	Limit in the change in the Q-value (dQ)	Row index for element • *if AA matrix, sample number/date;* • *if BB matrix, factor number*	Column index for element • *if AA matrix, factor number;* • *if BB matrix, species*	Weighting coefficient for element	0 (end of equa-tion)
Other (from Table 6)	0 (place-holder)	Target value of equation	'softness' of equation; smaller values of s correspond to strong pulls	Row index for element • *if AA matrix, sample number/date;* • *if BB matrix, factor number*	Column index for element • *if AA matrix, factor number;* • *if BB matrix, species*	Weighting coefficient for element	0 (end of equa-tion)

After the pulling equations are defined (or, if no `numAApulls` = 0 and `numBBpulls` = 0, after the first row), control values for AA.fkey and BB.fkey are provided. These control values are presented in three matrices for each of the AA and BB matrices. The first matrix contains the AA.fkey values, which defines the type of fkey that will be performed (Table 7); the second matrix contains the AA.flow values, which

specify a lower limit for each element; and the third matrix contains the AA.fhigh values, which specify an upper limit for each element. The AA.flow values are by default set to -0.1 in the EPA PMF script. Any entries in the AA.flow matrix will overwrite the default values. To avoid overwriting the default values, the user should not put any value in the AA.flow matrix (see example below). If the user does not wish to include upper limits in AA.fhigh, they should use a value of 0 (see example below). Each matrix should be transposed, which allows the use of repeat notation for elements of the AA matrix. Control values for the BB matrix are provided next as three matrices in the same order: BB.fkey, BB.flow, BB.fhigh. If `AAfkeyinput` and/or `BBfkeyinput` are zero, the appropriate matrices should be omitted. When pulling in the **BB** matrix, recall from section 2.2.4 that it is transposed.

Two examples of moreparams files and an interpretation of each part of the file are provided below. The first example contains four pulling equations; the second example contains control values for the AA matrix, which contains 200 elements in this example. Each entry of each row is defined in the interpretation, separated by a semi-colon. The line numbers and interpretation are for guidance purposes in this document only and should not be included in an actual moreparams file.

```
1  /numAApulls     numBBpulls     AAfkeyinput     BBfkeyinput     numconstweights
       0              4              0               0              1000
   /3 f-pulling equations will follow
2  -21   0.5   0   10   3   7   1.0   0 / pull element 3 in factor 7 down, dQ<10
3  -20   1.5   0   20   5   2   1.0   0 / pull element 5 in factor 5 up, dQ<20
4  -22   3     6   15   4   6   1.0   0 / pull element 4 in factor 6 to 7, dQ<15
5  -12   0     0   0.1  2   5   1.0   4   5   -1.0   0 / element 2 = element 4 in factor 5, s=0.1
```

Interpretation:

1: Perform 0 pulling equations in the AA matrix; 4 equations pulling elements of the BB matrix; do not perform AA.fkey pulls; do not perform BB.fkey pulls; maximum number of constants that will be provided is 1000

2: Perform an equation to pull down maximally; the expected change in the element value is 0.5; value of 0 as a place holder; the limit on the change in Q-value is 10; the element (row of BB matrix) to pull is 3; the factor this pull applies to is 7; the weight of this element is 1.0 (i.e., do not downweight/upweight); terminating 0

3: Perform an equation to pull up maximally; the expected change in the element value is 1.5; value of 0 as a place holder; the limit on the change in Q-value is 20; the element (row of BB matrix) to pull is 5; the factor this pull applies to is 2; the weight of this element is 1.0 (i.e., do not downweight/upweight); terminating 0

4. Perform an equation to pull to a target value; the expected change in the element value is 3; the target value is 7; the limit on the change in Q-value is 15; the element (row of BB matrix) to pull is 4; the factor this pull applies to is 6; the weight of this element is 1.0 (i.e., do not downweight/upweight); terminating 0

5. Perform an equation that will be defined; value of 0 as a placeholder; the target value of the equation is 0; the s is set to 0.1 to indicate a strong pull; the first term of the equation is element (row of BB matrix) 2; the factor this element is in is 5; the weight of this element is 1.0 (i.e., do not downweight/upweight); the second term of the euquation is element (row of BB matrix) 4; the factor this element is in is 5; the weight of this element is -1.0 (i.e., subtract the second term from the first term); terminating 0

```
   /numAApulls     numBBpulls     AAfkeyinput     BBfkeyinput
1  numconstweights
        0              0              1               0              1000
   /3 matrices defining the pulls on the AA matrix will folow
   /AA.fkey
   200*0
   100*0, 25*-6, 75*0
   200*0
2  50*0, 15*-5, 135*0
   150*0, 30*-1, 20*0
   200*0
   200*0
   /AA.flow
   200*,
   200*,
   200*,
3  200*,
   150*, 30*0.5, 20*,
   200*,
   200*,
   /AA.fhigh
   200*0
   200*0
   200*0
4  200*0
   150*, 30*2.0, 20*
   200*0
   200*0
```

Interpretation:

1: Perform 0 pulling equations in the AA matrix; 0 equations pulling elements of the BB matrix; do perform fkey pulls on the AA matrix; do not perform fkey pulls on the BB matrix; maximum number of constants that will be provided is 1000

2: AA.fkey matrix: first factor low limit constraint (cannot go negative); second factor: first 100 elements low limit constraint (=0), next 25 elements masked to 0 (=-6), next 75 elements low limit constraint (=0); third factor low limit constraint (=0); fourth factor: first 50 elements low limit constraint (=0), next 15 elements set to input value (=-5), next 135 elements only low limit constraint (=0); fifth factor: first 150 elements low limit constraint (=0), next 30 elements high and low limits will be defined (=-1), next 20 elements low limit constraint (=0); sixth and seventh factors: low limit constraint (=0)

3: AA.flow matrix: Do not change lower limit for any elements in any factor except five; factor five: first 150 elements do not change low limit, next 30 elements set low limit to 0.5, last 20 elements do not change low limit

4. AA.fhigh matrix: Do not change upper limit for any elements in any factor except five; factor five: first 150 elements do not change upper limit, next 30 elements set upper limit to 2.0, last 20 elements do not change upper limit

5.0 APPLYING ROTATIONAL TOOLS—EXAMPLES

The following sections present example ME-2 analyses of three types of data sets. In each example, various types of *a priori* information are included in ME-2 via one of the rotational tools described in Section 3. The first example uses hourly fine PM data from the St. Louis – Midwest Supersite in East St. Louis, IL. The second uses 24-hr $PM_{2.5}$ data from a Speciated Trends Network (STN) site in Cleveland, Ohio. The final example uses volatile organic compounds (VOCs) data from a Photochemical Assessment Monitoring Stations (PAMS) site in Baton Rouge, Louisiana.

Reference scripts, data sets, and results files are provided for each example analysis in zip files: St Louis example, GT Craig example, and Baton Rouge example. In addition, iniparams.txt and moreparams.txt files are provided for the examples. Following the examples, the user can exactly recreate the output provided using files provided with EPA PMF 3.0.

To run examples, the user should create a folder on their machine with the ME-2 executable and key. Different files are moved depending on whether the Individual License (IL) or public license (PL) versions of ME are being used. For example, if using the IL create the folder C:\Program Files\ME2, and for PL create the folder C:\Program Files\EPAME2. For the IL, the exe and key will be provided with the license. For the PL, copy the exe and key from C:\Program Files\EPA PMF 3.0.

To run an example with the IL, the user should copy the following files into their local ME-2 folder: data (e.g., StLouis_Data.txt) and script (e.g., StLouis_base.ini). Next, check that the path listed in the first line of the script file matches the directory of the local ME-2 folder created by the user; if not, adjust this path in the script to the local ME-2 folder. With the PL, the user should copy the data (e.g., StLouisData.txt), script file (e.g., PMF_bs2.ini from C:\Program Files\EPA PMF 3.0) plus the control file (e.g., iniparams_StLouis.txt) into their EPAME2 folder.

All the graphs and tables shown in this section were generated in MS Excel using the output files provided by ME-2. Summary contributions and contributions (Figures 8, 13, & 16) were generated using the output from the PL version of ME. References to the file used for each graph and table are provided in the first example. Step-by-step instructions are provided in Appendix A.

5.1 St. Louis Supersite Fine PM

In the data folder accompanying this document, data files are provided so a user can recreate the results presented here using the version of ME provided with EPA PMF 3.0, and use these examples to build their own files for their own data (St Louis IL ME2 Files). For users who wish to use the PL version ME-2 provided with EPA PMF 3.0, a folder of control files is provided (St Louis PL ME2 Files). In this folder are the iniparams and moreparams files than can be used with the public EPA PMF script. To follow the examples provided here using autopull in **AA**, the user can adjust the allowed change in Q in the autopullAA ini or iniparams/moreparams files. More detailed instructions are provided in Appendix A. The discussion in this section references the ini files, but these examples can also be run using the provided iniparams and moreparams files.

ME-2 was run on a data set of 13 species, sampled hourly during 6/01, 11/01, and 3/02, at the East St. Louis site (420 samples). Uncertainty estimates by species and sample were provided by the lab. Samples below the detection limit were given an uncertainty of 5/6 the detection limit, missing samples were given an uncertainty of 4 times the median concentration, and samples above the detection limit were given an uncertainty of one-third the detection limit plus a sample-specific laboratory uncertainty. This data set was chosen to illustrate adding constraints to the PMF model based on two types of *a priori* information: wind direction, which was used to identify when winds were/were not from known source regions (used to constrain the **AA** matrix) and known source profiles (used to constrain the **BB** matrix). The two examples detailed below explore pulling contributions of the lead factor to zero on hours when the wind is known not to be from the direction of the lead sources and using a steel profile to pull a ratio of iron (Fe) to manganese (Mn) to a set value for the steel manufacturing factor in the **BB** matrix (see **Figure 2** for site and source locations).

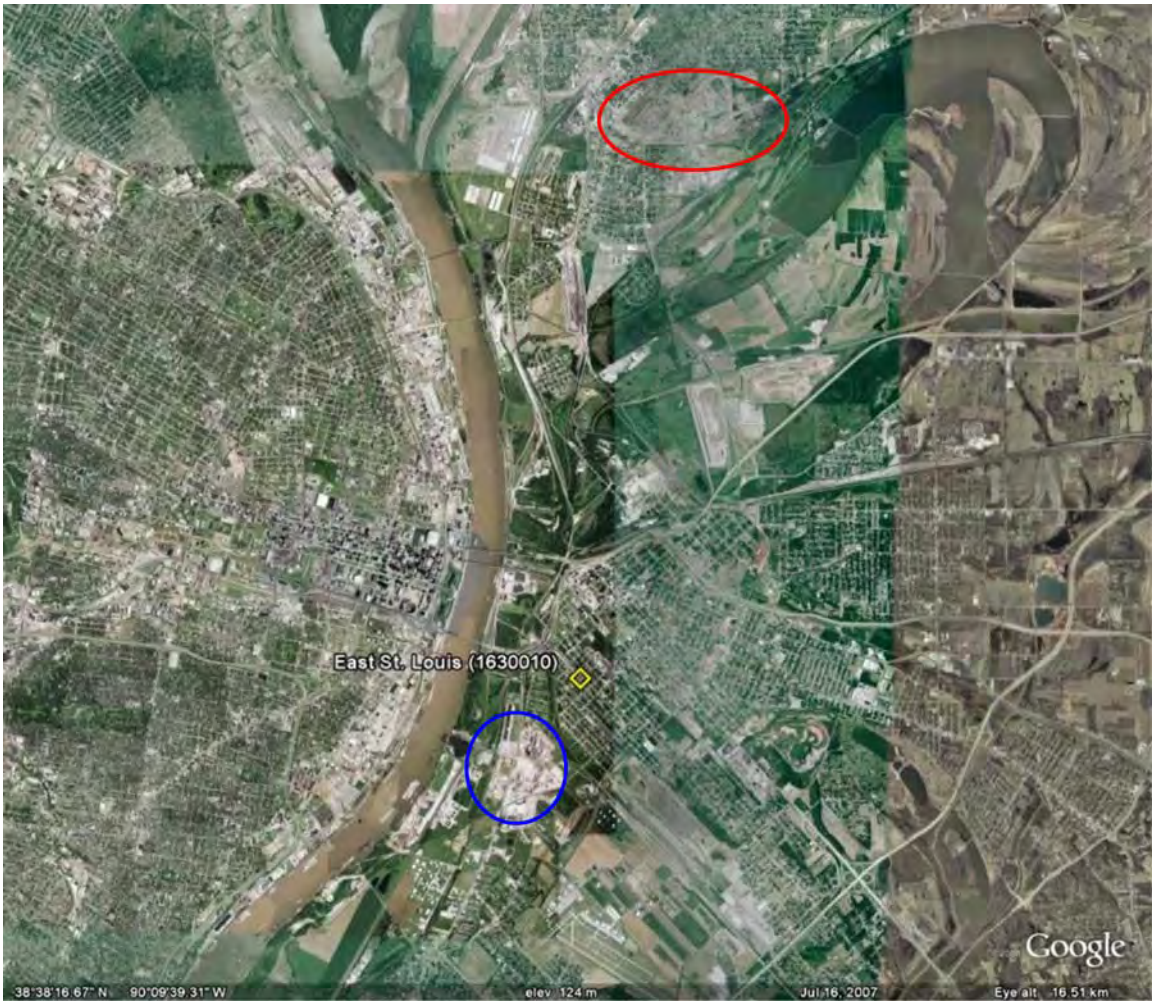


Figure 2. *Image of St. Louis supersite (yellow diamond); source of lead emissions circled in blue, major steel facility circled in red.*

5.1.1 Input Parameters and Base Solution

Most variables were left at their default value, and the script was simply adjusted for the St. Louis data set (reference files StLouis_base.ini or iniparams_StLouis.txt for public script). Different numbers of factors and values of C3 were explored; a final solution of 7 factors with a C3 value of 10% was chosen as the most physically realistic solution (based on known sources in the area). The 7 factors identified (noted by key species) were manganese/iron (Mn_Fe), nitrate (NO_3), lead (Pb), carbon (organic and elemental), copper (Cu), zinc (Zn), and sulfate (SO_4). The annotated code excerpt shown in **Figure 3** (from the '`defines`' section of the script) details the variables used in the base run. Blue comments describe the variables that are important in each run, red comments indicate the parameter is data set specific and should be changed for each data set. In this example, 20 runs from random starting points were conducted to look for the global minimum Q-value and to avoid a local minimum Q-value. The minimum Q-value was consistently around 1163 regardless of the start point, implying this is a global, not local, minimum. The 15th run (of the 20 initial random start runs) was chosen as the starting point for the continuation runs.

```
version=1.203;   % older than 1.203 will NOT work right!
monitor=5;

robust=1;

posoutdist=4; negoutdist=4;

missdatlim=-990;
bdlneg=0;

convtests
  0.100,       20,        300,        0,      0,    0.0001,   %level 1
  0.010,       40,        800,        0,      0,    0.0001,   %level 2
  0.0002,      60,       2000,        0,      0,    0.00002;  %level 3
% deltaQ   consecut.    max cumul.     not     not    gg2 norm
%   test      steps     step count    used    used     test

cgresets  10,  80,  1,  1,  1,  1;

precmode=15;
numtasks=20;

variables
  'numoldsol'=15,
  'alowlim'=-0.1,
  'blowlim'=0.0,
  'seed1'=483,
  'normc1'=0.01,
  'contrun'=0;

if> (contrun>0);
  numtasks=1; goodstart=1;
if!;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
##d='StLouisData.txt';
##c='apriori.txt';
##p='StLouis_Base.dat';
##m='StLouis_Base';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   number of rows    number of columns    number of factors
       nl=420;             n2=13;              np=7;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  std-dev coefficients and errormodel code for the main equations
      c1=0.0;   c2=0.0;   c3=0.10;     em=-14;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Downweight outliers (+/- 4 scaled residual)

Missing and below detection limit data not included in dataset, these values left at default

Convergence criteria: change in Q less than 0.002 over 60 iterations; 2000 iteration maximum

Perform one run

Number of old solution (not necessary when starting from random points)

Lower limit of G elements (contributions of factors)

Lower limit of F elements (species in profiles)

Starting point for generating random solution

C1 variable for auxiliary equations

Starting from random solution; change to 1 when starting from known solution, 2 if starting from a known solution and constraining some elements using input in a separate file (apriori.txt, below)

File with input data (concentration and uncertainty)

File with matrix of a priori informationl; used only if contrun=2

File with previous solution; used only if continuation run (contrun = 1 or 2)

Prefix for output files

Dimensions of model

Defining error model and parameters, C3 was increased in this example to account for extra modeling uncertainty

Figure 3. *Annotated code excerpt from ini file.*

5.1.2 Pulling Elements of G (AA) Matrix

To determine which elements of the **AA** matrix to constrain, the locations of nearby lead sources relative to the site were examined. No sources were obvious directly to the east of the site (see **Figure 2**). Based on this, all samples with wind direction between 67.5° and 112.5° should have low contributions from the lead factor. There are 25 samples in this wind sector with non-stagnant winds. **Figure 4** shows the G-space plots of the Pb factor, with the 25 samples from the east in red. Most of these samples are already close to zero, indicating the solution is already rotated in this direction (most apparent in the NO_3, OC_EC, and SO_4 graphs). However, the larger samples are not near the same edge and will likely not be easily rotated.

Several methods of pulling the contributions down were compared, including: masked AA.fkey (forcing elements to equal zero) and autopull of the elements to zero (with various Q limits). These methods and the corresponding files to use to recreate the results are discussed below. Different limits of Q can be explored by changing the value for the pulls in the ini or moreparams files.

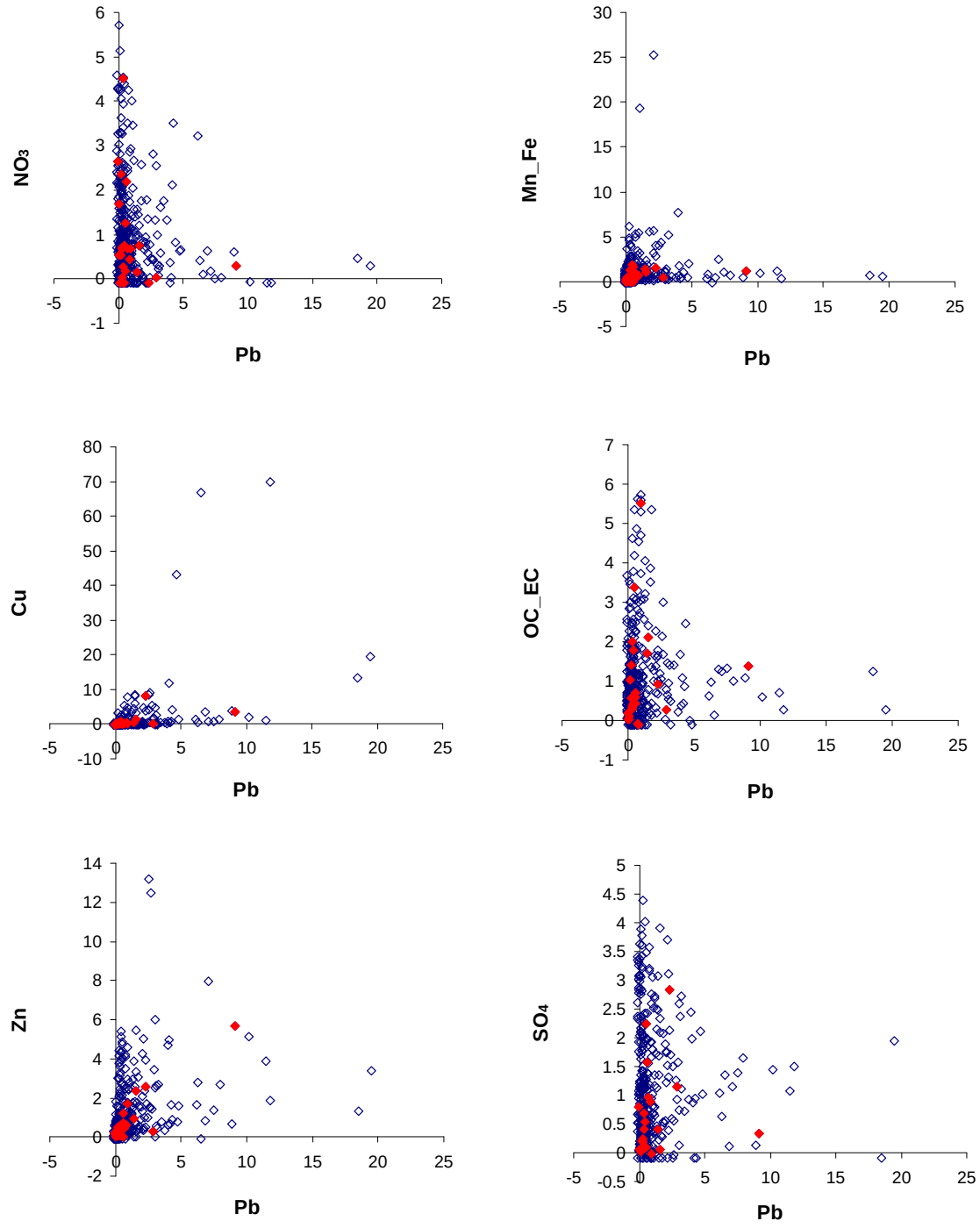


Figure 4. G-space plots of the Pb factor versus the other 6 factors resolved for the St. Louis base run (data from StLouis_base.dat).

AA.fkey

A masked AA.fkey was applied to this data set. To use AA.fkey, four variables were changed in the '`defines`' section of the script (reference files StLouis_masked.ini and moreparams_StLouisMasked.txt):

- `contrun=1` to indicate that this run is starting from a prior ME-2 solution,
- `numoldsol=1` to indicate that the first solution in the file of output from previous runs is to be used as the starting point for this run,
- `##p='StLouis_base.dat'` to point ME-2 to the file with the previous solutions, and
- `##m='StLouis_Gkey'` to tell ME-2 the prefix for the output from this run.

To define each pull, the command:

'`AA.fkey[i,j]=-6;`'

was used in the '`preproc`' section of the script, where i is the sample number and j is the column of the Pb factor. For this example, 25 fkey's were defined. For each one, the j value was constant (they all applied to the Pb factor) and the i value indicated each of the 25 samples where the wind was from the east. Although this guarantees that the element will be equal to 0, the Q-value is allowed to increase as much as possible to accommodate the constraint. Additionally, because of the non-negativity constraint, it is possible that accommodating the additional constraints will cause other factor elements to change considerably.

When the masked AA.fkey was used, all of the pulled elements were zero in the modeling results, as required by using masking; however, the resulting Q-value increased by over 100 units (**Table 11**). When using a masked fkey, the user should be very certain that the a priori information is accurate. In this example, there could be other influences on the Pb factor that are causing it to have a non-zero contribution even when wind is not from the direction of known sources (for example, stagnant conditions which allow concentrations to accumulate/increase in the area). It is possible that for some of the pulled samples, the wind direction is the most important influence on the Pb concentrations, but for other samples, other influences are more important. Pulling the samples with other influences would result in a large change in the Q-value. However, it is impossible to determine what is causing the changes in the Q-value without additional analysis.

Table 11. *Q-values and number of pulls that reached the target value for each type of rotational tool used. (Data from StLouis_base.txt, StLouis_masked.txt, StLouis_autopull.txt.) Results using iniparams and moreparams may vary slightly from those below.*

Method	Q	Qmain	Qaux	Number of pulls that reached target (out of 25)
Initial Run	1118.1	1118.1	0.0	--
AA.gkey-masked	1344	1344	0.0	25
Autopull- 10	1245	1181	63	5
Autopull- 50	1654	1351	303	18
Autopull- 100	1983	1372	610	25

Autopull

Autopull equations were used to pull the factor elements to a target value of -0.1, the lower limit of the **AA** matrix in this example (reference files StLouis_autopull_AA.ini and moreparams_StLouisAutoAA.txt). The example provided with the moreparams file is only an autopull

to 10. Autopull allows limits on the Q-value, i.e., a constraint will only be implemented if the Q-value does not increase more than the user defined limit. In this example, 3 different Q-limits were compared: 10, 50, and 100. The Q-limit is the change in Q that was allowed for each autopull equation. A larger Q-limit makes it more likely that the target value will be reached. The same variables were changed as in the AA.fkey example (`contrun=1`, `numoldsol=1`, `##p='StLouis_base.dat'`, `##m='StLouis_Gkey'`). The following commands were used in the '`equations`' section of the script to develop the autopull equations:

```
ii+1;
equ>
      AUTO[ii], errmod=-21;
      term>
            pos; @AA[i,j];
      term!;
equ!;
AUTO.aux1[ii]=50; AUTO.aux3[ii]=-0.1;
```

Where ii was previously defined as 0, i is the sample number and j is the column of the Pb factor. Each term in the equation is identified as positive or negative by using the phrase '`pos;`' or '`neg;`'. In this example (and in general for single-term equations such as this), the term that consists of element i,j in the contribution array is positive. As with the AA.fkey commands, twenty-five autopull equations, each for a different sample number (i), were included. In the three separate autopull runs, only the `AUTO.aux1[ii]` value was changed (10,50,100).

With the first Q-limit of 10, only 5 of the elements were pulled to the lower limit and the Q-value increased almost as much as with the AA.fkey- masked method (Table 12). With a Q-limit of 50, 18 elements were pulled to the lower limit, but the Q main increase was over 200 units. With a Q-limit of 100, all of the elements were pulled to the lower limit, and the Q-main also increased by over 200 units. **Figure 5** shows that the elements that started with the lowest values were pulled to zero by most of the constraints, and the elements with the largest start values were only pulled to zero when the Q-value was allowed to change by 100 units for each pull. This supports the hypothesis that some of the elements (specifically the ones with higher start values) are not influenced by just the wind direction.

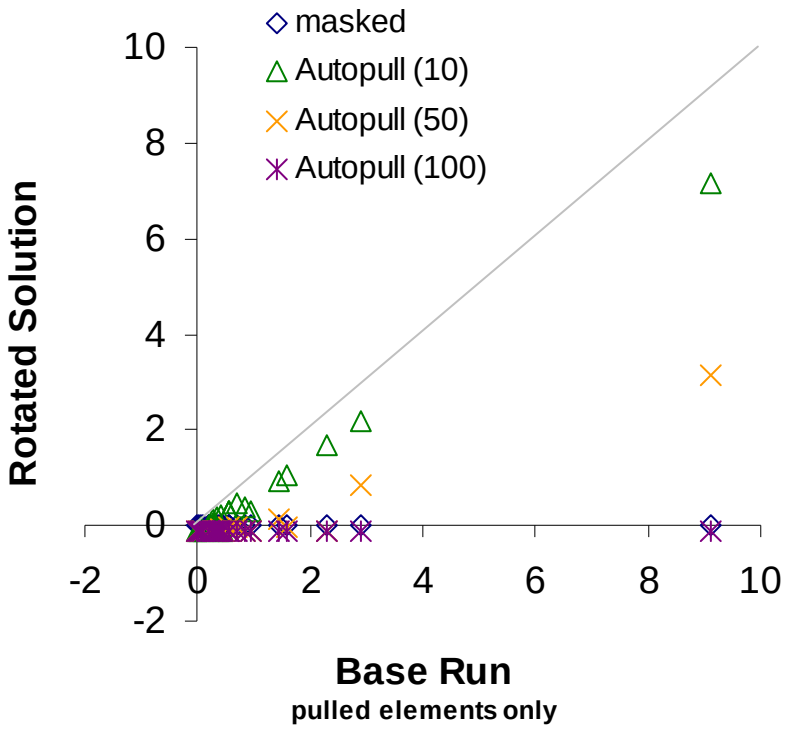


Figure 5. *Comparison of lead factor base run (x-axis) and each rotated solution (y-axis). Only pulled values are shown.* (Data from StLouis_base.dat, StLouis_masked.dat, StLouis_autopull.dat.)

5.1.3 Evaluation of results

Although the information in Table 11 can give an indication of whether a particular set of constraints "worked" or not, more analysis is necessary to understand the full impact of the constraints. Other factor elements will likely be impacted, and the interpretation of the solution could change with some constraints. For this example, contributions and profiles of each factor and total mass apportioned to each factor were examined for each constrained run.

Several factors had noticeable changes in their contributions, particularly NO_3, Pb, and OC_EC. Scatter plots of contributions of the initial run versus each constrained run showed lots of scatter around the one-to-one line for the NO_3 and OC_EC factors (**Figure 6**). Along with the changes in specific elements noted above, the Pb factor had distinct shifts off of the one-to-one line for each constrained run. Generally, all of the non-constrained elements in the Pb contributions were higher in a constrained run than in the initial run.

Profiles of each factor also had noticeable changes among runs (**Figure 7**). In particular, the NO_3 mass shifted between factors based on the run. In the initial run, there was mass from NO_3 in the Mn_Fe, Zn, and OC_EC factors. In most of the constrained runs, none of these factors had mass from NO_3. In addition, the total mass apportioned to each factor showed a large difference in the Zn and NO_3 mass between the initial run and the constrained runs (**Figure 8**). Other factors also varied in total mass, including the copper factor, which had no total mass for the Gkey runs. It should be noted that since the nitrate is overwhelmingly regional, at hourly resolution we expect its behavior as a "source profile" to vary wildly due to diurnal nitrate dynamics. Thus, we EXPECT it to be unstable in the PMF solutions, moving around between factors, and quite possibly there is no optimal solution with respect to nitrate.

As the initial G-space plots implied, not all elements were easily pulled to zero in this example. Only masked Gkey and autopull with a Q-limit of 100 were able to set/pull all of the defined elements to zero. However, these both also had impacts on other elements, such as the contributions of the nitrate and

OC_EC factor. The full solution should be evaluated with each set of constraints before choosing a reasonable "final" solution.

The solution using autopull with a Q-limit of 10 is the most reasonable solution based on this analysis: some of the elements were pulled to zero and the Q value only increased by 123 units. Although the masked option was able to set all of the elements to zero with a similar change in Q value, it is apparent from the autopull results that some elements should not be equal to zero. The autopull with a Q-limit of 10 solution had the smallest deviation from the original solution in terms of percent mass of each factor and the factor profiles were also similar.

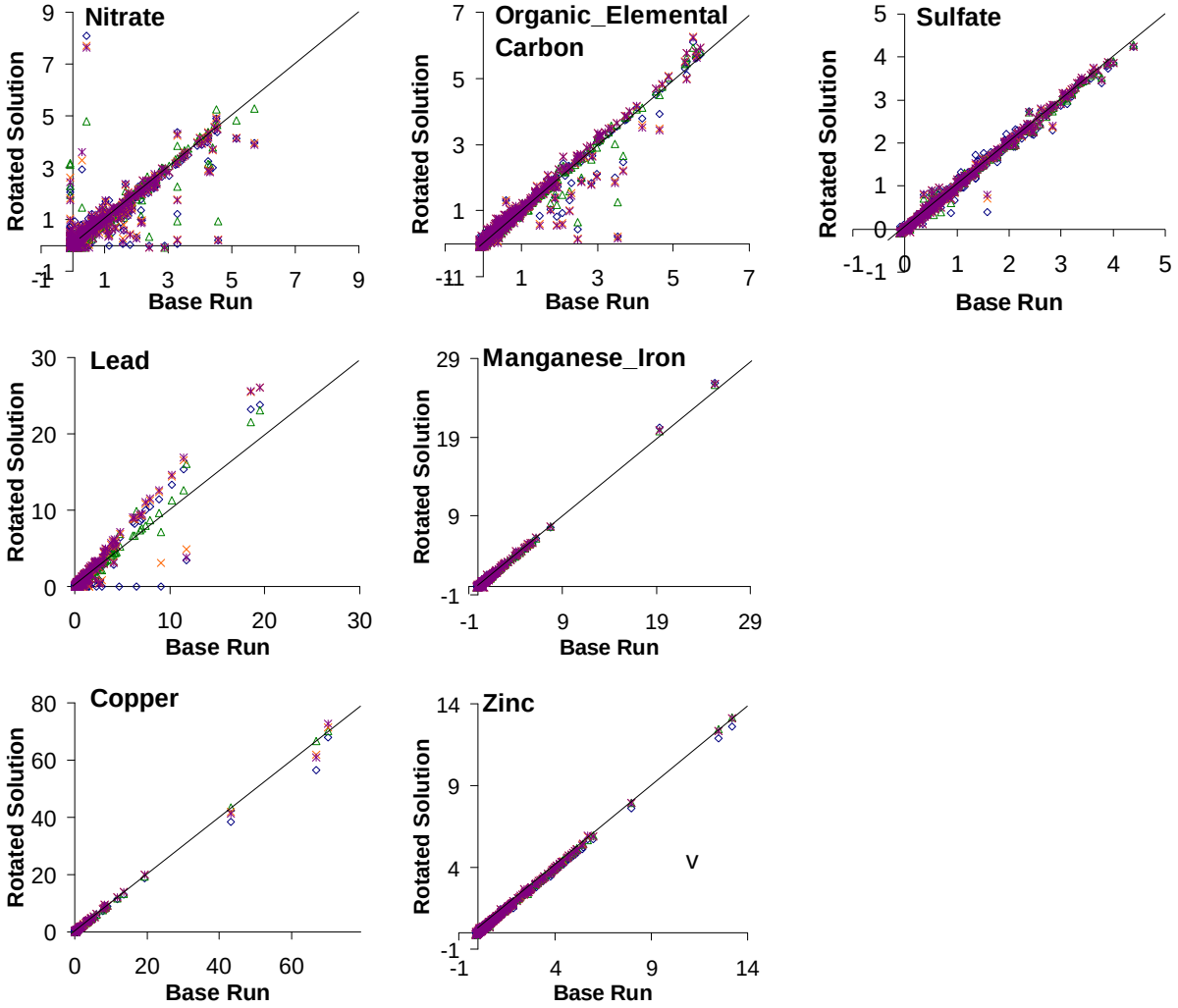


Figure 6. *Comparison of contributions of the base run (x-axis) and rotated solutions (y-axis).* (Data from StLouis_base.dat, StLouis_masked.dat, StLouis_autopull.dat.)

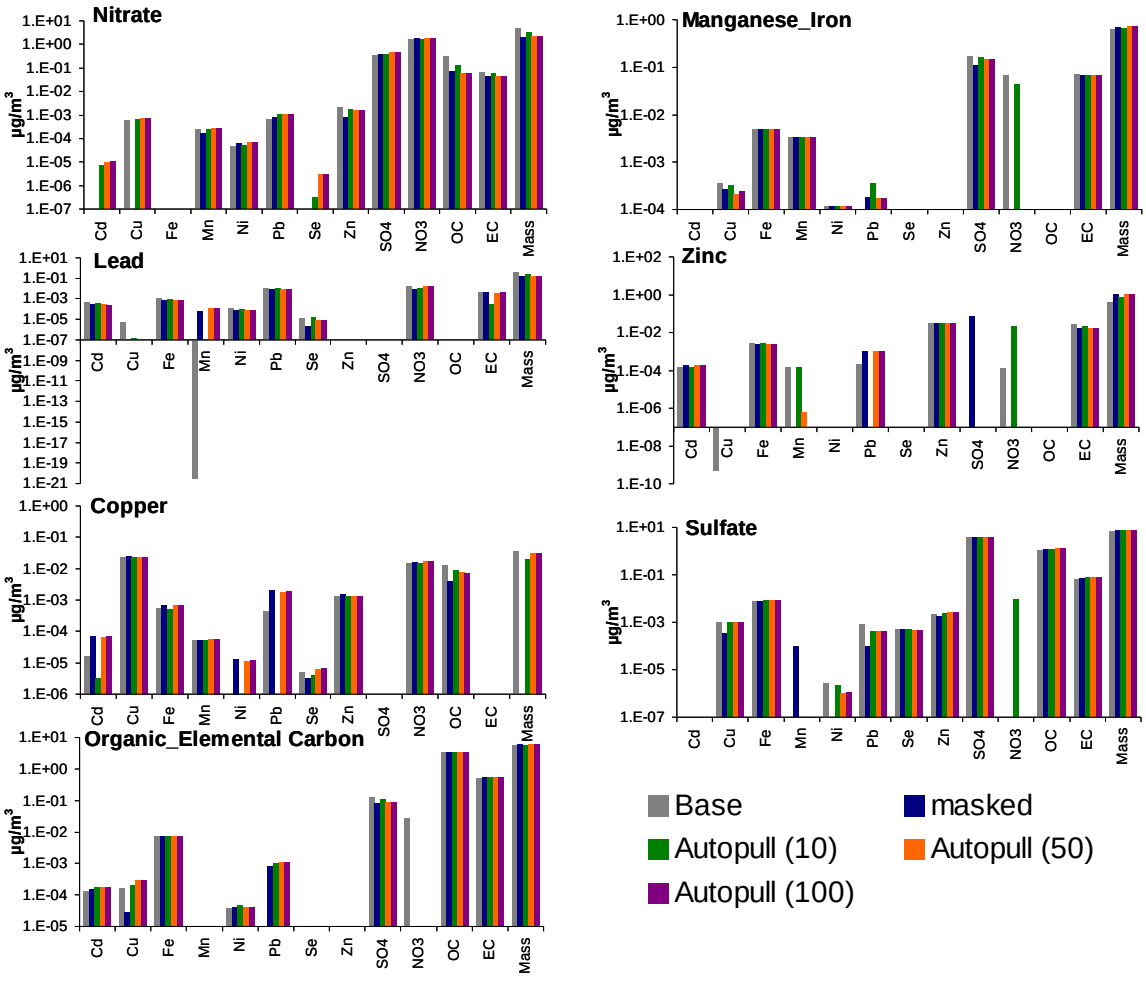
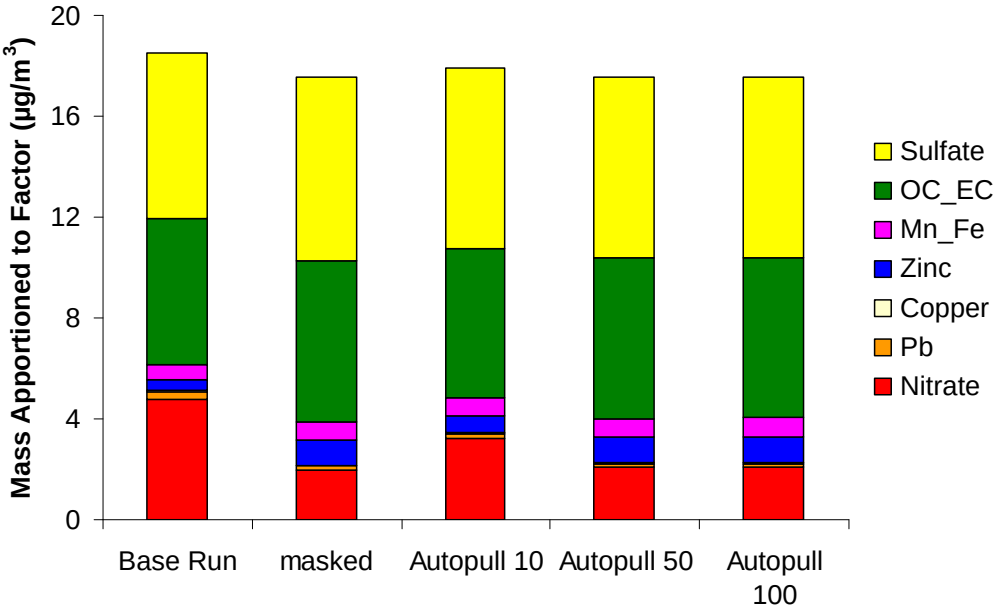
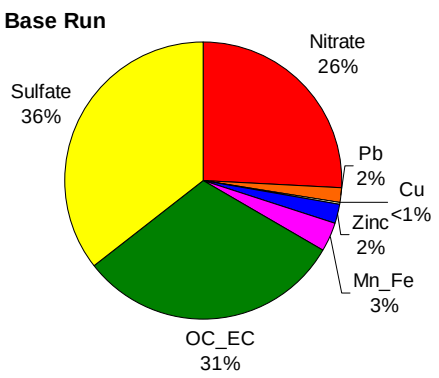
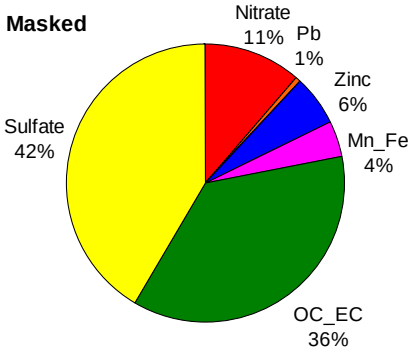
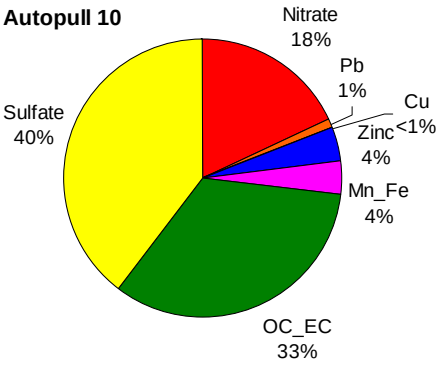


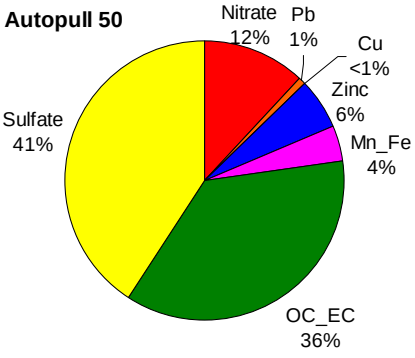
Figure 7. *Comparison of profiles of the base run and rotated solutions.* (Data from StLouis_base.dat, StLouis_masked.dat, StLouis_autopull.dat.)


Mass Apportioned to Factor (µg/m3)
20
16
12
8
4
0
Base Run
masked
Autopull 10
Autopull 50
Autopull 100
Sulfate
OC_EC
Mn_Fe
Zinc
Copper
Pb
Nitrate


Base Run
Nitrate 26%
Sulfate 36%
Pb 2%
Cu <1%
Zinc 2%
Mn_Fe 3%
OC_EC 31%


Masked
Nitrate 11%
Pb 1%
Zinc 6%
Mn_Fe 4%
Sulfate 42%
OC_EC 36%


Autopull 10
Nitrate 18%
Pb 1%
Cu <1%
Zinc 4%
Mn_Fe 4%
Sulfate 40%
OC_EC 33%


Autopull 50
Nitrate 12%
Pb 1%
Cu <1%
Zinc 6%
Mn_Fe 4%
Sulfate 41%
OC_EC 36%

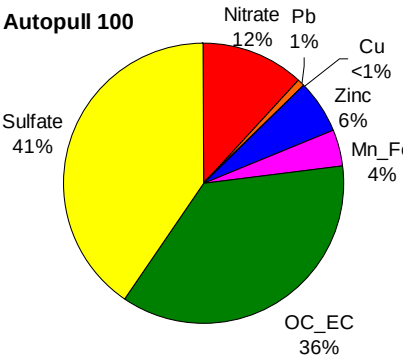

Autopull 100
Nitrate 12%
Pb 1%
Cu <1%
Zinc 6%
Sulfate 41%
OC_EC 36%

Figure 8. Distribution of mass for the base run and each constrained run as bar (a) and pie chart (b). *(Data from StLouis_base.dat, StLouis_masked.dat, StLouis_autopull.dat.) Results with iniparams and moreparams files may vary slightly.*

5.1.4 Pulling Elements of F (BB) Matrix

Source profiles of local steel facilities were used to determine appropriate ratios of iron and manganese in the steel factor. In the St. Louis dataset, the average ratio of iron to manganese in the ambient air was 10. Only 8 samples had ratios above 40. In the ME-2 results, the ratio was 1.5. Based on the ambient data, it seems likely that the ratio in the ME-2 results is low. The profile of the Granite City Steelworks basic oxygen furnace was used as a representative sample as it is believed to be heavily impacting the site. However, the ratio of iron to manganese in the source profile was 60. It is unlikely that the ratio constraint will be successful without a large change in the Q-value, so autopull was used with Q-limits of 100, 1000, and 10000 (reference files StLouis_autopull_BB.ini and moreparams_StLouisAutoBB.txt).

In this example, the ratio was defined as [Fe]/[Mn]=60. In order to maintain the normalization equations, auxiliary equations should be written as a sum that equals zero. Rearranging produces the equation [Fe]-60*[Mn]=0. Fe is the 3rd row and Mn is the 4th row of the BB matrix and the Mn/Fe factor is the fifth factor. Thus the resulting autopull equation is:

```
ii=1;
equ>  AUTO[ii], errmod=-22;
      term>pos; @BB[3,5]; term!;
      term>neg; @PULLCONST[1]; @BB[4,5]; term!;
equ!;

%Qmain limit          Expected/target value
AUTO.aux1[ii]=10000;     AUTO.aux3[ii]=-0.1;
```

A new constant factor array (elements will not be modified during modeling) was defined for the constants in the pulling equations as follows:

```
defarr constfact, PULLCONST[nc];  % Constants for pulling equations
```

Where nc is the number of pulling equations (1 in this example). This array was populated with only one value, 60. Therefore the code above can be interpreted as an autopull equation of sum of Fe in the steel factor minus 60 times the manganese in the steel factor, pulled to zero with a Q-limit of 10000.

Regardless of the Q-limit, the ratio was not pulled above 1.88. This confirms the hypothesis, based on the ambient data analysis, that this constraint cannot reasonably be met. It is likely that this factor is influenced by sources other than the Granite City Steelworks stack.

5.2 Craig (Cleveland) STN $PM_{2.5}$

ME-2 was run on a data set of 21 species, sampled every six days from December 26, 2000 through December 31, 2006, at the Craig site (**Figure 9**). Uncertainty estimates were calculated based on Wade et al. 2008. This data set was chosen to illustrate adding constraints to the PMF model based on two types of *a priori* information: 1) knowledge of steel facility closure during early 2002 (used to constrain the **AA** matrix by pulling the steel contribution to zero) and 2) a presumed source profile for steel (used to constrain the **BB** matrix by pulling the ratio of iron to manganese to ratio in the known profile).

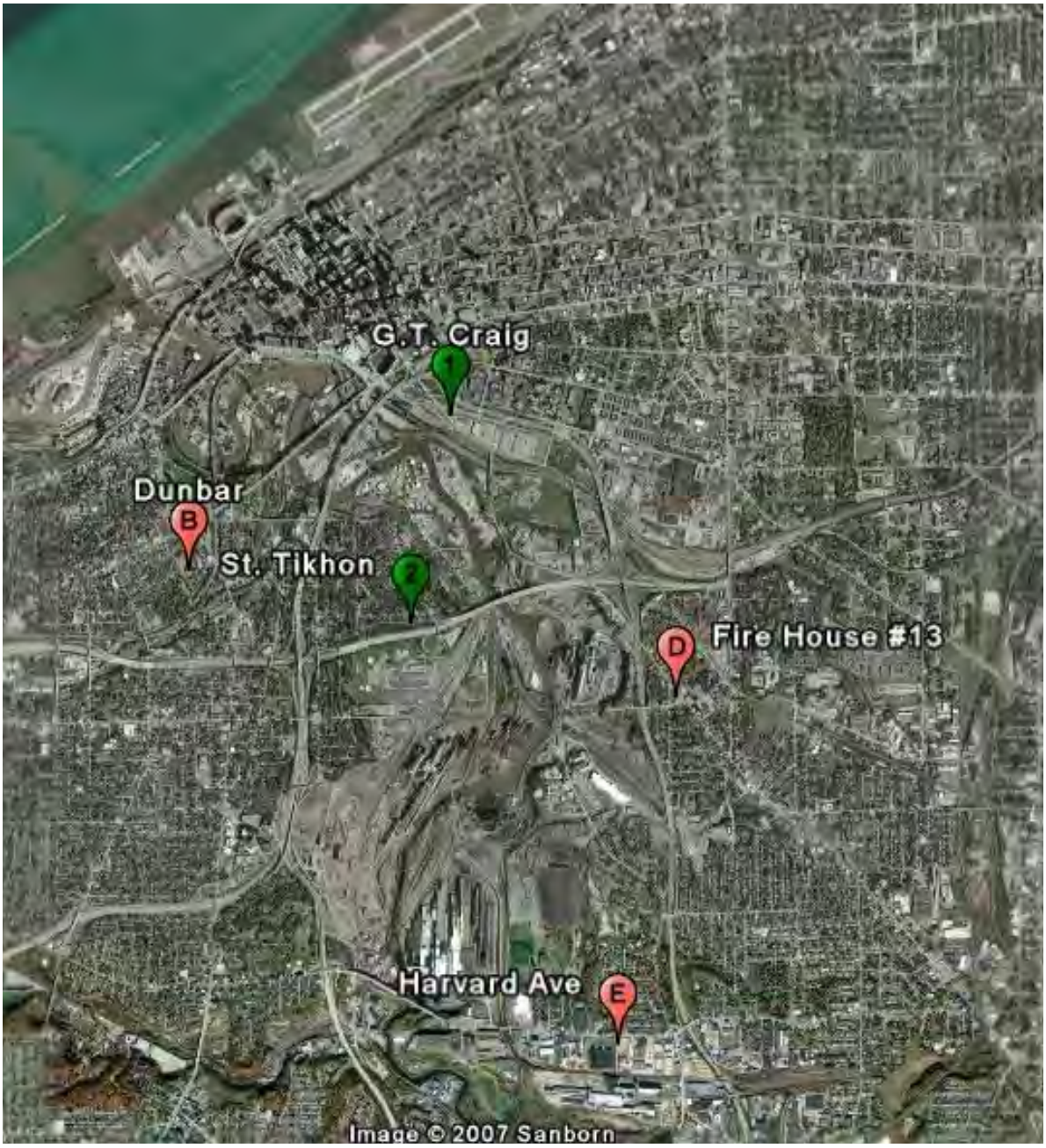


Figure 9. *Image of GT Craig and other monitoring sites in Cleveland. Green sites are in nonattainment for the $PM_{2.5}$ air quality standard.*

5.2.1 Input Parameters and Base Solution

After exploring different numbers of factors and input parameters, a base solution of 8 factors was chosen as the most physically realistic solution. The 8 factors identified (noted by key species) were zinc, soil, organic carbon, copper/nickel, elemental carbon, nitrate, steel, and sulfate. The initial parameter values for this example were as shown in the script excerpt below (reference file Craig_base.ini). In this example, 20 runs from random starting points were conducted to look for the global minimum Q-value and to avoid a local minimum Q-value. The minimum Q-value was consistently around 8995 regardless of

the start point, implying this is a global, not local, minimum. The first run was chosen as a starting point for constrained runs.

```
version=1.203;
monitor=5;
robust=1;
posoutdist=4;
negoutdist=4;
missdatlim=-990;
bdlneg=0;

convtests
  0.100,       20,        300,        0,      0,    0.0001,   %level 1
  0.010,       40,        800,        0,      0,    0.0001,   %level 2
  0.0002,      60,       2000,        0,      0,    0.00002;  %level 3
% deltaQ   consecut.  max cumul.    not     not   gg2 norm
%   test      steps    step count    used    used     test
cgresets  10,  80,  1,  1,  1,  1;
precmode=15;
numtasks=20;
variables
  'numoldsol'=1,
  'alowlim'=-0.1,
  'blowlim'=0.0,
  'seed1'=483,
  'normc1'=0.01,
  'contrun'=0;

if> (contrun>0);
  numtasks=1; goodstart=1;
if!;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
##d='CraigData.txt';
##p='Craig_Base.dat';
##m='Craig_Base';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   number of rows   number of columns    number of factors
        n1=584;            n2=21;              np=8;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  std-dev coefficients and errormodel code for the main equations
       c1=0.0;   c2=0.0;   c3=0.0;     em=-14;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

5.2.2 Pulling Elements of G Matrix

The closure of a prominent steel-producing facility near the Craig site from January to March 2002 likely impacted the steel contribution to total $PM_{2.5}$. Therefore, contributions of the steel factor should be lower than normal during that period. Most steel contributions during this time frame were low in the base run, but four were greater than 0.5. Two methods of pulling the all of contributions of the steel factor from January to March 2002 down were compared, including: masked AA.fkey (forcing elements to equal the zero) and autopull of the elements to zero (with various Q limits).

AA.fkey

A masked AA.fkey was applied to this data set (reference file Craig_masked.ini). As with the St. Louis example, four variables were changed in the '`defines`' section of the script:

- `contrun=1` to indicate that this run is starting from a prior ME-2 solution,
- `numoldsol=1` to indicate that the first solution in the file of output from previous runs is to be used as the starting point for this run,
- `##p='Craig_base.dat'` to point ME-2 to the file with the previous solutions, and
- `##m='Craig_AAfkey'` to tell ME-2 the prefix for the output from this run.

To define each pull, the command:

`'AA.fkey[i,j]=-6;'`

was used in the '`preproc`' section of the script, where i was the sample number and j was the column of the steel factor. For this example, 22 fkeys were defined. For each one, the j value was constant (all pulls carried out on the steel factor) and the i value indicated each of the 22 samples during which the contribution from the steel factor was taken to be zero. Although this guaranteed that the element would be equal to zero, the Q-value was allowed to increase as much as necessary to accommodate the constraint. Additionally, because of the non-negativity constraint, it was possible that accommodating the additional constraints would cause other factor elements to change considerably.

When the masked AA.fkey was used, all of the pulled elements were zero in the modeling results, as required by masking; however, the Q-value increased by over 170 units (**Table 11**). As explained in the St. Louis example, when using a masked AA.fkey (or BB.fkey), the user should be very certain that the a priori information is accurate. In this example, additional steel sources that remained active during the pulling period could have resulted in non-zero contributions at the Craig site. Additional sources that may contribute to this factor, and their activity during the pulling period, should be examined.

Table 11. *Q-values and number of pulls that reached the target value for each type of rotational tool used.*

Method	Q	Qmain	Qaux	Number of pulls that reached target (out of 22)
Base Run	8995	8995	0.0	--
AA.fkey-masked	9165	9165	0.0	22
Autopull- 10	9077	9022	55	11
Autopull- 50	9336	9062	274	19
Autopull- 100	9644	9101	543	19

Autopull

Autopull equations were used to pull the factor elements to the user-defined lower limit of -0.1. Autopull allows for limits on the Q-value, i.e., a constraint will only be implemented if the Q-value does not increase more than the user defined Q-limit, the maximum change in Q that is allowed for each autopull equation. A larger Q-limit makes it more likely that the target value will be reached. In this example, 3 different Q-limits were compared: 10, 50, and 100 (reference file Craig_autopull_AA.ini, only a Q-limit of 10 is provided as a reference script). The same variables were changed as in the AA.fkey example (`contrun=1`, `numoldsol=1`, `##p='Craig_base.dat'`, `##m='Craig_AAfkey'`). The following commands were used in the '`equations`' section of the script to develop the autopull equations:

```
ii+1; equ>  AUTO[ii], errmod=-21; term>pos; @AA[i,j]; term!;equ!;
AUTO.aux1[ii]=50; AUTO.aux3[ii]=-0.1;
```

Where ii was previously defined as 0, i is the sample number and j is the column of the steel factor. As with the AA.fkey commands, twenty-two autopull equations, each for a different sample number (i), were

included. In the three separate autopull runs, only the `AUTO.aux1[ii]` value was changed (10, 50, 100).

With the first Q-limit of 10, only 11 of the elements were successfully pulled to the lower limit, and Qmain increased by 27 units (Table 11). With a Q-limit of 50, 19 elements were successfully pulled to the lower limit, while Qmain increased by 67 units. With a Q-limit of 100, still only 19 elements were successfully pulled to the lower limit, and Qmain increased by 106 units. Additional tests indicated that all 22 samples were successfully pulled to zero only when the Q-limit was increased to 500. **Figure 10** shows that the elements that started with the lowest values were pulled to zero by most of the constraints; however, for larger normalized contributions (greater than one), elements could not be pulled to zero using autopull, even when the Q-value was allowed to change by 100 units for each pull.

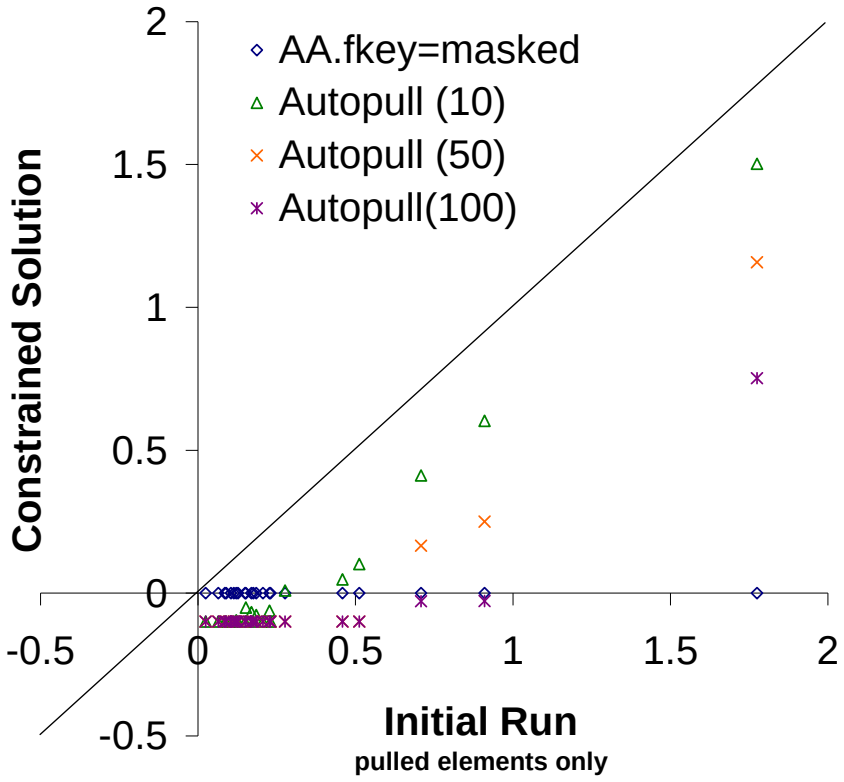


Figure 10. *Comparison normalized contributions to base run (x-axis) and each rotated solution (y-axis)*

5.2.3 Evaluation of Results

As with the St. Louis example data set, contributions and profiles of each factor and total mass apportioned to each factor were examined for each constrained run.

Pulling the contribution of the steel factor to zero for this particular period had a small, although noticeable, affect on other factors. This is illustrated in the scatter plots of normalized contributions of the base run versus contributions in each constrained run (**Figure 11**). Samples of the steel factor, itself, were pulled down significantly when base-run contributions were low (below one). However, many steel contributions increased for samples with base-run contributions greater than two. Most other factors remained relatively unaffected by the pull, as evidenced by their staying near the one-to-one line in the scatter plots.

Profiles of each factor also had noticeable changes among runs (**Figure 12**). In particular, OM was eliminated from the sulfate factor in most of the constrained runs. EC shifted mass between factors; for instance, its mass was reduced in the soil factor while it increased in the OM factor. Trace metals also exhibited some variation between the base run and various rotations. However, none of these shifts in mass between factors altered the identification of the factors. Furthermore, the total mass apportioned to each factor did not exhibit any significant changes between the base run and the constrained runs (**Figure 13**).

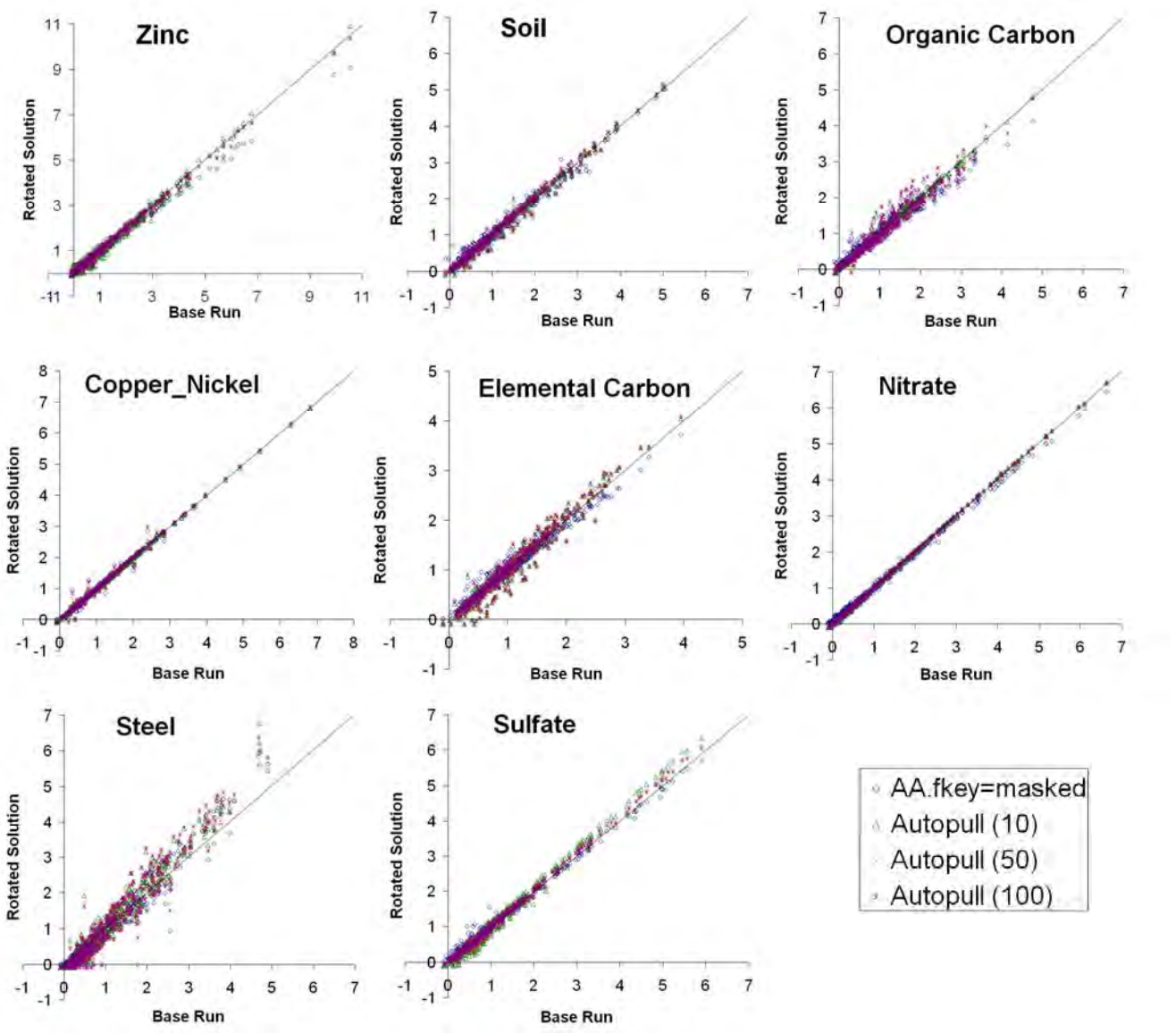


Figure 11. *Comparison of contributions of the base run (x-axis) and rotated solutions (y-axis).*

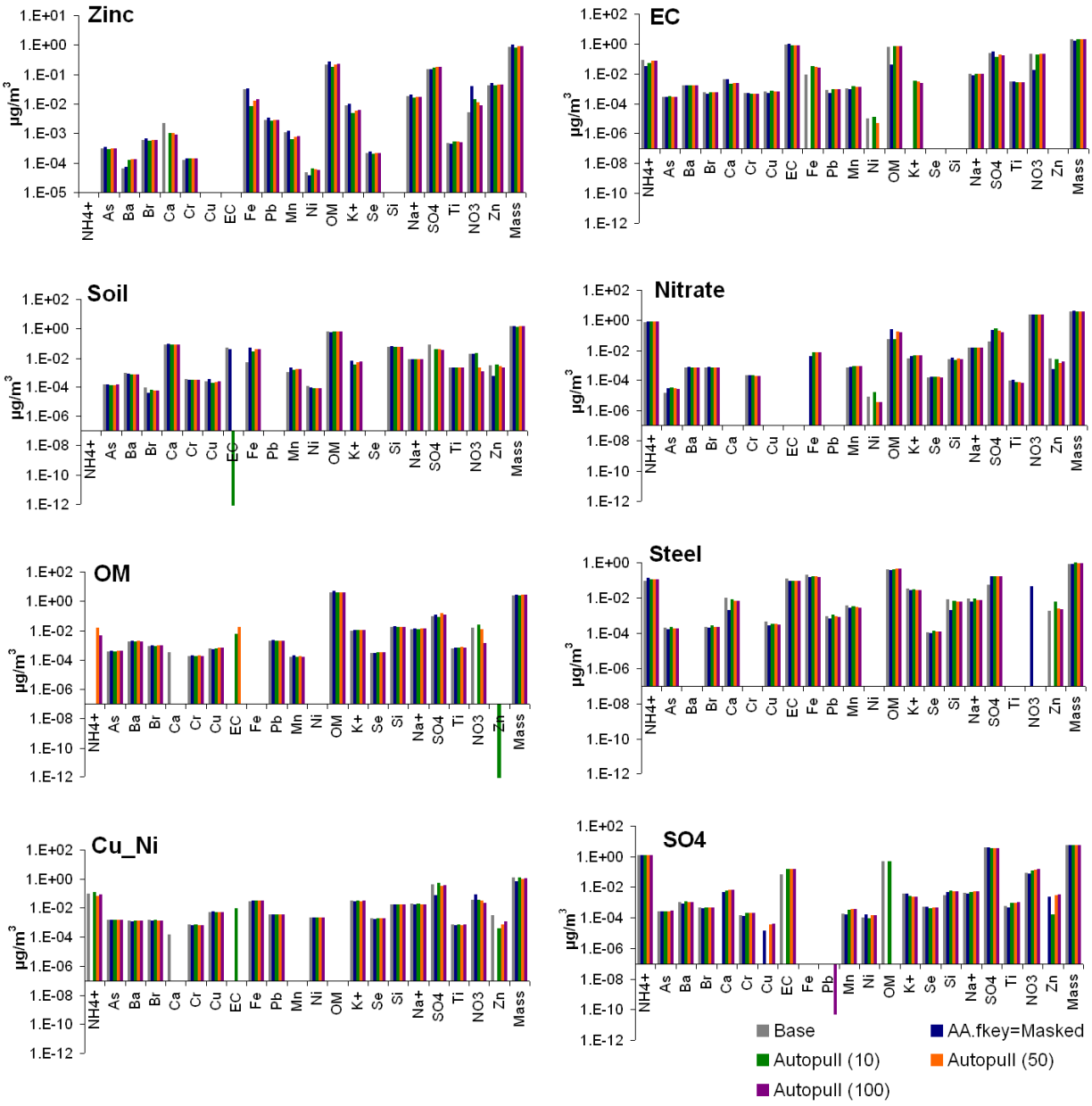


Figure 12. *Comparison of contributions of the base run (x-axis) and rotated solutions (y-axis).*

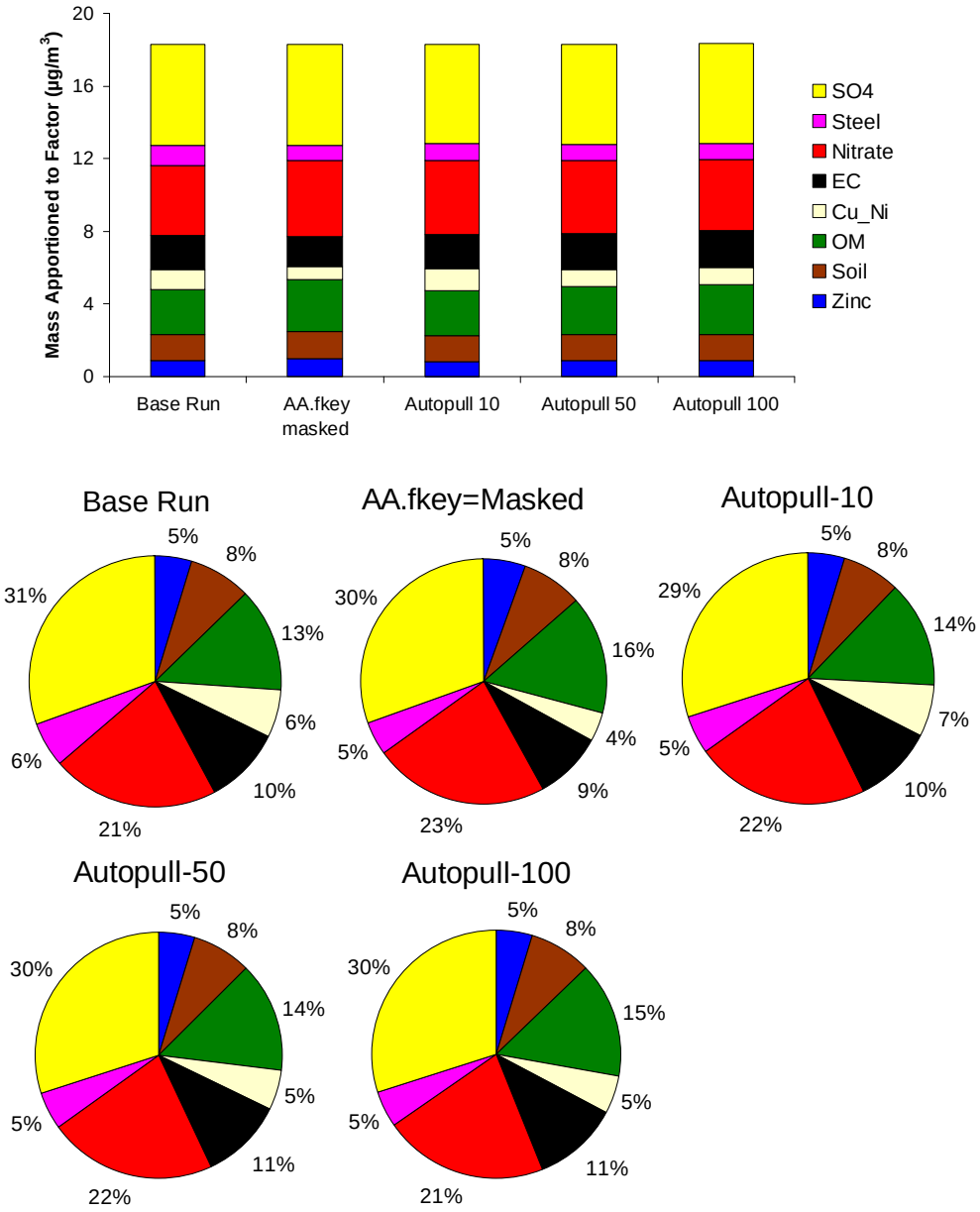


Figure 13. *Distribution of mass for the base run and each constrained run.*

5.2.4 Pulling Elements of F Matrix

As in St. Louis, the profile of the Granite City Steelworks basic oxygen furnace in St. Louis, MO, was used as a representative sample of a steel source in constraining the steel factor profile in the Craig results. Specifically, the ratio of iron to manganese was used to constrain that same ratio in the steel factor at Craig. In the Craig dataset, the initial ratio of iron to manganese was 54.1. However, the ratio of iron to manganese in the source profile was 60. Since the ratio from the steel factor was already close to the target ratio, using an fkey constraint was reasonable. The constraint was applied using an auxiliary equation.

The following code was used (reference file Craig_aux_bb.ini):

```
equ>  AUXAR[2,7], Data=0,C1=normc1, C3=0.0, errmod=-12;
      term>
            pos; @BB[10,7]; term!;
      term>
            neg; @PULLCONST[1]; @BB[12,7]; term!;
equ!;
```

where `BB[10,7]` represents the 10th element (Fe) of the 7th factor (steel) and `BB[12,7]` represents the 12th factor (manganese) of the same factor. The desired ratio (`PULLCONST[1]`) was set to 60 (see below). The expression defining the target ratio ([Fe]/[Mn]=60) could be rearranged, producing the equation [Fe]-60*[Mn] = 0. A new constant factor (entries will not be modified during modeling) array was defined for the constants in the pulling equations as follows:

```
defarr constfact, PULLCONST[nc];  % Constants for pulling equations
```

Where nc is the number of pulling equations (1 in this example). This array was populated with only one value, 60. Therefore the code above can be interpreted as an equation of sum of Fe in the steel factor minus 60 times the manganese in the steel factor, pulled to zero.

Using auxiliary equations and experimenting with different Q-limits and C1 values, the Fe/Mn ratio was pulled up to 57.7 with an increase in Qmain of 3 compared with the base run. Using autopull equations resulted in an Fe/Mn ratio of 60.0 with an increase in Qmain of 6 (reference file Craig_autopull_bb.ini). This pull was achieved with a Q-limit of only 10, therefore additional autopull runs are not shown here.

Scatter plots of normalized contributions were very close to the one-to-one line (**Figure 14**), indicating little change in contributions for each factor from the base run to rotated runs. Although the factor profiles showed some shifting of nitrate and trace metals between factors (**Figure 15**), the factor identifications remained unchanged. The total mass contributions remained nearly identical to those in the base run (**Figure 16**). This supports the initial identification of this factor as being from a steel facility and indicates that 60 is a reasonable ratio for Fe/Mn in this factor.

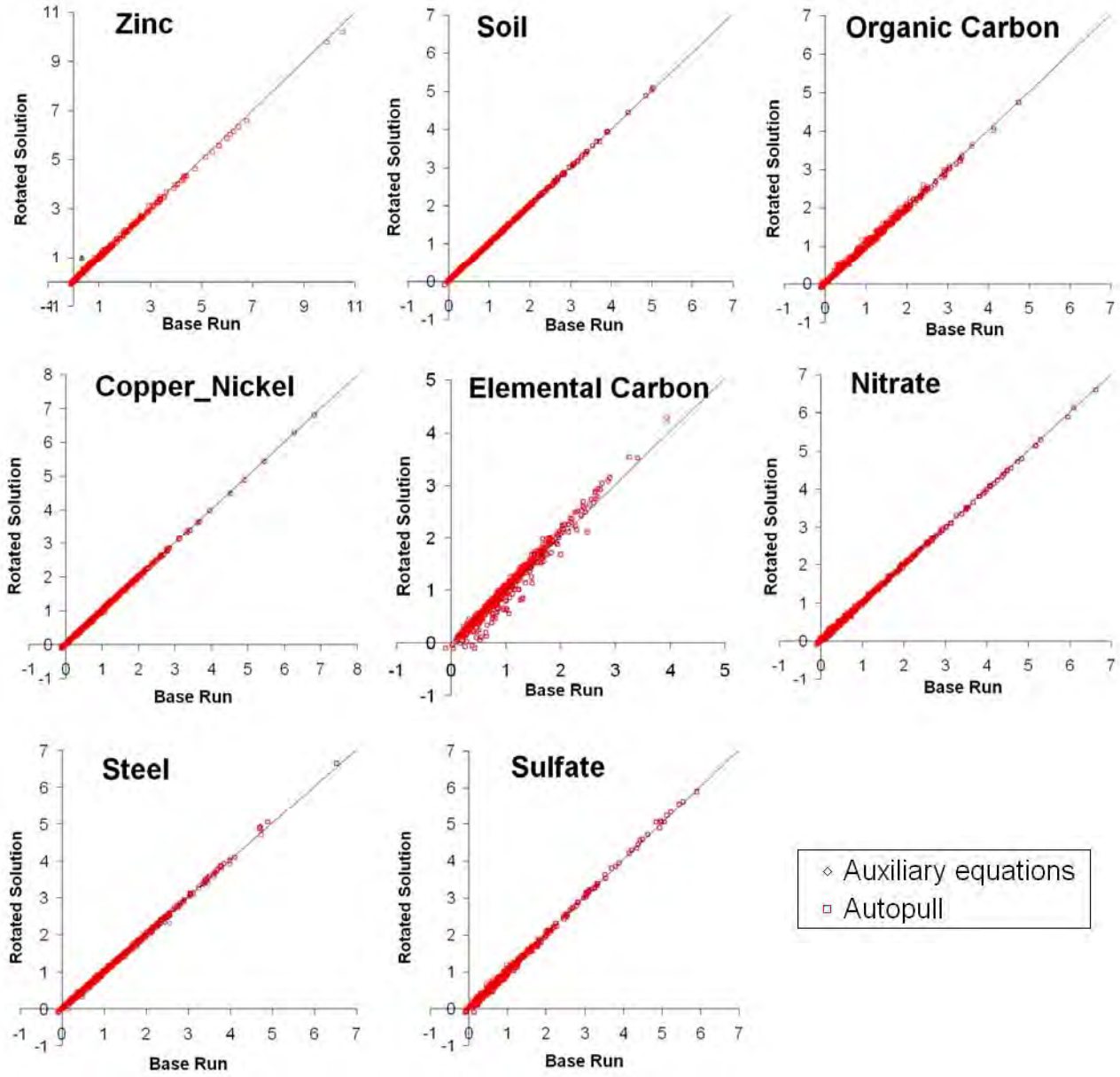


Figure 14. *Comparison of contributions of the base run (x-axis) and rotated solutions (y-axis).*

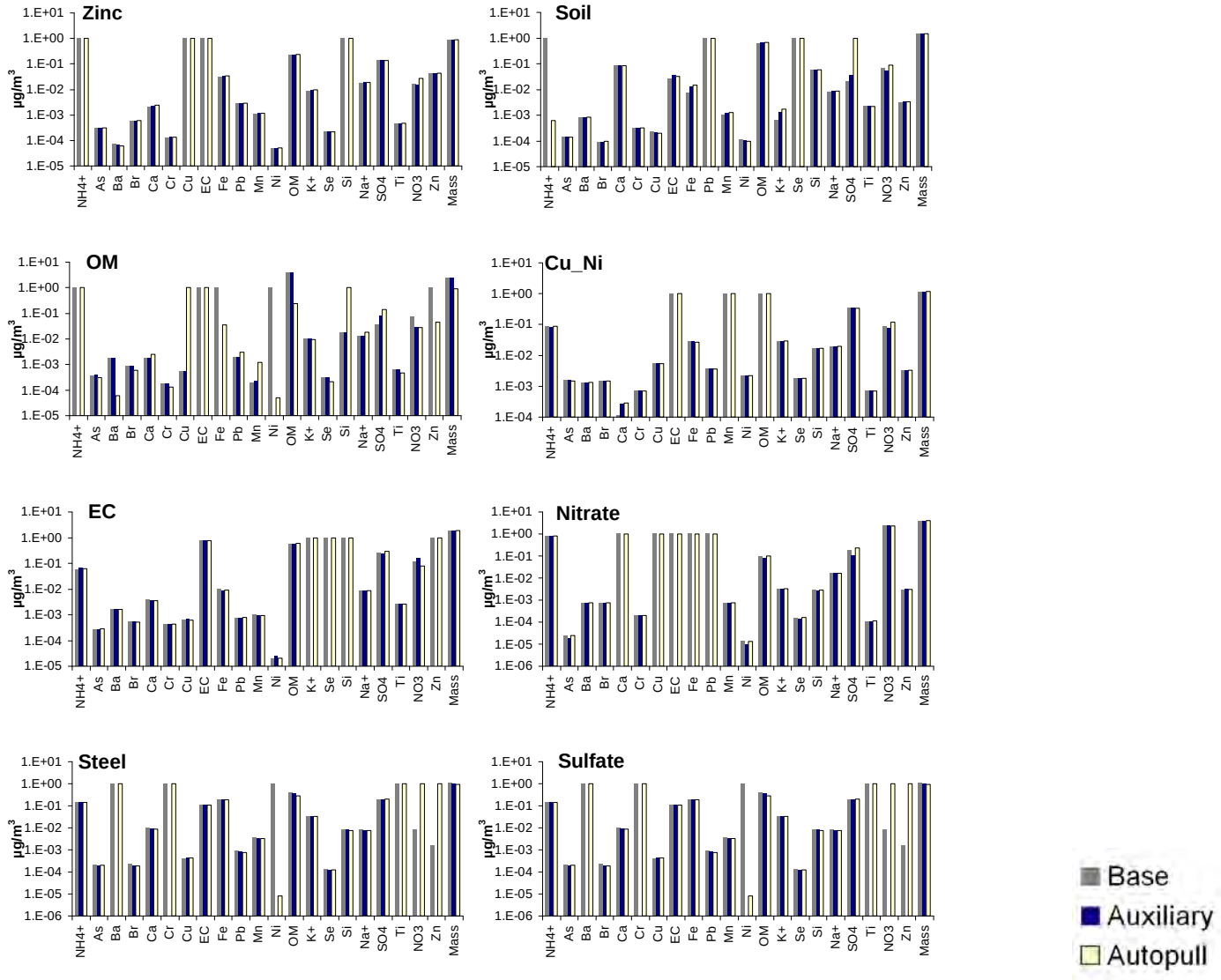


Figure 15. *Comparison of contributions of the base run (x-axis) and rotated solutions (y-axis).*

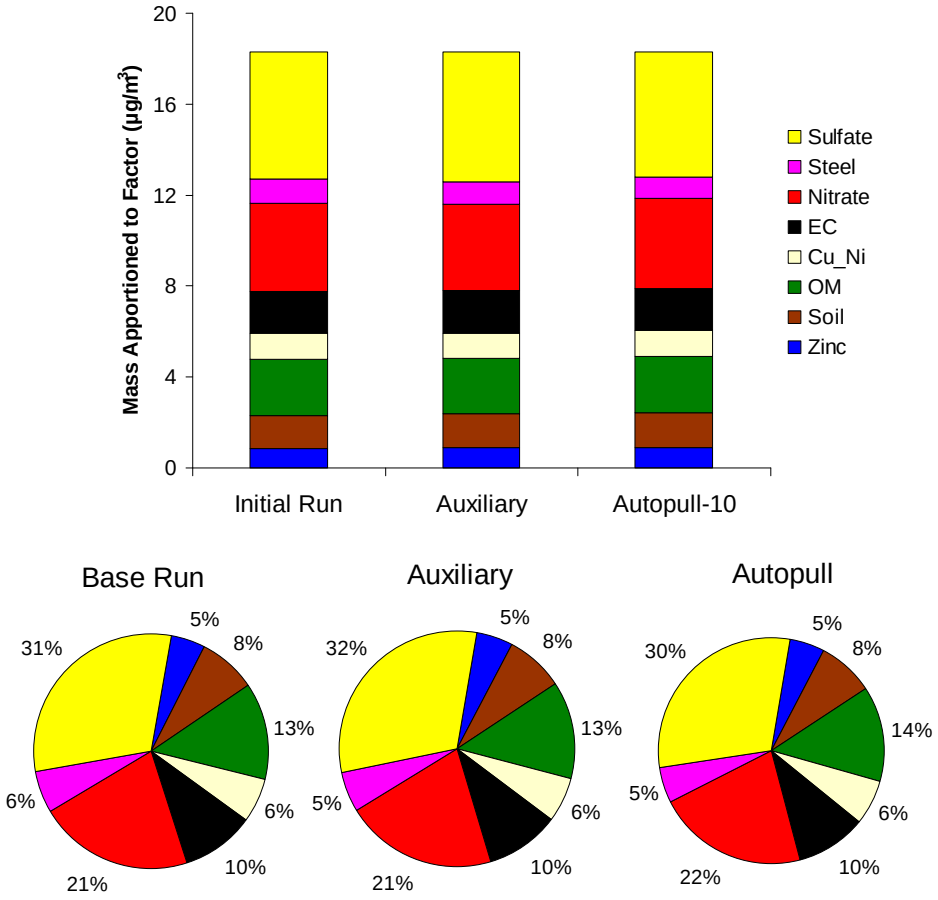


Figure16. *Distribution of mass for the base run and each constrained run.*

5.3 Baton Rouge PAMS VOCS

A key component in receptor modeling is utilizing unique or near-unique species as tracers for specific sources. This is needed whether conducting analysis with a chemical mass balance (CMB) model, where ambient data are fit to specific source profiles, or with PMF and Unmix. We assume that if the a tracer species originates from only one source, then we can attribute mass to that source by quantifying the amount of other species associated with that tracer. When we know the tracer only has one source, it would be useful to apply this *a priori* knowledge to acquire a revised solution.

In this example, Speciated VOC data from 3-hr samples taken during morning hours in summer of 2005-2006 at the Baton Rouge Photochemical Assessment Monitoring Stations (PAMS) site (**Figure 17**) were modeled using ME-2. Uncertainties are not routinely characterized as part of PAMS, so values were set between 20% and 60% depending on the species. 307 samples and 21 species, including total mass (total nonmethane organic compounds (TNMOC)), were used. This data set provides an example for pulling a unique source marker, acetylene from mobile exhaust, in the **BB** matrix. If unique source markers are available in a dataset, pulling them so that all or nearly all of their mass is loaded into the appropriate factor may be an effective way to achieve the most reasonable solution. In the following examples, the loading of acetylene was pulled using BB.fkey and autopull equations. The scripts used in this section are: BR_baserun, BR_BBfkey, BR_acetauto-20, BR_acetauto-21_1 (only ini version as this proves to be redundant), BR_acetauto-21_50, BR_acetauto-21_14.

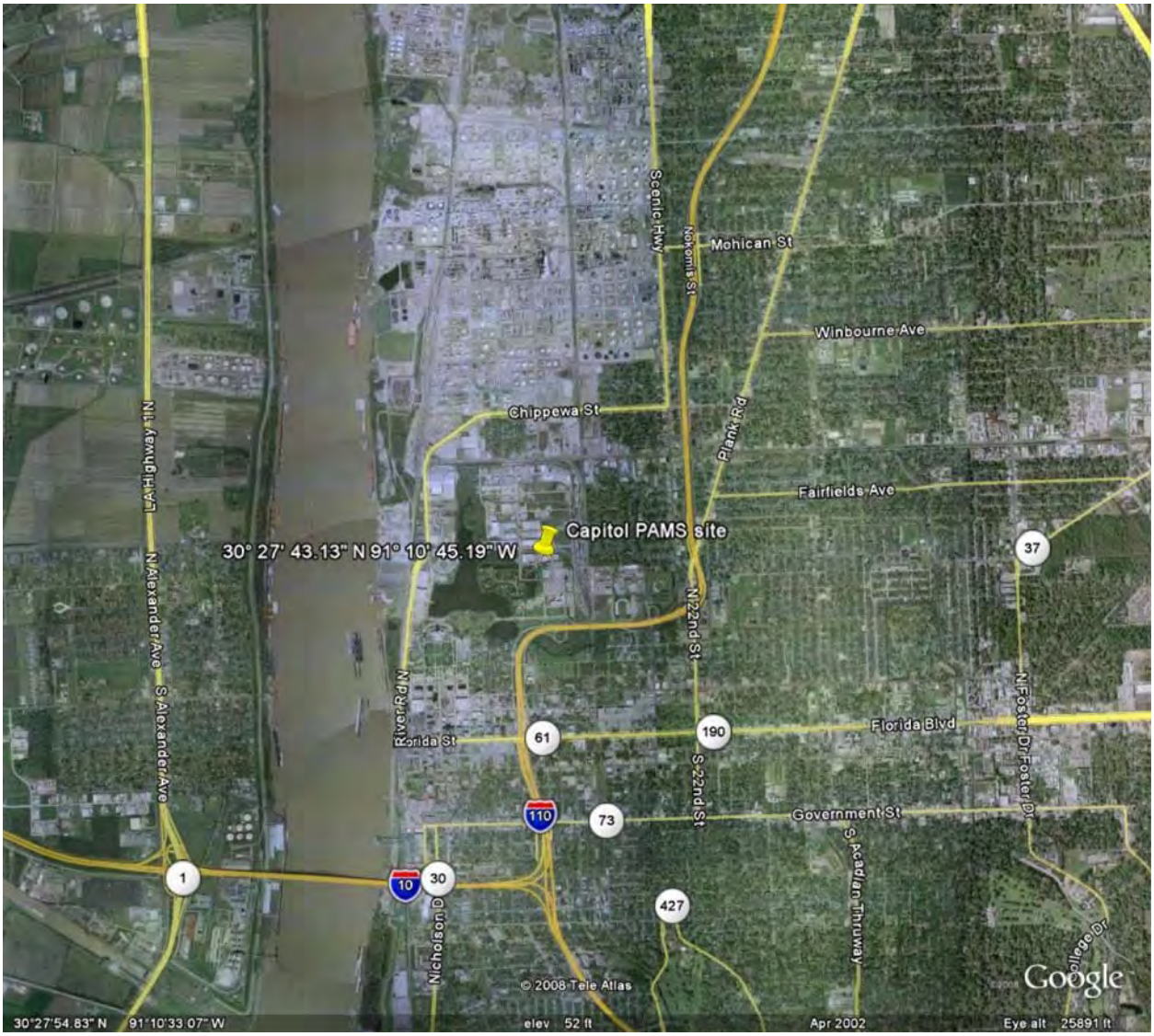


Figure 17. *Site location of Baton Rouge Capitol PAMS site (site ID 220330009).*

5.3.1 Input Parameters and Base Solution

Five factors were determined in the base run. These factors, their overall average contributions, and the species used to identify each factors are: 1) evaporative emissions (24% of the mass on average; markers used: butanes, propane); 2) motor vehicle exhaust (32%; acetylene, aromatics); 3) liquid/unburned gasoline and feedstock (29%; pentanes, hexane); 4) diesel/heavy duty fuel (9%; C-9 and C-10 alkanes); and 5) fresh industrial and refinery emissions (6%; ethylene and propylene). Q was equal to 6928 for the selected base run; this value was consistently obtained within a few units over 20 random runs. In the base run, seventy percent of the acetylene was loaded in the mobile exhaust factor (this value was similar in all random runs), and will be used as the tracer for this source. In the next runs, BB.fkey and autopull will be used to increase the amount of acetylene in the mobile exhaust factor (factor 2). A code excerpt for the base run is provided below.

```
version=1.203;
monitor=5;
robust=1;
```

```
posoutdist=4; negoutdist=4;
missdatlim=-990;
bdlneg=0;
convtests
  0.100,       20,        300,        0,      0,    0.0001,   %level 1
  0.010,       40,        800,        0,      0,    0.0001,   %level 2
  0.0002,      60,       2000,        0,      0,    0.00002;  %level 3
% deltaQ   consecut.   max cumul.     not     not   gg2 norm
%   test      steps    step count    used    used     test
cgresets  10,  80,  1,  1,  1,  1;
precmode=15; numtasks=20;
variables
  'numoldsol'=1,   % The number of the desired starting-point solution
  'alowlim'=-0.1,  % Used as low limit for AA factor elements
                   % A better rotation is obtained with a slightly
                   % negative alowlim, such as -0.1.
  'blowlim'=0.0,   % Used as low limit for BB factor elements
  'seed1'=7,       % for initializing random numbers for initial factor
values
  'normc1'=0.01,   % Std-dev coefficient C1 for A normalization equs
  'contrun'=0;     % Change as needed. Meaning of contrun values:
      % =0: start from random initial factor values
      % =1: start from factor values read from a file
      % =2: start from factor values read from a file,
if> (contrun>0);
  numtasks=1; goodstart=1;
if!;

% Specify names for files to be opened
% Input files
##d='BR0506m5_6_pmfdata.txt';  % Data matrix XX (and possibly std-dev of XX)
##p='BR0506m5_6_PMF_ab_base.dat';  % Input file 39: the results of a previous
run.
                    % Needed if contrun==1 or contrun==2.
##m='BR0506m5_6_base';       % Main part of title of files to be written

%   number of rows   number of columns    number of factors
        n1=307;            n2=21;              np=5;
%  std-dev coefficients and errormodel code for the main equations
       c1=0.0;   c2=0.0;   c3=0.00;     em=-14;
```

5.3.2 Pulling Tracer Acetylene with BB.fkey

Next, the same run was done but BB.fkey was used to pull acetylene down in all factors but the mobile factor, factor 2. This was done by setting the BB.fkey for acetylene in all but the mobile factor to -6. This sets the element value in the profiles (acetylene in factors 1, 3, 4 and 5 in the profile matrix, BB) to zero. Specifically, the following syntax for the BB.fkey was used:

```
BB.fkey[3,1]=-6; BB.fkey[3,3]=-6; BB.fkey[3,4]=-6; BB.fkey[3,5]=-6;
```

A masked fkey on the non-mobile exhaust factors was used to set, rather than “pull”, acetylene to zero in these factors. This is typically more advantageous than the alternative of attempting to lock the acetylene in the exhaust factor to a fixed value (in this case, of 100%). This is of course only useful if the user is certain that the species (in this case acetylene), does not originate from any other source.

This resulted in a change in Q of about 45 units, and acetylene fully loaded in mobile exhaust profile (100%). The profiles of all factors changed somewhat, including changes in total mass apportioned by factor. This is expected with a “hard pull”, but the change in Q on the order of tens of units and the

variation in the profiles were acceptable, meaning that the profiles still had a reasonable mix of species associated with the likely source category. The somewhat small change in Q and in profiles indicates this new solution is within the minimum solution space.

5.3.3 Pulling Tracer Acetylene with Autopull

From the base run, two types of autopull runs were done, pulling 1) acetylene up in mobile factor and 2) pulling acetylene down in non-mobile factors. Autopull allows the user to place limits on the change in Q-value as a result of the pulling; the elements will be pulled but only if the Q-value does not increase more than the limit, defined as Qlim. In the first type, pulling acetylene up in the mobile factor, a limit on Q of 100 and of 50 was tried. In the second type, three scenarios were examined: a) pulling acetylene in non-mobile factors, such that the sum of acetylene elements in all non-mobile factors approaches zero; b) pulling acetylene in non-mobile factors such that the increase in Q from acetylene in each profile is less than 50 units; and c) pulling acetylene in non-mobile factors such that the increase in Q from acetylene in each profile is less than 14 units (i.e., a total change of about 50 units). In all cases, the expected change in acetylene concentration was set to 1 ppbC; acetylene was 1.6 ppbC in the mobile factor in the base case, with 0.7 ppbC in other factors, so 1 ppbC is a reasonable value. The purpose within each scenario is to understand how much the acetylene loading would change depending on how much the change in Q was limited. A user would expect that with a larger Qlim allowed acetylene would get closer to 100% in the desired factor, but with a larger penalty to Q. By exploring how much acetylene can be pulled within user-defined acceptable Q limits the user can further understand their solution.

The syntax used for these examples is shown below. In addition, these auxiliary equations are brought to the attention of ME-2 by the following line inserted at line 119.

```
defarr auxdata, AUTO[10];
```

(1)

```
ii=1;
equ> AUTO[ii],  errmod=-20;   %pull BB[3,2] up
    term> pos; @BB[3,2]; term!;
equ!
%%%  Qmain limit    Expected change
AUTO.aux1[ii]=50;   AUTO.aux3[ii]=1;
```

(2a)

```
equ> AUTO[ii],  errmod=-21;   %pull BB[3,x] down, x is all factors but 2
term> pos; @BB[3,1]; term!;
term> pos; @BB[3,3]; term!;
term> pos; @BB[3,4]; term!;
term> pos; @BB[3,5]; term!;
equ!
AUTO.aux1[ii]=50;   AUTO.aux3[ii]=1;
```

(2b & c, with different Q limits of 50 or 14)

```
ii=1;
equ> AUTO[ii],  errmod=-21; term> pos; @BB[3,1]; term!; equ!;
AUTO.aux1[ii]=50;   AUTO.aux3[ii]=1;

ii=ii+1; equ> AUTO[ii],  errmod=-21;  term> pos; @BB[3,3]; term!; equ!;
AUTO.aux1[ii]=50;   AUTO.aux3[ii]=1;

ii=ii+1; equ> AUTO[ii],  errmod=-21;  term> pos; @BB[3,4]; term!; equ!;
AUTO.aux1[ii]=50;   AUTO.aux3[ii]=1;
```

```
ii=ii+1; equ> AUTO[ii],  errmod=-21;  term> pos; @BB[3,5]; term!; equ!;
AUTO.aux1[ii]=50;   AUTO.aux3[ii]=1;
```

Similar results were achieved with both types of scenarios (1 and 2), with a change in Q of around 30 units in all scenarios. Acetylene was not fully (100%) loaded into mobile factor, but 89% of it is now loaded, versus 70% in base run. This “soft pull” yielded similar results to the BB.fkey run, but with a smaller penalty to Q and smaller shifts to the profiles’ composition.

5.3.4 Comparison of Results

A summary of the various scenarios and the associated change in Q is presented in **Table 12**. Profiles and mass apportioned by run are shown in **Figures 18-22**. Using BB.fkey to force 100% of our tracer into its associated factor yielded a reasonable change in Q, and profiles that, while different from the base run, were reasonable. Soft pulling with autopull yielded the results with a smaller change in Q and a higher amount of acetylene loaded into the appropriate factor compared to the baserun, when constrained by a reasonable Qlim.

The scenarios using autopull with acetylene in the mobile factor could be argued to have resulted in the “best’ solution, since there was more acetylene loaded in the mobile factor than in the base, there was not a large detriment to Q, and the profiles are still physically realistic. This resulted in an increase in TNMOC apportioned to the mobile factor, from 32% to 36%, with a reduction of 6% of TNMOC apportioned to evaporative emissions and minor changes in the other factors. In this case, pulling the tracer species up in its appropriate factor, with a fairly strict limit in the increase of Q, yielded an improved result over the base case.

Overall this series of case studies demonstrated the ability of using near-unique markers to refine a well-understood and stable solution. The mobile exhaust profile was pulled in a direction to approximate the likely mix of source profiles in the area, where all or nearly all of the acetylene is associated with mobile exhaust. With the movement of acetylene to the mobile exhaust factor from the evaporative factor in the pulled runs, additional mass was allocated to the mobile exhaust factor. Acetylene shows some collinearity with other species, typical of VOC data, so the association of acetylene with evaporative emissions is not unsurprising, as evaporative emissions are typically associated with mobile sources.

Table 12. *Summary of results from iterations using provided ini files. Results with public script and provided iniparams and moreparams files may vary slightly.*

Method	Q	Qmain	Qaux	Change in Q from base	% of acetylene in mobile factor
Initial run	6926	6926	0	n/a	70%
BB.fkey – masked	6977	6977	0	51	100%
Autopull acetylene in mobile factor, Qlim 50	6963	6948	15	37	88%
Autopull acetylene in non-mobile factors, sum of acetylene in these is zero, Qlim 50	6955	6944	11	27	88%
Autopull acetylene in non-mobile factors, Qlim of 50 for each factor	7078	6953	125	152	91%
Autopull acetylene in non-mobile factors, Qlim of 14 for each factor	6976	6941	35	50	77%

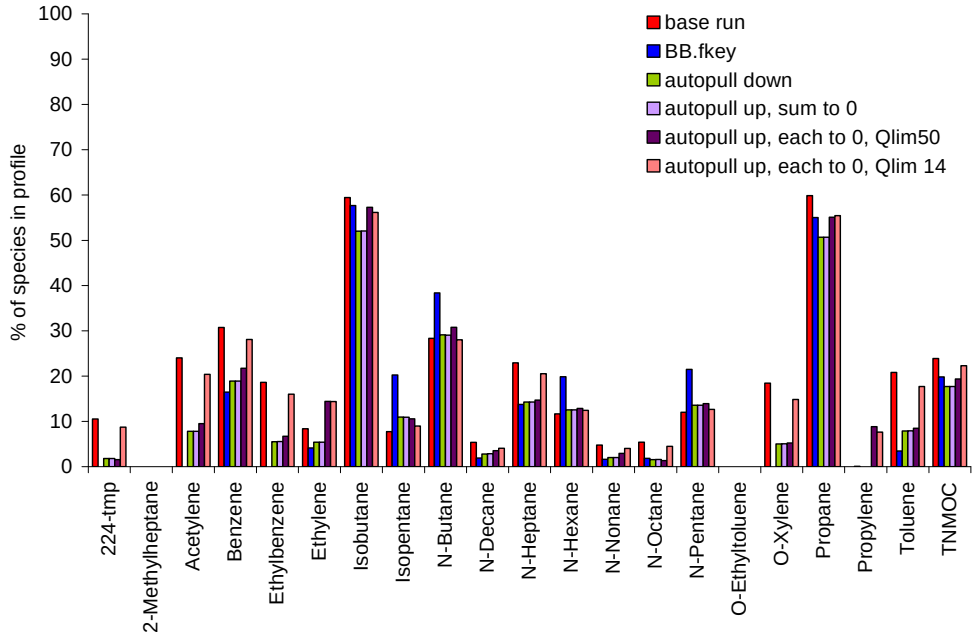


Figure 18. *Profile (% of species) for factor 1, evaporative emissions, through the base run, using BB.fkey to pull acetylene, and using autopull to pull acetylene down and up (where the sum of elements is zero, where each element is set to zero with a Qlim of 50, and where each element is set to zero with a Qlim of 14).*

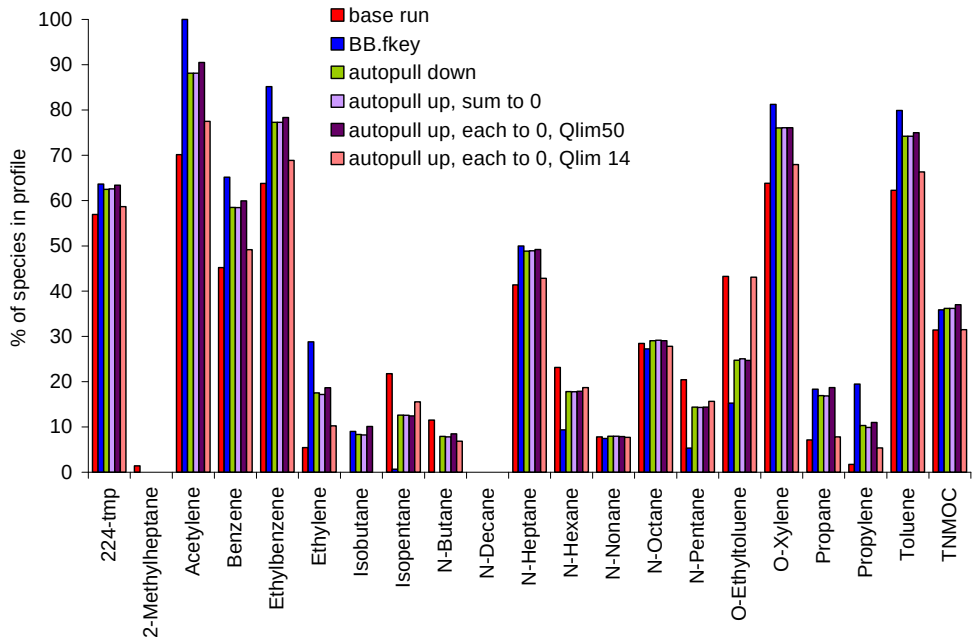


Figure 19. *Profile (% of species) for factor 2, motor vehicle exhaust, through the base run, using BB.fkey to pull acetylene, and using autopull to pull acetylene down and up (where the sum of elements is zero, where each element is set to zero with a Qlim of 50, and where each element is set to zero with a Qlim of 14).*

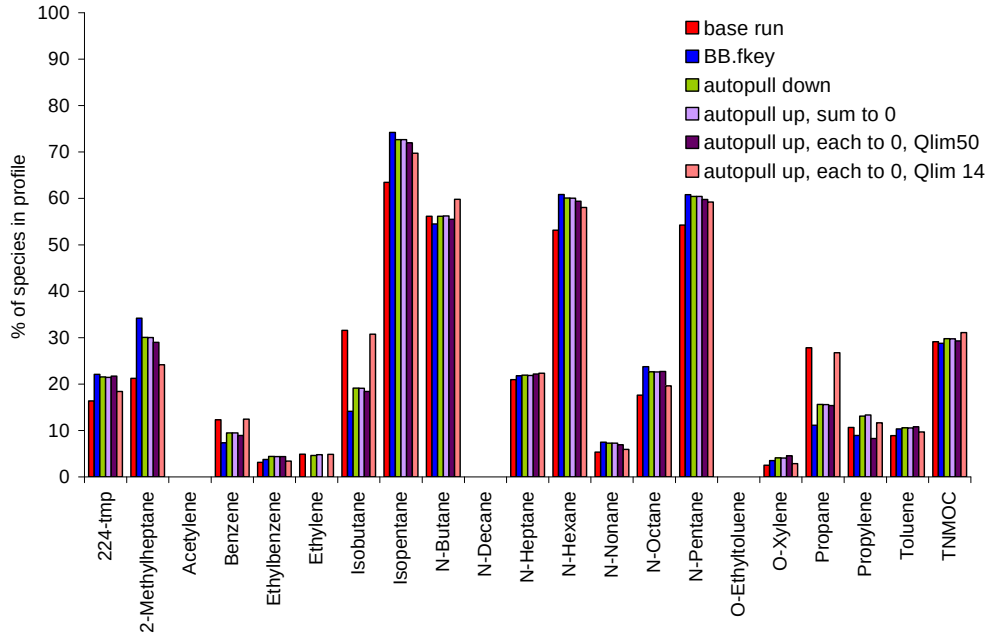


Figure 20. *Profile (% of species) for factor 3, liquid/unburned gasoline, through the base run, using BB.fkey to pull acetylene, and using autopull to pull acetylene down and up (where the sum of elements is zero, where each element is set to zero with a Qlim of 50, and where each element is set to zero with a Qlim of 14).*

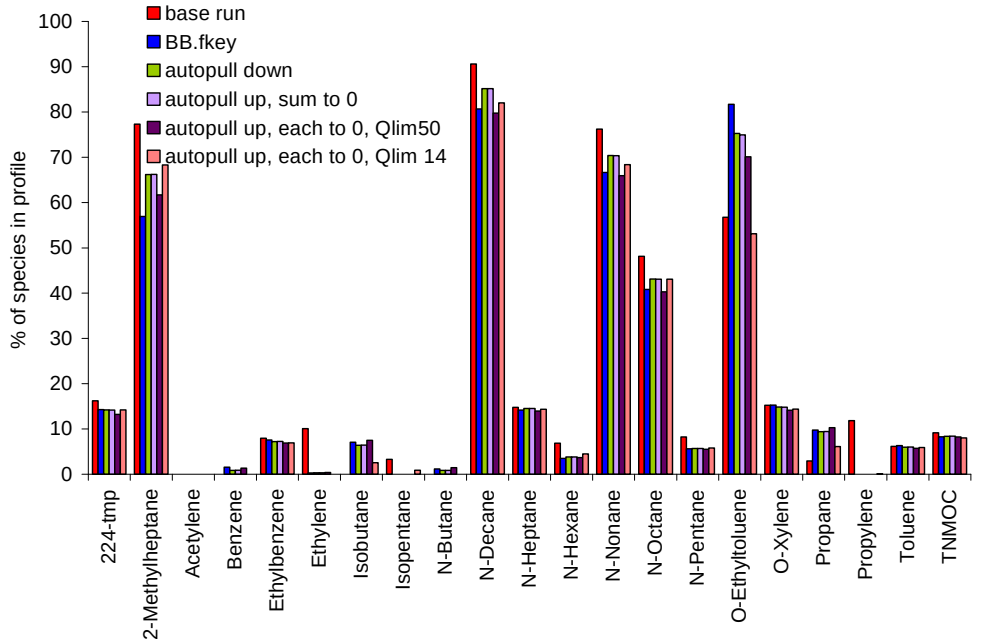


Figure 21. *Profile (% of species) for factor 4, diesel exhaust and heavy fuel use, through the base run, using BB.fkey to pull acetylene, and using autopull to pull acetylene down and up (where the sum of elements is zero, where each element is set to zero with a Qlim of 50, and where each element is set to zero with a Qlim of 14).*

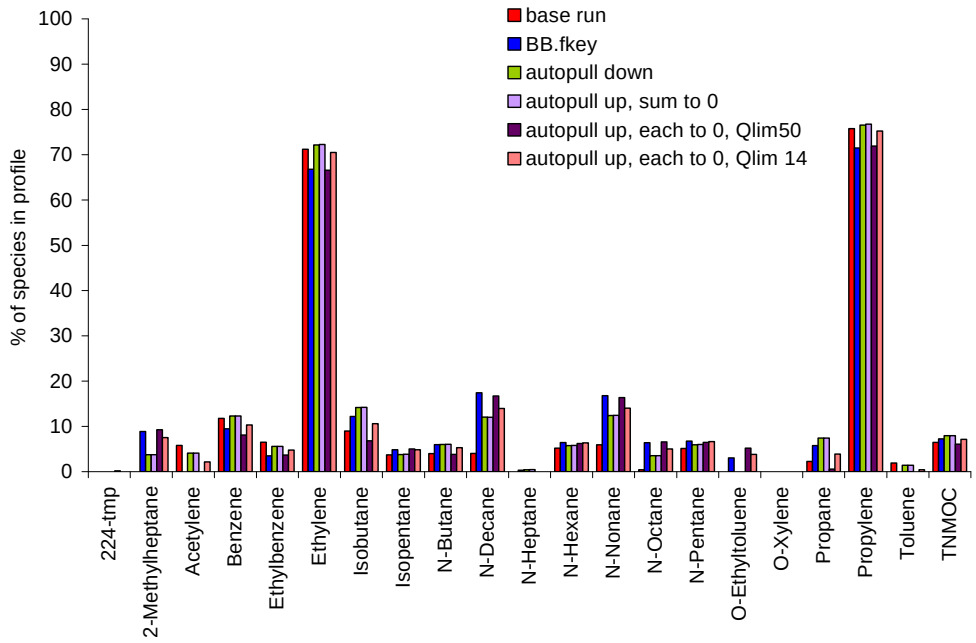


Figure 22. *Profile (% of species) for factor 5, fresh industrial emissions, through the base run, using BB.fkey to pull acetylene, and using autopull to pull acetylene down and up (where the sum of elements is zero, where each element is set to zero with a Qlim of 50, and where each element is set to zero with a Qlim of 14).*

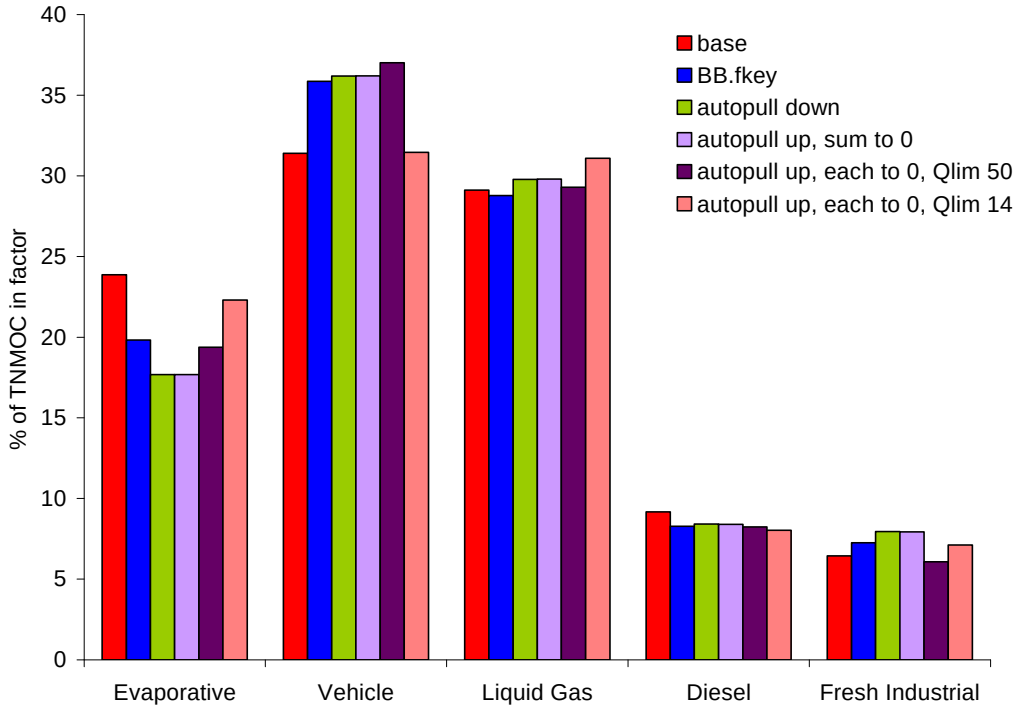


Figure 23. *Percent of TNMOC apportioned by factor through the base run, using BB.fkey to pull acetylene, and using autopull to pull acetylene down and up (where the sum of elements is zero, where each element is set to zero with a Qlim of 50, and where each element is set to zero with a Qlim of 14).*

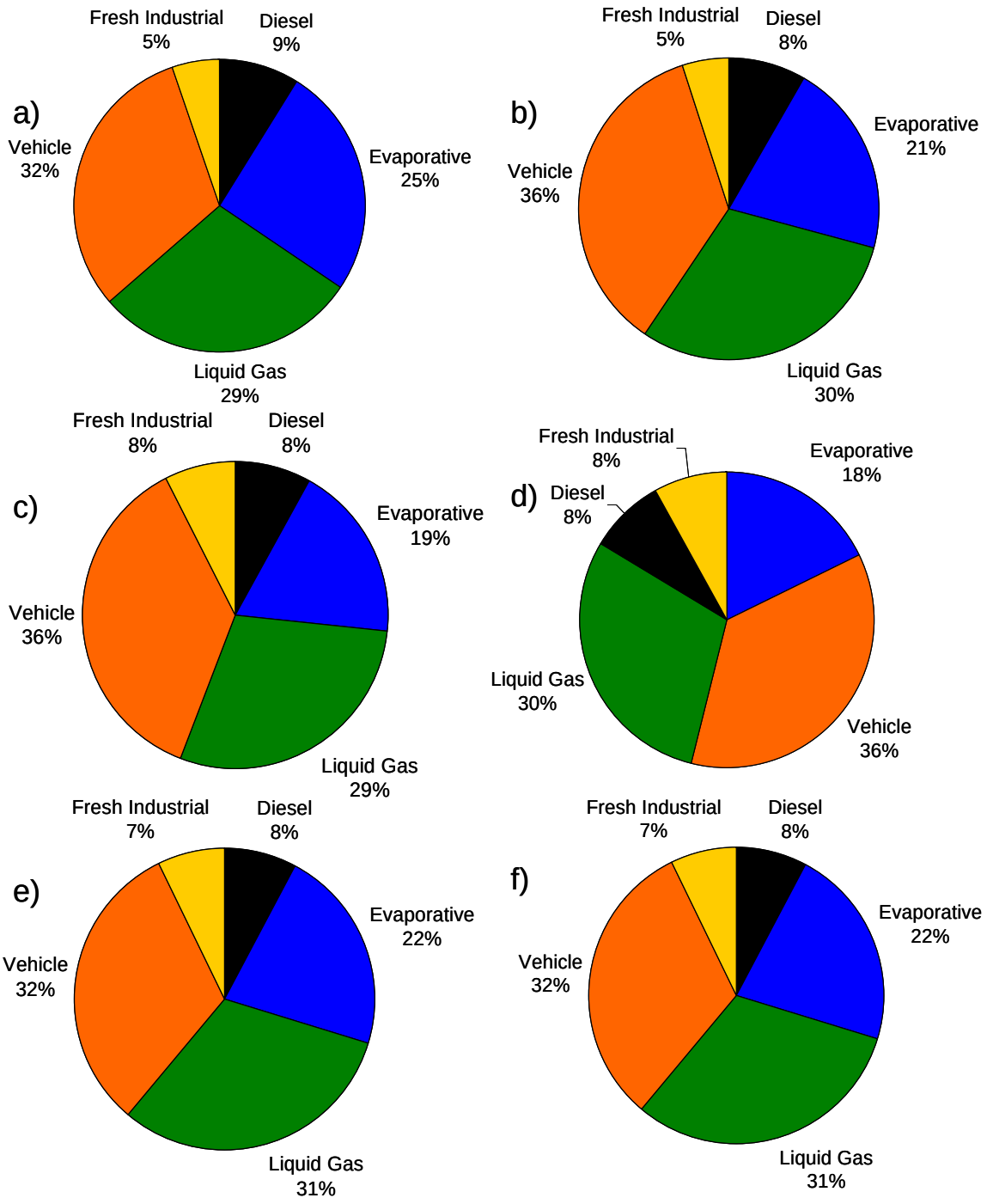


Figure 24. *Percent of TNMOC apportioned by factor through the base run (a), using BB.fkey to pull acetylene (b), and using autopull to pull acetylene down (c) and up (where the sum of elements is zero [d], where each element is set to zero with a Qlim of 50 [e], and where each element is set to zero with a Qlim of 14 [f]).*

6.0 APPENDIX A – OVERVIEW OF USING EXAMPLE DATA SETS

Below are steps for running ME-2 with the ini files and example data sets provided with this document and the user's ME-2 key:

1. Copy the files from the ME-2 CD onto your hard drive, such as in C:\ME2.
2. Transfer the example dataset files provided with this document into this folder as well.
3. To run an example, put the ini and data files from the IL ini files folder such as Baton Rouge Example\Baton Rouge IL ini files (BR_baserun.ini, BR_PMFData.txt, and BR_ab_base.dat) into the folder with ME-2 exe.
4. Open a command prompt by going to Start-Run-"cmd"
5. Navigate to C:\ME2, to access C: type "cd.." until you are at C: and use dir or dir/w to list directories, to change directories once in C, type "cd ME2"
6. To run ME-2, type the name of the exe followed by the name of the ini, e.g., me2wG17 BR_baserun.
7. This will output the base case runs as BR_base into .dat, .rsd, and .txt files. Check results by opening BR_base.txt and comparing to the file provided in the case study example folder where you got the ini file from.

Below are instructions to run ME-2 with the PL associated with EPA PMF for the example data sets provided with this document.

1. Create a directory and put all needed files here
    a. Make a new folder, such as C:\EPA_ME2
    b. Copy the following files from C:\Program Files\EPA PMF 3.0: the EPA PMF script (PMF_bs2.ini), the key associated with EPA PMF (e.g., me2key.key), the exe associated with EPA PMF (e.g., me2wopt.exe).
    c. Copy the folder "St Louis control files" that comes with the guidance document into C:\EPA_ME2.
2. Run base case: as described in the iniparams file, this will initiate 20 runs which will be output to the St_Louis_base files.
    a. Copy iniparams_StLouis.txt from the St Louis example\St. Louis PL control files folder into C:\EPA_ME2 and rename as iniparams.txt
    b. Copy StLouis_Data.txt from the St. Louis PL control files into C:\EPA_ME2; this file contains the St. Louis concentration and uncertainty data. The first half of the data is concentrations and the second are the associated uncertainties. The StLouis_species_sampledatetime.xls file contains the species labels and sample date and time.
    c. Begin a command prompt by going to Start-Run-"cmd"
    d. Navigate to C:\EPA_ME2, to access C: type "C:", to change directories once in C, type "cd\EPA_ME2"
    e. Run ME-2 by typing "me2wopt PMF_bs2.ini". Here me2wopt is the executable, followed by the initialization file (PMF_bs2.ini). This will automatically invoke the parameters chosen in iniparams.txt. Once the run is complete, the iniparams.txt file is automatically deleted. To initiate another run, the user will need to create another iniparams.txt file here as detailed in step a shown above.

f. Base run results will be stored in StLouis_base.dat, StLouis_base.rsd and StLouis_base.txt files. The dat and rsd files are for internal program use and friendly to the eyes. The txt file contains output in a user friendly format for each of the random runs (profiles, contributions, and other diagnostics). Check results by opening the StLouis_base.txt file, compare to the file provided in the case study example folder "St Louis iniparams/base run" where you got the iniparams file from.

3. Run autopull AA run: This will pull contributions in Factor 3 from run 5 to zero with a dQ of 10.

   a. Copy iniparams_AutoAA_10.txt and moreparams_StLouisAutoAA_10.txt from the St Louis control file folder into C:\EPA_ME2 and rename the files as iniparams.txt and moreparams.txt, respectively. Do not delete the base run files from the directory.

   b. Open iniparams.txt file and enter the desired base run number under the label “numoldsol”. This specifies the run on which the autopull will be performed. Typically this is the run with the lowest Qrobust value (listed at the start of each run). For this example the random run seeds are fixed so the default run in the iniparams.txt file has already been set to the lowest Qrobust value run. See Table 3 for guidance on additional parameters in the iniparams.txt file.

   c. Run ME-2 again as specified in step 2e. The program creates the same outputs as listed in step 2f except the files will be named StLouis_autopull_10 with the appropriate extensions. The user need not type anything special to alert the model to the existence of the moreparams.txt file. The parameter values in iniparams.txt and existence of moreparams.txt file is sufficient.

   d. Check results before opening the StLouis_AA_10.txt file, compare to the file provided in the case study example folder "Autopull AA_10" where you got the iniparams file from.

4. To run additional case studies, copy the iniparams and moreparams files from the case study folder (e.g., masked AAfkey). Delete the existing moreparams file. Rename the copied files to iniparams and moreparams, and run.

Instructions for generating example dataset graphs from ME-2 output are provided below.

As discussed in Section 2.3, ME-2 generates output files that can be used in a spreadsheet program, such as MS Excel, to calculate statistics or graph output. Both the raw .dat and .txt files contain the **AA** and **BB** matrices, which usually correspond to factor contributions and profiles, respectively. In both the .dat and .txt files, the **AA** matrix is the first set of values (rows representing samples, columns representing factors), followed by a blank line, followed by the **BB** matrix (rows representing species, columns representing factors). The species in the **BB** matrix are listed in the same order as in the concentration input file provided by the user. For the example datasets included in Section 5 of this document, no alterations to these output files were performed; the data used to generate the graphs were used in their original output format. The graphs shown for the example datasets are not exhaustive; users should use their own discretion to illustrate their results in the manner they deem most effective.

a. **Contributions** (normalized and mass): The first portion of the .dat and .txt output files contains the **AA** (contributions) matrix, with the normalized contribution of each factor (columns) to each sample (rows). The user may wish to produce scatterplots of normalized factor contributions from the base run (one column from the **AA** matrix) along with normalized contributions from various model runs (the corresponding **AA** matrix column from autopull, masked runs etc.) for each factor. Plotting mass contributions instead of normalized contributions requires multiplying the normalized contributions in the **AA** matrix by the total mass for each factor found in the factor profile (the total mass row in the **BB** matrix, if total mass is included as a species). An alternative to using total mass as a species is to regress the contribution of each factor against the total mass, using the coefficient for each factor to

then scale the normalized factor contributions to mass units (Section 2.2.4). Contributions are allowed to go slightly negative to allow leeway when modeling points near zero; these can be viewed by the user as essentially zero.

b. **Profiles** (mass and mass fraction): The second portion of the .dat and .txt output files (one can use either) contains the **BB** (profiles) matrix, showing the species mass (rows) apportioned to each factor (columns). If one has included total mass as a species, it will appear as a species in each factor. The user may find it helpful to plot these profiles in bar chart form, with mass on the y-axis and factors on the x-axis; using a logarithmic scale can highlight changes between runs, although users should be careful to remove zero values from the **BB** matrix in their spreadsheet before doing so. If users wish to plot mass fraction instead of mass (note that this was not carried out on the included example datasets), they should divide each species by the total mass apportioned to that factor. The total mass in one factor will be the sum of one column in the **BB** matrix, unless total mass is included as a species, in which case it will be included as a one element in a row of the **BB** matrix).

c. **Total mass apportioned to each factor**. Once total mass has been determined for each factor the total mass apportioned to each factor can be graphed. The user may produce, for example, a pie chart or a stacked bar chart illustrating total mass in each factor. Pie charts or stacked bar charts can be produced for each model run to illustrate any differences between runs.

United States
Environmental Protection
Agency

(8101R)
Washington, DC 20460

Official Business
Penalty for Private Use
$300

EPA 600/R-09/032
April 2009
www.epa.gov