United States
Environmental Protection
Agency

Office of Air Quality
Planning and Standards
Research Triangle Park NC 27711

EPA-450/3-78-040
August 1978

Air

EPA

Carbon Monoxide Hot Spot Guidelines

Volume VI: User's Manual for the Modified ISMAP Model

EPA-450/3-78-040

Carbon Monoxide Hot Spot Guidelines
Volume VI: User's Manual for the Modified ISMAP Model

by

Frank Benesh

GCA Corporation
GCA/Technology Division
Burlington Road
Bedford, Massachusetts 01730

Contract No. 68-02-2539

EPA Project Officer: George J. Schewe

Prepared for

U.S. ENVIRONMENTAL PROTECTION AGENCY
Office of Air, Noise, and Radiation
Office of Air Quality Planning and Standards
Research Triangle Park, North Carolina 27711

August 1978

This report is issued by the Environmental Protection Agency to report technical data of interest to a limited number of readers. Copies are available free of charge to Federal employees, current contractors and grantees, and nonprofit organizations - in limited quantities - from the Library Services Office (MD-35), U.S. Environmental Protection Agency, Research Triangle Park, North Carolina 27711; or, for a fee, from the National Technical Information Service, 5285 Port Royal Road, Springfield, Virginia 22161.

This report was furnished to the Environmental Protection Agency by GCA Corporation, GCA/Technology Division, Burlington Road, Bedford, Massachusetts 01730, in fulfillment of Contract No. 68-02-2539. The contents of this report are reproduced herein as received from GCA Corporation. The opinions, findings, and conclusions expressed are those of the author and not necessarily those of the Environmental Protection Agency. Mention of company or product names is not to be considered as an endorsement by the Environmental Protection Agency.

Publication No. EPA-450/3-78-040

ABSTRACT

A modified version of the ISMAP model has been developed for the analysis of carbon monoxide hot spot locations. The original version of ISMAP (Indirect Source Model for Air Pollution) was developed by Stanford Research Institute to be used in the evaluation of indirect source impact upon ambient carbon monoxide concentrations. Due to the size and complexity of the original version of the model, the modifications were made without changing the basic structure of the program. Since the analysis of parking lot traffic and emissions was an integral part of the original model structure, it was left in the modified version, but the code was modified to negate their effect upon traffic flow and air quality. In the modified ISMAP, internal zones (indirect sources) are used together with external zones to generate flow within a traffic network. This modified version of the model sets parking lot emissions equal to zero and provides a near infinite parking lot capacity so that vehicle routing between internal zones will not occur because of parking lot overcrowding. Other modifications made to ISMAP include the incorporation of a street canyon submodel. Version 2 of modified ISMAP utilized the most recent (1978) motor vehicle emission correction factors and modal analysis model coefficients and deterioration. In other aspects, it is unchanged from Version 1, written by Michael T. Mills.

PREFACE

This document is the sixth in a series comprising the Carbon Monoxide Hot Spot Guidelines. The purpose of this series is to provide state and local agencies with a relatively simple yet accurate procedure for assessing carbon monoxide hot spot potential on urban street networks. Included in the Hot Spot Guideline series are:

Volume I:	Techniques
Volume II:	Rationale
Volume III:	Summary Workbook
Volume IV:	Documentation of Computer Programs to Generate Volume I Curves and Tables
Volume V:	Intersection-Midblock Model User's Manual
Volume VI:	Modified ISMAP User's Manual
Volume VII:	Example Applications at Waltham/Providence/Washington, D.C.

CONTENTS

Abstract iii
Figures vi
Tables vii

1. Introduction 1
2. Model Description 3
3. Program Documentation 19
4. Input Data Specification 36
5. Sample Application 54
6. Use of the Modified ISMAP Model in Conjunction with APRAC 70
Appendix A. Program Listing 74
References 221

FIGURES

Number		Page
1	ISMAP flow diagram	4
2	Flow chart of vehicle flow model	5
3	Simple model of intersection delay	7
4	Flow diagram of link emission calculations	10
5	Flow diagram of link dispersion calculations	13
6	Schematic of cross-street circulation between buildings	16
7	Specification for leeward and windward cases on the basis of receptor location, street orientation, and wind direction	17
8	Route vehicles model flow diagram	21
9	Signalized intersection model flow diagram	24
10	Parking model flow diagram	27
11	Subroutine DISPERS flow diagram	30
12	Diagram for test case of modified ISMAP	55

TABLES

Number		Page
1	Estimates of Pasquill-Turner Stability Categories for Urban Areas Computed From Airport Weather Observations	12
2	Modified ISMAP Basic Input Information Cards 0 Through 10 (Data Read in by Subroutine INPT)	37

ACKNOWLEDGMENTS

We wish to acknowledge the significant contributions made early in the development of the Modified-ISMAP Model by previous GCA/ Technology Division staff members, including Dr. Michael Mills and Mr. Victor Corbin. We are also indebted to the EPA Project Officer, Mr. George Schewe and to Ms. Nancy Mayer of the Source Receptor Analysis Branch, Who performed technical and editorial Review of the final reports and computer programs. We also thank SRI International for the use of ISMAP and Mr. Erik Sieurin of the Source Receptor Analysis Branch for his computer assistance in updating and finalizing the model.

SECTION 1

INTRODUCTION

This manual describes a version of the Stanford Research Institute ISMAP Model, which has been modified by GCA/Technology Division for the evaluation of carbon monoxide hot spot locations. The purpose of the original version of ISMAP (Indirect Source Model for Air Pollution)[1] was the analysis of the effect of indirect sources such as shopping centers and sports stadiums upon ambient carbon monoxide concentrations in the surrounding area. In modifying the model for application to hot spot analysis, GCA has de-emphasized the role of parking lots while leaving the zone attractions still in the model. Also, street canyon simulation and an emission factor correction routines have been added. The basic structure of the model was, however, not altered. It still provides a convenient mechanism for evaluation of the effects of changes in traffic patterns, zone attractions, signal parameters and roadway characteristics upon carbon monoxide concentrations at selected points.

The basic driving force in the ISMAP model are the trip attractions and productions and their specified distribution among exterior and interior traffic zones. Interior zones are considered to be parking areas surrounded by links of the network. An exterior zone is defined to be an exterior street by which vehicles enter or leave the network. Travel from the network to any interior zone is required to take place through a gate, defined by two links in the network. Based upon link capacities, link to link connections, intersection control type and intersection approach capacities, minimum travel times are calculated for each gate-link combination. This information is in turn used to determine the number of vehicles traveling between an external and internal zone over a number of alternative routes. The program has the capability of handling traffic volumes not generated by zone attractions.

Once the link traffic volumes have been generated by the program, average queue lengths and delay times for each intersection approach are determined based upon signalization parameters. By use of the EPA Modal Analysis Model,[2] emission rates for cruising, idling and acceleration-deceleration are assigned to the appropriate sections of the traffic links. These emissions together with hourly meteorological data are input to a line source diffusion model for calculation of hourly carbon monoxide concentrations at selected receptor locations.

Three primary revisions or additions were made by GCA to change the focus of ISMAP from indirect source review to hot spot analyses. These included: (1) removal of parking lots in terms of their effects on emissions and traffic

flow while still retaining the concept of interior zones, (2) the use of new emission factors, and (3) the incorporation of a street canyon submodel.

In the version of ISMAP developed by GCA, the distinction between internal zones and external zones has been removed, so that internal zones no longer serve as parking lots but simply serve to attract and produce vehicle trips, as do the external zones. The only distinction that remains between internal and external zones is the numbering system that is used to designate each.

Emission factor correction capabilities have been added to permit the user to account for local conditions such as vehicle-age distribution, vehicle-type mix, ambient temperature, region, inspection-maintenance, and percentages of vehicles operating in the cold or hot start mode. These correction factors are computed within the program utilizing the user's specific input parameters, and are applied automatically. The subroutine from which the correction factors are derived was developed from MOBILE1.[14]

The user also has the option of utilizing a street canyon diffusion model for analyzing receptors located in street canyons. The particular subroutine available is from the APRAC diffusion model.

One additional modification to the original ISMAP model that was made by GCA was to convert the program code from CDC (Control Data Corporation) FORTRAN to a FORTRAN code compatible with all other computer systems (IBM, UNIVAC, etc.), thereby enhancing the program and increasing its availability.

In the next section of this manual we shall describe in more detail the modified version of ISMAP. This will be followed in Section 3 by a documentation of the individual subroutines in ISMAP. Much of the material for these two sections has been extracted from the original SRI ISMAP User's Manual. The remaining sections contain input data specifications, sample model application, discussion of the use of the program in conjunction with APRAC-1A, a source listing of the program and finally a list of references.

SECTION 2

MODEL DESCRIPTION

The model incorporates three logical submodels for the simulation of (1) traffic flow, (2) vehicle emissions, and (3) pollutant dispersion that, when operated sequentially, produce estimates of air pollution on a fine scale. The functions of the three submodels of ISMAP, and their component subroutines, are described in this section and are illustrated diagrammatically in Figure 1.

TRAFFIC FLOW ANALYSIS

The traffic flow submodel is rerun for each time period that is specified (see flow chart, Figure 2). A time period is specified as an input parameter and a situation is modeled during this time. Accumulations of vehicles in interior zones and queues on streets are retained and during subsequent time periods these act as initialization data for the subsequent iteration. When running a second time period, a new time of day, day of week, number of trip attractions, and number of trip generations must be specified.

Vehicles will select routes between interior and exterior zones that tend to minimize travel time. ISMAP does fine-scale routing of traffic in two steps. In the first step alternative routes from each origin zone to each destination zone are selected. This is done by finding the minimum travel time and route from each zone to each entrance/exit gate and the minimum travel time from each gate to each destination zone. In this process a route is generated between each origin and destination zone and through each gate. Additional routes are generated to a second alternate access link to each interior zone. The four alternative routes that show the minimum travel time are selected from this set.

The second step in the fine-scale routing of traffic is to reassign vehicles to alternate routes between each origin and destination zone. The number of vehicles (v) routed through gate (g) is directly proportional to gate capacity (C_g) and indirectly proportional to the total number of vehicles using the gate (V_g) and the total vehicle trip travel time (TT_g), i.e.:

$$v = K \cdot \frac{C_g}{V_g \cdot TT_g} \cdot T \qquad (1)$$

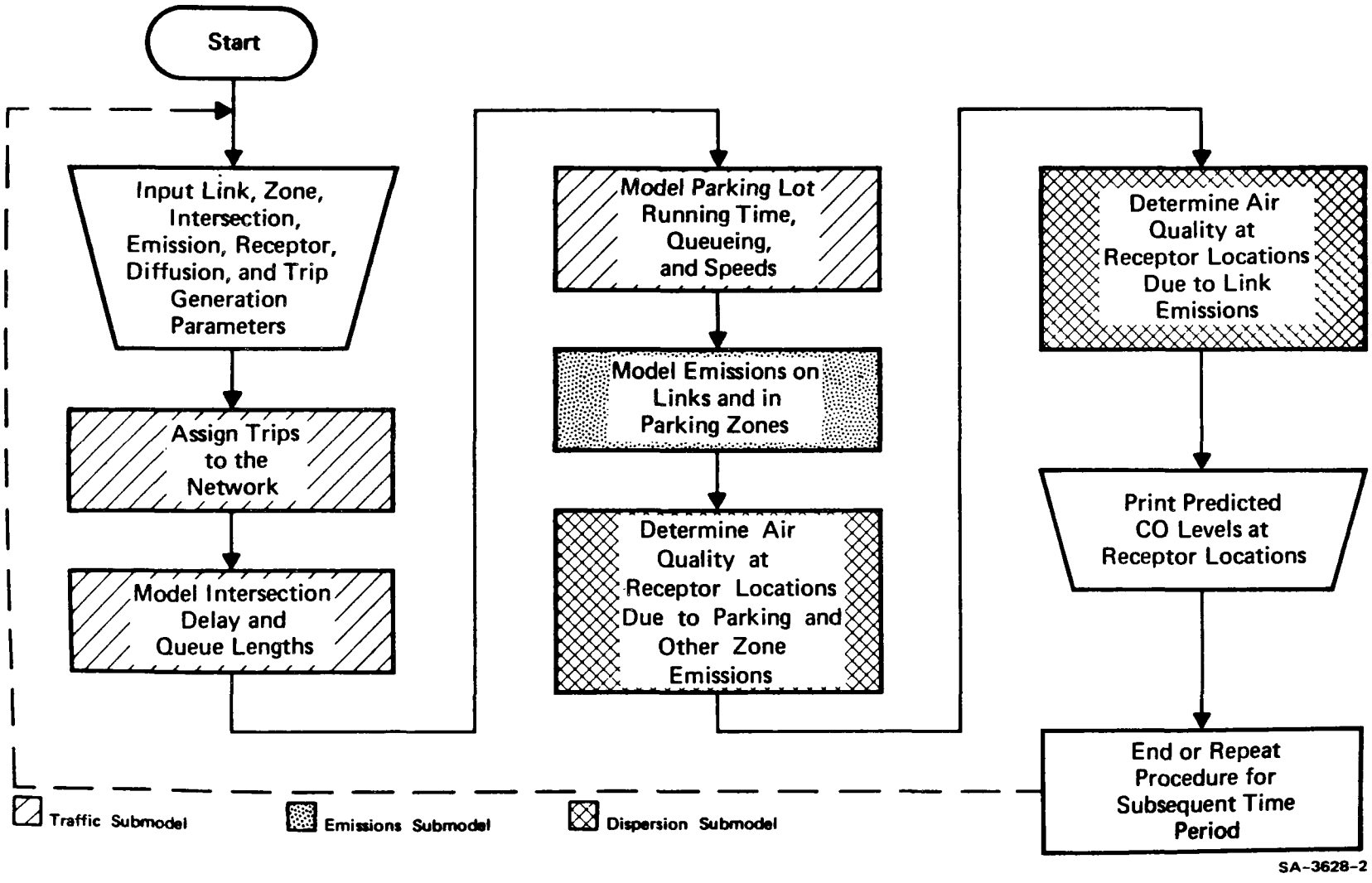

Figure 1. ISMAP flow diagram.

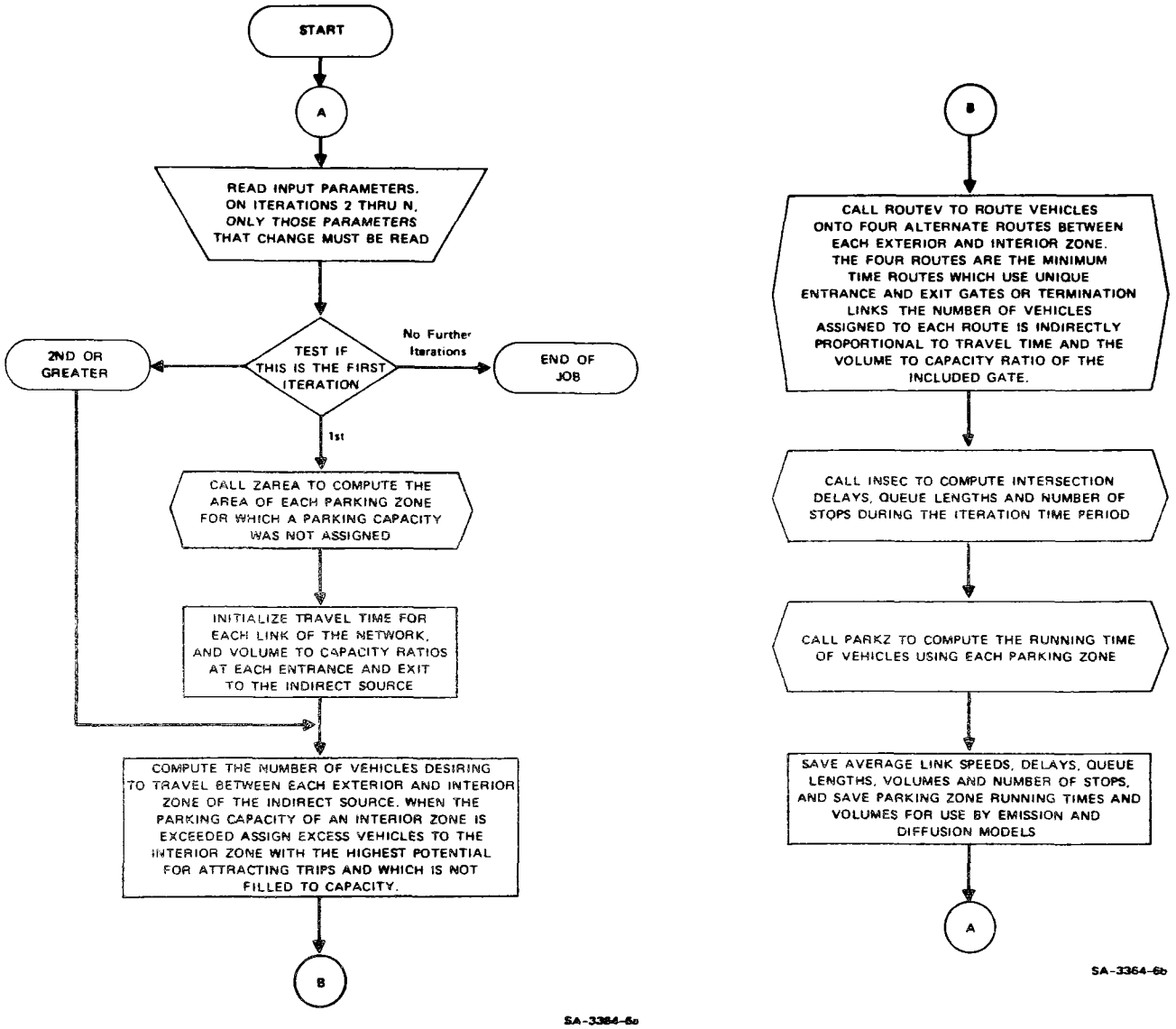

Figure 2. Flow chart of vehicle flow model.

where T is the total trips being made between the origin and destination zone; and

K is a constant insuring that the sum of the v's is equal to T.

Traffic is assigned to the alternate routes on the basis of time and capacity; the resulting travel times and other measures of effectiveness, such as stops, delay, queue length, and flow, are then calculated. The delays at each intersection are calculated by a subroutine that models an intersection on the basis of traffic flow and physical characteristics. The outputs of the subroutine are average vehicle delay and average queue length for each approach and turning movement at the intersection.

The method of computing delay at a signalized intersection is based on an approximate method of computing delays and queues[3] in which the discrete nature of the cars is disregarded in favor of considering traffic as a continuous fluid, which arrives at a uniform rate (q), is dammed for a time (R), and is then released at a rate (s) until the dam is empty. Traffic therefore moves out of the intersection at the arrival rate (q) provided the green time (G) is long enough.

In Figure 3, the variable v(t) is the vehicle arrivals over time t. The total delays for all vehicles during any time interval are represented by the area bounded by the arrival and departure flow rates. The average delay (d) per vehicle is:

$$d = \frac{s \cdot R^2}{2 \cdot (R + G) \cdot (s - q)} \quad . \tag{2}$$

Actually, vehicles rarely arrive at a uniform rate but tend to arrive in random small groups. In the model, a Poisson distribution has been assigned to this characteristic, and, when such arrivals occur, queues of vehicles form waiting to be serviced by the intersection. A second element of delay (i.e., queueing delay) must be added to the basic delay (d) caused by the gating of vehicles. This delay is based on the average queue length (Q), which is given by the equation:

$$Q = \frac{q}{\left(2 \frac{G}{R + G} \cdot s - q\right)} \quad . \tag{3}$$

Then the average delay per vehicle is the delay due to the gating effect of the signal plus the time spent waiting in the queue of length Q, which is being serviced at a rate $(s \cdot G)/(R + G)$. Combining the delay per queued vehicle (which is the reciprocal of the service rate) with Equation 3 and combining this result with Equation 2, the average delay per vehicle including delay due to queueing (D) is:

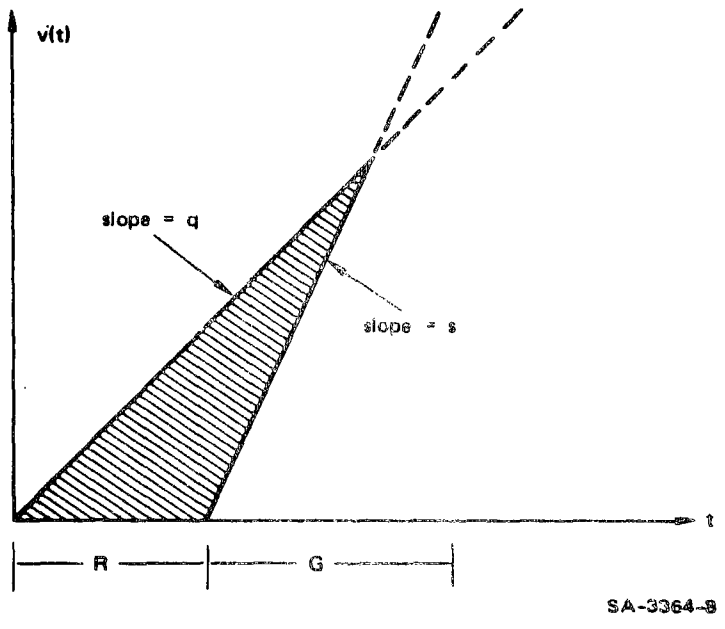

Figure 3. Simple model of intersection delay.

$$D = \frac{s \cdot R^2}{2 \cdot (R + G) \cdot (s - q)} + \frac{q \cdot (R + G)^2}{2 \cdot s \cdot G \cdot (s \cdot G - q \cdot [R + G])} \cdot \quad (4)$$

The intersection submodel will also predict delay and queue length at unsignalized intersections. As with a signalized intersection, a queue may be formed because of random arrivals at the intersection. If q is the arrival rate of vehicles at the intersection and C is capacity flow rate of vehicles departing the intersection, the average queue length is:

$$Q = \frac{q}{(C - q)} \quad . \quad (5)$$

The average delay experienced by vehicles at an unsignalized intersection is:

$$D = \frac{1}{C} \cdot \frac{q}{(C - q)} \cdot \quad (6)$$

The computing delay at an intersection requires predetermination of intersection approach capacity, signal phase lengths, and cycle times. The ISMAP intersection subroutine recomputes these variables for each simulated time period (usually 1 hour) as follows:

- Adjustments to intersection approach capacities are automatically made based on the number of left-turning vehicles at an intersection.

- Red and green phase lengths as well as cycle times are automatically computed at actuated signals.[4]

- Left-turn phases may be specified at a signalized intersection, and capacity for such phases may be input or automatically calculated based on the intersection approach capacity for the through and right-turning traffic.

MULTIMODAL AUTOMOTIVE EMISSIONS ANALYSIS

The emissions submodel of ISMAP is based on a method of calculating the amount of carbon monoxide (CO) produced by specified distributions of light-duty vehicles under four operating conditions: combinations of constant speed, acceleration, deceleration, and idling.[2] The formulation centers around an instantaneous emission rate $\dot{e}(t)$ that is a function of vehicle speed (v) and acceleration (a). Since speed and acceleration are functions of time, the emission rate function can be expressed as

$$\dot{e}(t) = \dot{e}[v(t), a(t)] \quad . \quad (7)$$

By assuming that the acceleration from/to a given speed is a perturbation to the steady state emission rate, two equations describe the emission rate:

$$\dot{e}(v,a) = b_1 + b_2 v + b_3 a + b_4 av + b_5 v^2 + b_6 a^2 + b_7 v^2 a + b_8 a^2 v + b_9 a^2 v^2 \quad (8)$$

$$\dot{e}(v,0) = b_{10} + b_{11} v + b_{12} v^2 \quad (9)$$

where Equation 8 describes the nonzero acceleration emission rate, and Equation 9 determines the steady-state emission rate.

Modified ISMAP first computes the emission rates for three of the four operating conditions (deceleration, acceleration, and cruise) for a 1977 LDV hot stabilized vehicle mix. A correction factor is then applied to adjust the emission estimates to the user specified scenario. The correction factor is basically the ratio of the MOBILE1 emission factor for the specified scenario and the MOBILE1 emission factor for a 1977 LDV hot stabilized vehicle mix. In the case of idle emissions, the MOBILE1 idle emission factor, adjusted for the specified scenario, is used directly.

The emissions model requires the following outputs of the traffic flow analysis for each link:

- Total number of vehicles traveling on the link
- The speed achieved during steady-speed operation
- Total number of vehicles stopped on link
- Total travel time of all vehicles on the link
- Total length of the link
- Acceleration rates.

These inputs are used to describe the CO emissions as a variable (time and distance) along each traffic link. This is done by determining the number of vehicles on each link that are delayed and the number of vehicles that are not delayed. The emissions are then calculated for each group, delayed and not delayed. The emissions for delayed vehicles are determined by calculating the amount of time spent on each link in one of the four operating conditions (constant speed, deceleration, idling, and acceleration) and multiplying this time, along with the number of vehicles involved, by the average emission rate for that specific operating condition. The emissions for those vehicles not delayed are simply the product of the total vehicle-seconds of steady-speed operation and the average emission rate for that speed. A variable emission rate of CO is determined for each link on the basis of this information. This rate varies with distance along the link and depends on the duration and type of each vehicle operating mode. A flow diagram of these calculations is given in Figure 4.

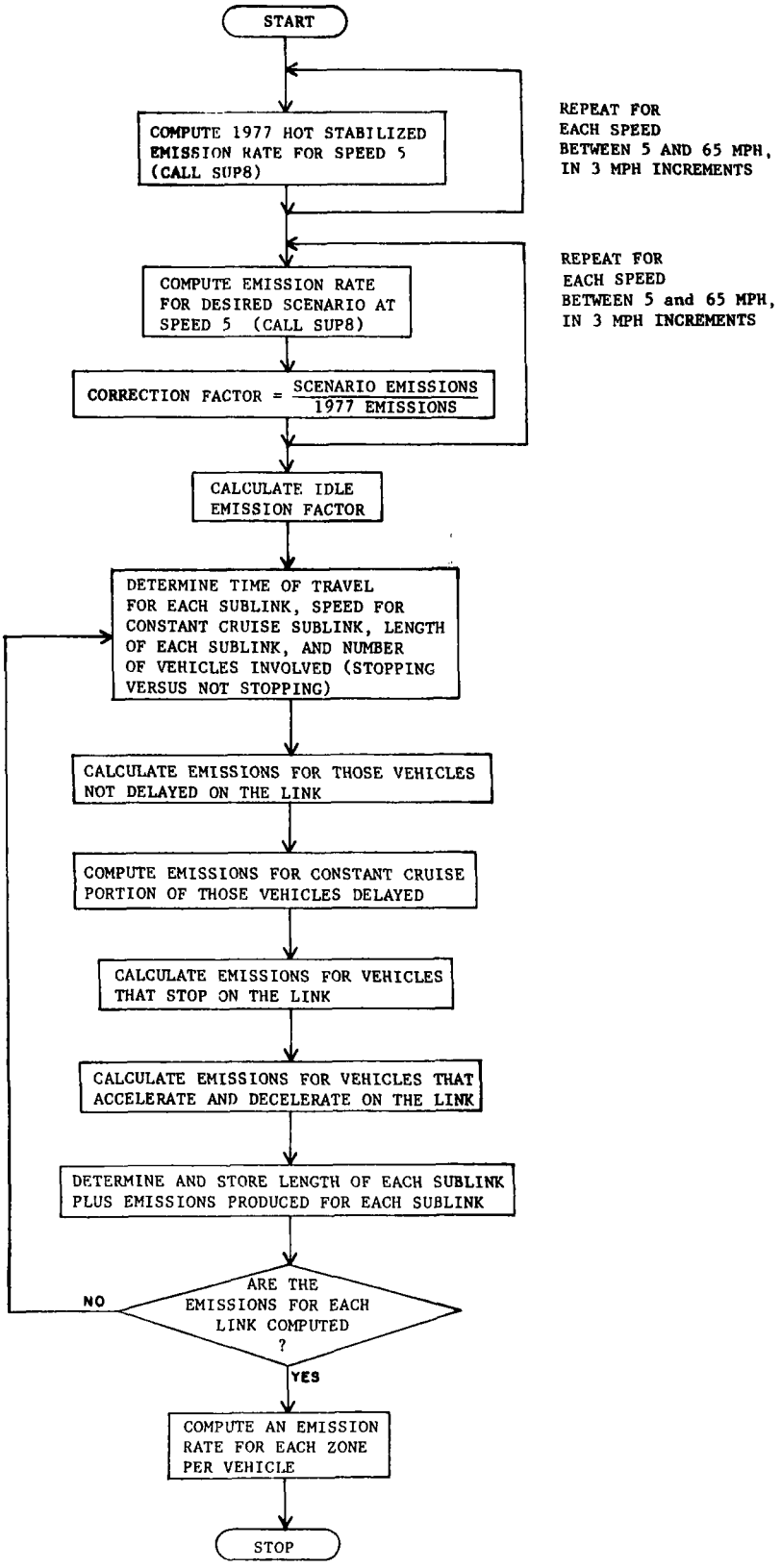

Figure 4. Flow diagram of link emission calculations.

ATMOSPHERIC DISPERSION ANALYSIS

The detailed description of the transport and diffusion of pollutant from individual traffic links necessary for microscale analysis of pollutant distribution is performed by the dispersion submodel portion of ISMAP developed by SRI for this program. Concentrations resulting from line source emissions are calculated separately and summed. Any wind direction may be used, including winds parallel to roadways.

Atmospheric stability may be specified as a model input, or the user may input wind and cloud cover data and allow ISMAP to determine the stability. ISMAP accomplishes this using a table based on modified Pasquill-Turner stability categories.[7] Atmospheric conditions are classified according to prevailing insolation strength and wind speed for daytime hours, and according to cloud cover and wind speed for nighttime hours, as shown in Table 1.

Figure 5 is a flow chart of the line source calculation procedure in the dispersion model. For each receptor, the location of each link is first tested to determine whether the link contributes to the concentration at the receptor for the input wind direction. This test is made by taking only those links that pass through, or whose end points lie within, a 90° segment extending upwind of the receptor and bisected by the wind vector. If the link does not fall into this pie-shaped slice, concentration calculations for it are bypassed and the next link is checked. If a link does extend into the segment, the model proceeds with calculation of pollutant concentration from the link.

Each traffic link is represented by a series of point sources whose total emissions equal the link's total emissions. A distance between points that will accurately represent the line source must be chosen. The maximum tolerable error resulting from the finite distance between point sources was taken to be 5 percent. Thus,

$$\frac{\chi_{ref} - \chi}{\chi_{ref}} = 0.05 \tag{11}$$

where χ_{ref} is the concentration at a receptor resulting from some reference point on the link, and χ is the concentration of a point on the link a distance ΔL away. The case in which the wind is perpendicular to the link is the most sensitive to the spacing between point sources (ΔL). Assuming the reference point to be on the centerline of the plume with the wind perpendicular to the link, Equation 11 can be solved for ΔL:

$$\Delta L = \sqrt{-2\ln(0.95)\ \sigma_y^2} \tag{12}$$

TABLE 1. ESTIMATES OF PASQUILL-TURNER STABILITY CATEGORIES FOR URBAN AREAS COMPUTED FROM AIRPORT WEATHER OBSERVATIONS

Surface winds (knots)	Daytime (SR* + 4 hours to SS† - 3 hours)			Early a.m. and late p.m. (SR + 1 to SR + 3 and SS - 2 to SS - 1)	Nighttime (SS to SR)	
	Strong insolation	Moderate insolation	Slight insolation		$\geq$ 5/10 clouds	$\leq$ 4/10 clouds
$\leq$ 3	1	2	2	4	5	5
3-6	1	2	3	4	4	5
6-10	2	3	3	4	4	4
10-12	3	3	4	4	4	4
$\geq$13	3	4	4	4	4	4

*SR = sunrise

†SS = sunset

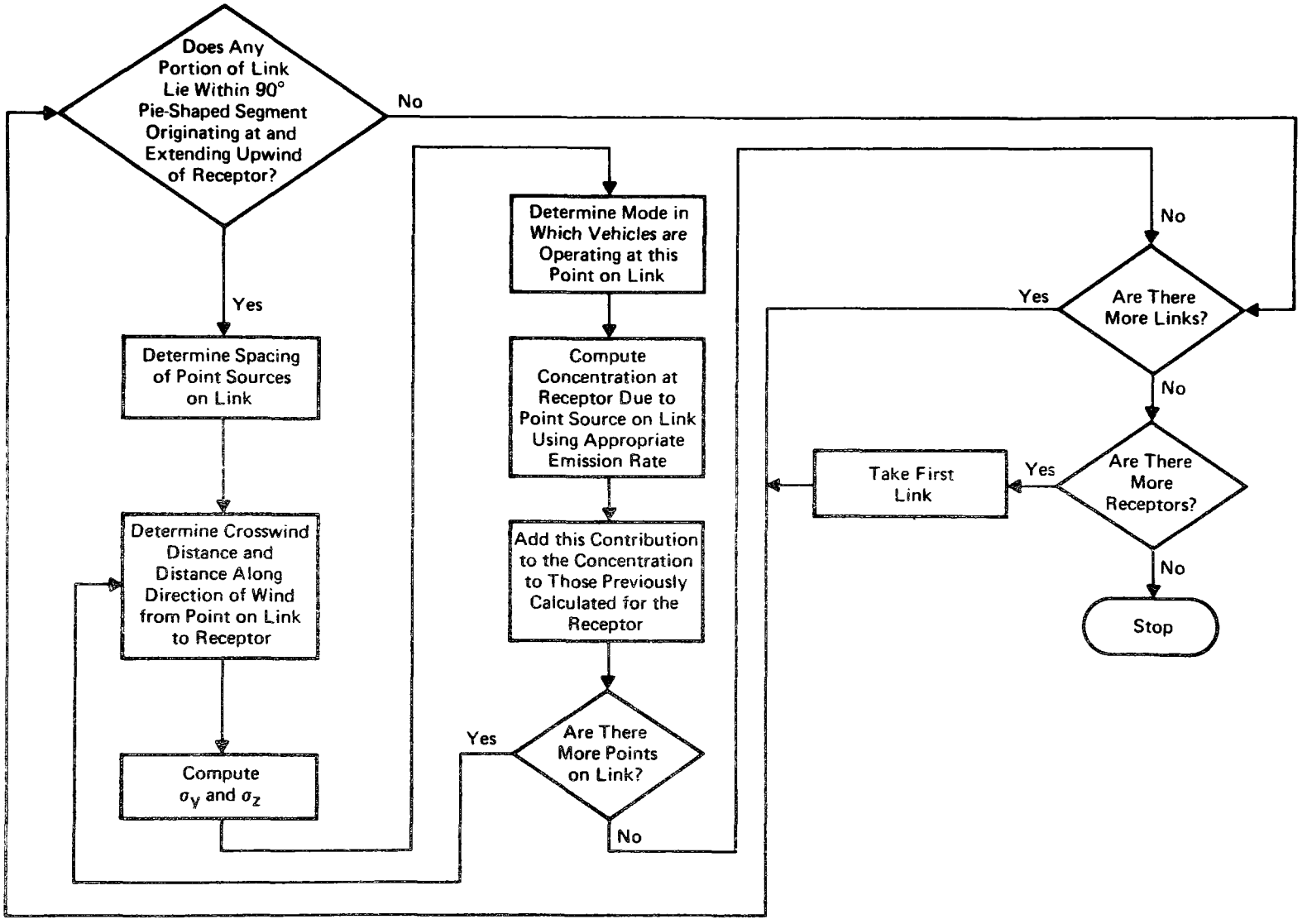

Figure 5. Flow diagram of link dispersion calculations.

The distance along the wind direction used for calculation of σ_y in Equation 12 is the distance from the receptor to the nearest point on each link. To be conservative, ISMAP uses one-half of the value of ΔL as the spacing.

Having found the spacing between point sources for the link, the model proceeds to calculate the CO concentration at the receptor for each point source. The equation for the CO concentration from a point source is

$$\chi_p = \frac{Q_p}{\pi\sigma_y(d_w)\sigma_z(d_w)u} \exp\left\{-1/2 \frac{d_c^2}{\sigma_y^2(d_w)}\right\} \exp\left\{-1/2 \frac{H^2}{\sigma_z^2(d_w)}\right\} \tag{13}$$

and

$$\begin{aligned} \sigma_y(d_w) &= a|d_w|^b + c \\ \sigma_z(d_w) &= f|d_w|^g + h \end{aligned} \tag{14}$$

where χ_p is the CO concentration, gm m^{-3}

u is the wind speed, m sec^{-1}

Q_p is the point-source emission rate, gm sec^{-1}

$\sigma_y(d_w)$ is the lateral standard deviation of plume concentration, m

$\sigma_z(d_w)$ is the vertical standard deviation of plume concentration, m

H is the emission height, m

d_c is the crosswind distance from point source to receptor, m

d_w is the distance along wind direction from point source to receptor, m

a,b,f,g are the diffusion coefficients and exponents

c,h are the constants representing initial diffusion, m

For each link, a coordinate transformation of the link end points and the receptor is required to allow the same point source equation formulation to apply to all receptor-link configurations. The crosswind and upwind distances in Equation 13 are then found. The diffusion coefficients and exponents are determined by the stability class: initial horizontal diffusion for links is determined by the point source spacing on the link but is not allowed to exceed the half-width of the link.

ISMAP uses variable emission rates along the link that correspond to different modes of vehicle operation; e.g., steady speed, idling, acceleration, deceleration. When the contribution to the concentration at a receptor from

each point source on the link is calculated, the emission rate used in the calcualtion is determined by the driving mode being practiced at the location of the point source.

The concentrations resulting from point sources on a link are summed for each receptor and multiplied by the distance between point sources. Then the individual link concentrations are summed to give the total line source concentration at the receptor.

STREET CANYON CALCULATION

The modified ISMAP program provides the option of a special dispersion calculation for those receptors situated near urban roadways with buildings on both sides. The Street Canyon Model[8] is used only for those link-receptor combinations for which both the link and receptor are associated with a street specified by the user as a street canyon <u>and</u> for which the following relationship holds on an hourly basis:

$$H > 7\sqrt{\frac{KW}{u}} \tag{15}$$

where H = building height (m)

W = street canyon width (m)

u = wind speed (m/sec)

K = turbulent diffusivity (m^2/sec)

Values for K are 25.5, 5.5, 1.75, 1.0, 0.5 and 0.5 m^2/sec for atmospheric stabilities 1 through 6, respectively. Once both of the conditions for a street canyon configuration are met, the concentration assigned to a street canyon receptor will depend upon which side of the street the receptor is located and the direction of the wind with respect to the street orientation. The assignment of a receptor to either the windward or leeward side of the street canyon is illustrated by Figures 6 and 7. For a receptor located at the leeward side of the street canyon the concentration, C_L (g/m^3), is given by

$$C_L = \frac{\zeta Q}{(u + 0.5)\left[\left(\frac{W^2}{4} + z^2\right)^{1/2} + L_o\right]} \tag{16}$$

where ζ = empirical nondimensional constant ($\simeq$ 7)

Q = line source emission rate (g/m/sec)

L_o = approximate vehicle size ($\simeq$ 2m)

The windward concentration, C_W (g/m^3), is calculated by the following expression:

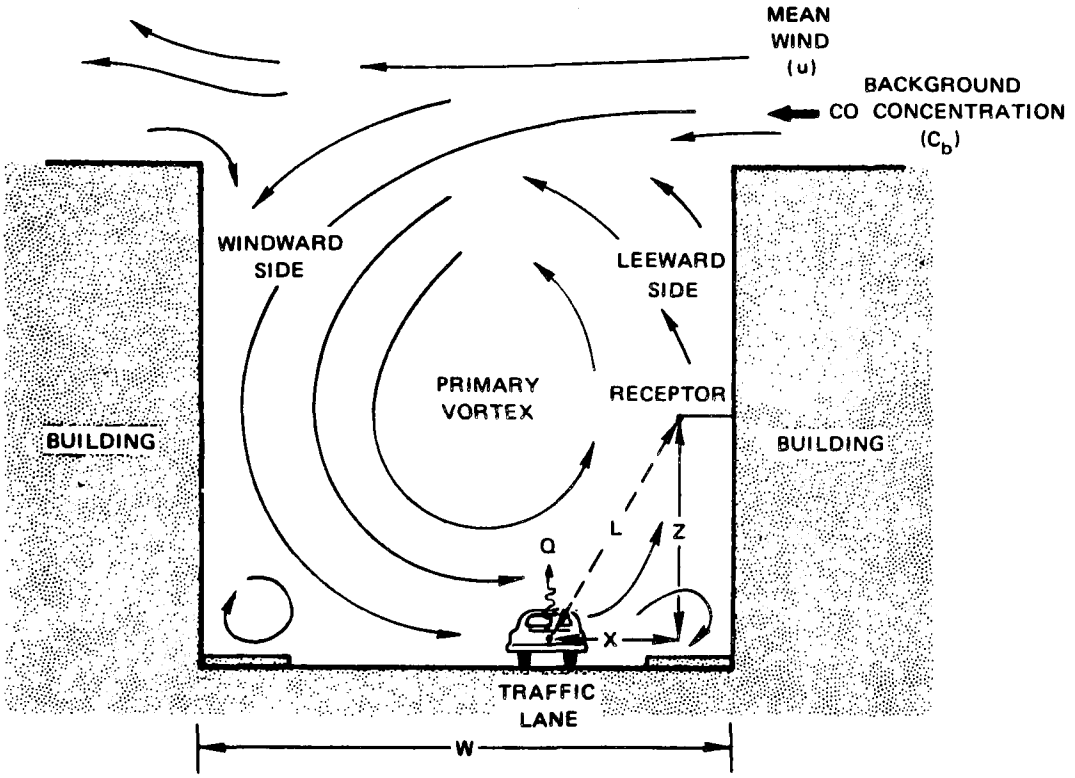

Figure 6. Schematic of cross-street circulation between buildings.

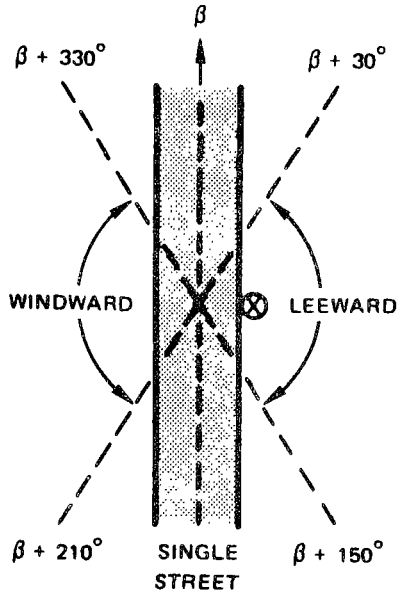

Figure 7. Specification for leeward and windward cases on the basis of receptor location, street orientation, and wind direction.

$$C_W = \frac{KQ\ (H-z)}{W\ (u + 0.5)\ H} \tag{17}$$

If the wind direction for a given hour does not place the receptor either on the windward or leeward side of the street canyon then the concentration is given by

$$C_I = \frac{1}{2}\ (C_L + C_W) \tag{18}$$

The emission rate, Q, which is used in the Street Canyon Submodel is chosen to be the emission rate due to one of the four modes: cruising, deceleration, idling and acceleration. Each one of these modes is assigned to a different portion of the link. The emission rate which is selected corresponds to that travel mode whose segment centroid lies closest to the receptor location.

SECTION 3

PROGRAM DOCUMENTATION

This section will describe the operation of the ISMAP main program and subroutines. Although a number of the subroutines deal with vehicle routing through parking lots and parking lot area emissions, we present a discussion of these functions for the sake of completeness even though the effects of parking lots upon link volumes and interior zone emissions have been negated within the modified ISMAP.

ISMAP (MAIN PROGRAM)

The ISMAP program acts as a calling program for the various subroutines. The flow chart shown in Figure 1 indicates the calling sequence of subroutines in the traffic flow portion of the model. This is essentially the flow chart of ISMAP except that the link emissions model, EMIT, and the emissions dispersion model, DISPER, are called at the end of the program.

SUBROUTINE INPT

This subroutine is responsible for reading all the data described in Section 4 Inputs Description. The simulation is normally terminated in this routine when an end of file card is read instead of a header card for the next iteration. The INPT routine prints out card images of the data read into the computer and the default values assigned by the program. One exception was made to this rule for card type 9 data. These data are printed out in meters with six coordinate pairs per card image, although it is read in feet as one coordinate pair per card image.

SUBROUTINE ZAREA (EFFECTIVELY BYPASSED IN THE MODIFIED VERSION OF ISMAP)

This subroutine is called to compute the area of a connected figure of up to six sides. The method used is to construct triangles from pairs of connected sides using each side only once. This is done by constructing a third side between the unconnected end points of each pair of connected sides. Two such triangles define the area of any four-sided figure. A five-sided figure is defined by three triangles, two triangles being constructed from the first two pairs of adjacent sides, and the third triangle being constructed by the fifth side of the figure and the two previously constructed sides. A six-sided figure is defined by four triangles; three of these are constructed from pairs of adjacent sides. The fourth triangle is constructed from the constructed sides of the other three triangles. The triangle areas so determined are added to or subtracted from the areas of the other triangles to determine the area of the figure.

Another function of this subroutine is to determine an estimate of the length (PLL) of a trip into the parking zone. The trip length is determined as the length of the longest side plus the computed area divided by the longest side.

SUBROUTINE ROUTEV

This routine calls the MINPTH subroutine to determine the minimum path between all gates and all other links in the network (see Figure 8 for flow chart). Minimum paths are saved in array RG, and minimum costs are saved in array CG for each gate. The minimum paths are based on travel times for each link by turning movement, which are stored in the TT array. The TT array is initialized with the travel lines that would be experienced if all vehicles traveled at the input velocity VEL on each link. Subsequent iterations of ROUTEV use a TT array based on intersection delays and link velocities as modeled by ISMAP.

For each zone, a minimum path and cost are determined to every link in the network and stored in arrays R and C, respectively. Four alternative routes are selected using the RG, CG, R, and C arrays. This is done by finding the minimum route to each gate using the R and C arrays and combining this with the minimum and second most minimum route from each gate to each destination zone. The minimum and second most minimum routes from each gate to each zone are determined from the minimum route and cost to each link that accesses each zone, which is stored in the RG and CG arrays. If four alternative routes are discovered, they may lead through as many as four gates, or as few as two gates, and may terminate on any link that accesses the destination zone. Of course, when less than four alternative routes are possible, only the number of possible alternative routes is generated. When there is no alternative route between external and internal zones, the program fails to apply traffic demand between these zones to the network. If an alternative route does not pass through a gate, the program will fail with an arithmetic error when it tries to compute the number of vehicles using that route (i.e., the sum of the gate volume to capacity ratios, GCRT, will equal zero).

Vehicles are applied to the links of each alternative route in inverse proportion to gate volume to capacity ratio (GCR), route travel time (C + CG), and the number of iterations specified for the route selection process (ITM). The procedure outlined above is followed until 1/ITM of all vehicles have been routed between all internal and external zones. If ITM is greater than 1, new travel times are determined on all links, a new set of alternative routes is generated, and vehicles are applied to these routes.

This subroutine is responsible for generating the minimum path between each origin and destination zone in the network. The method generates the minimum path from one origin AA to all links of the network. (Recall that a link is one direction of flow on a street and has a load point halfway between each end.) Two arrays are initialized before calling MINPTH. The $TT_{\ell,j}$ array represents the travel time from link ℓ to a connected link j. The C_ℓ array represents the cost of travel from the origin zone to any link ℓ of the network. The C array is initialized to zero for any link connected to the origin zone and to infinity for any unconnected link.

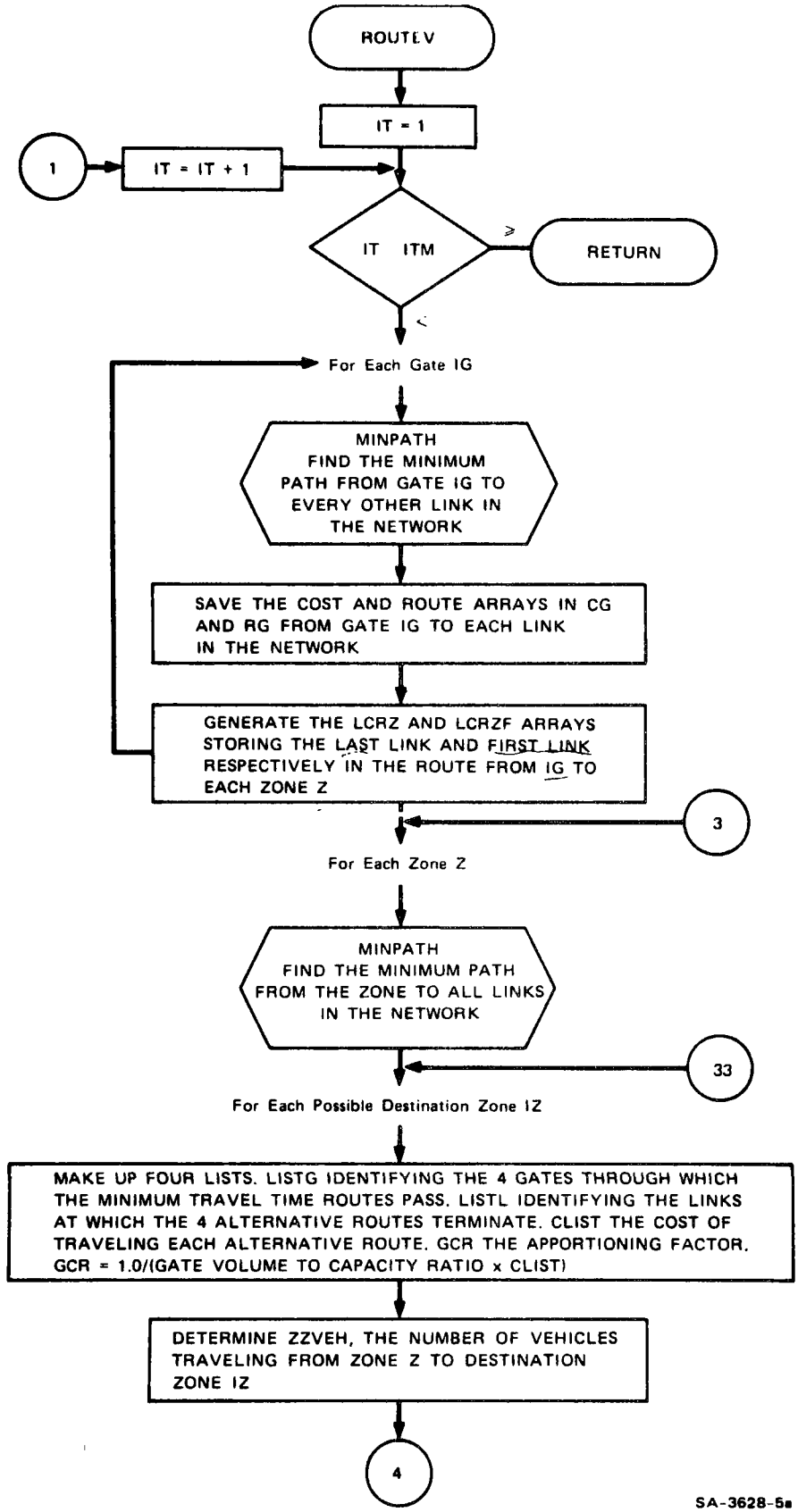

Figure 8. Route vehicles model flow diagram.

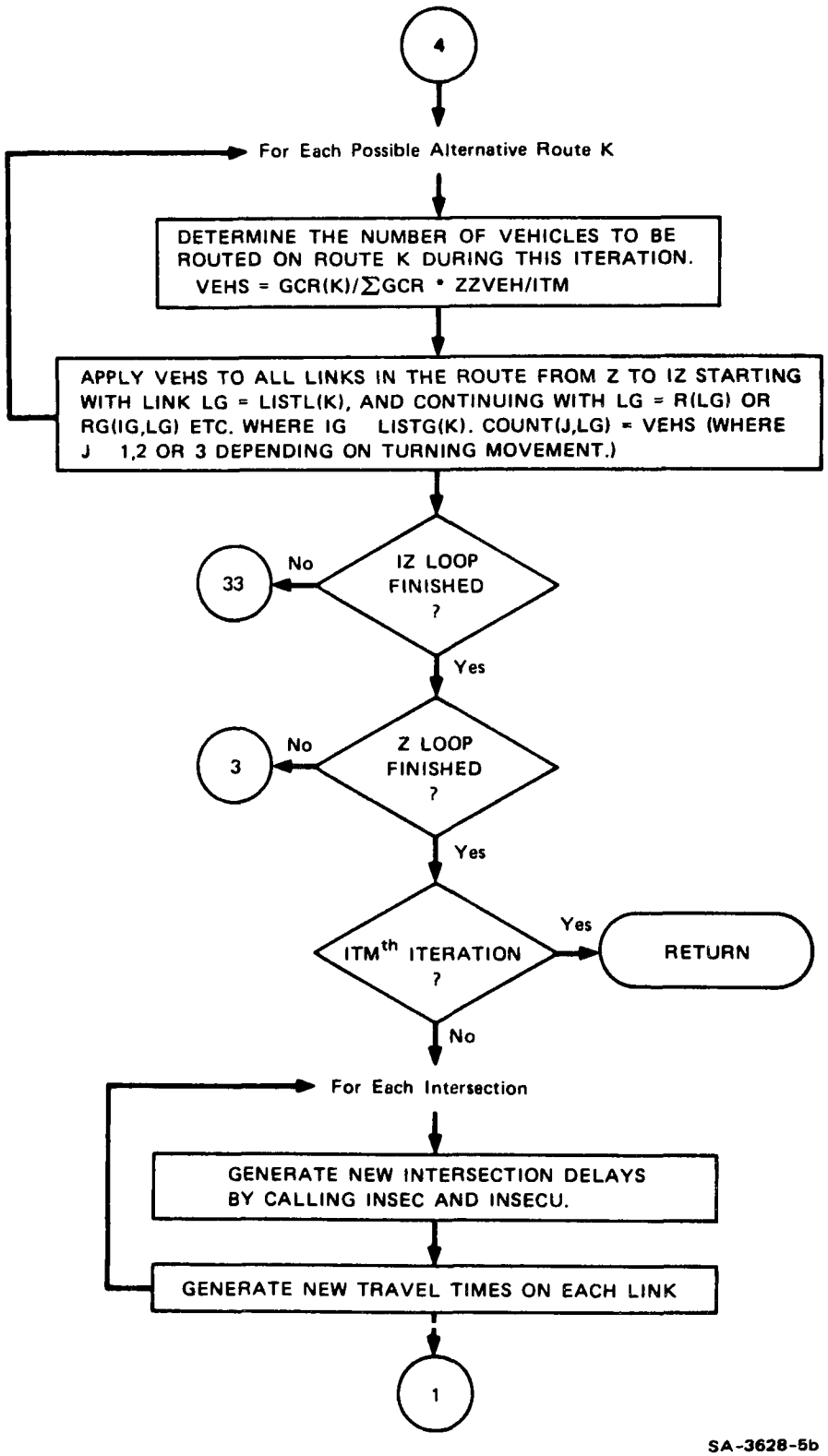

Figure 8 (continued). Route vehicles model flow diagram.

SUBROUTINE MINPTH

The algorithm described below is carried out using these two arrays. On completion, the C_ℓ array represents the minimum travel time (or cost) from the origin zone to each link ℓ of the network. During this process, the R_ℓ array is generated, which contains the link number that immediately precedes link ℓ in the route from the origin to ℓ. The following is a step by step description of the algorithm for route generation.

Step 1 — Initialize $TT_{\ell,j}$ with the constant cost in traveling from link ℓ to three possible adjacent links, $j = j_1, j_2, j_3$.

Step 2 — For an origin zone initialize with a large number the cost, C_ℓ, to travel from the origin to each link in the network ($C_\ell =$ for all ℓ).

Step 3 — For the links adjacent to the origin zone store a travel time cost in C_ℓ ($\ell = \ell_1, \ell_2, \ldots \ell_{12}$).

Step 4 — Set the travel time costs of Step 3 negative, to flag that all links connected to these may have an improved travel time cost.

Step 5 — Find a negative C_ℓ and the connected links j_1, j_2, j_3 to link ℓ. If no negative C_ℓ exists then do Step 9. If no connected links exist then do Step 8.

Step 6 — For each $j = j_1, j_2, j_3$ test $|C_\ell| + TT_{\ell,j} < |C_j|$. If the relationship does not hold true then do Step 7. Otherwise replace C_j by $|C_\ell| + TT_{\ell,j}$ and set C_j negative. The route is saved by storing the previous link to j (i.e., $R_j = \ell$).

Step 7 — Repeat from Step 6 until all j's have been tested.

Step 8 — Set C_ℓ to $|C_\ell|$ and repeat from Step 5.

Step 9 — The C_ℓ array is the cost of travel from the origin to each link ℓ. Repetition from Step 2 will generate new costs of travel from other origins.

SUBROUTINE INSEC

This subroutine cmmputes delays and queue lengths at signalized intersections within the simulated network (see Figure 9 for flow chart). Three types of signal controllers can be specified. The first type is a fixed-time controller in which the north-south, east-west, and left-turning phases are all of fixed time length. The other types of controllers are vehicle-actuated, for which types the INSEC routine must determine the length of each signal

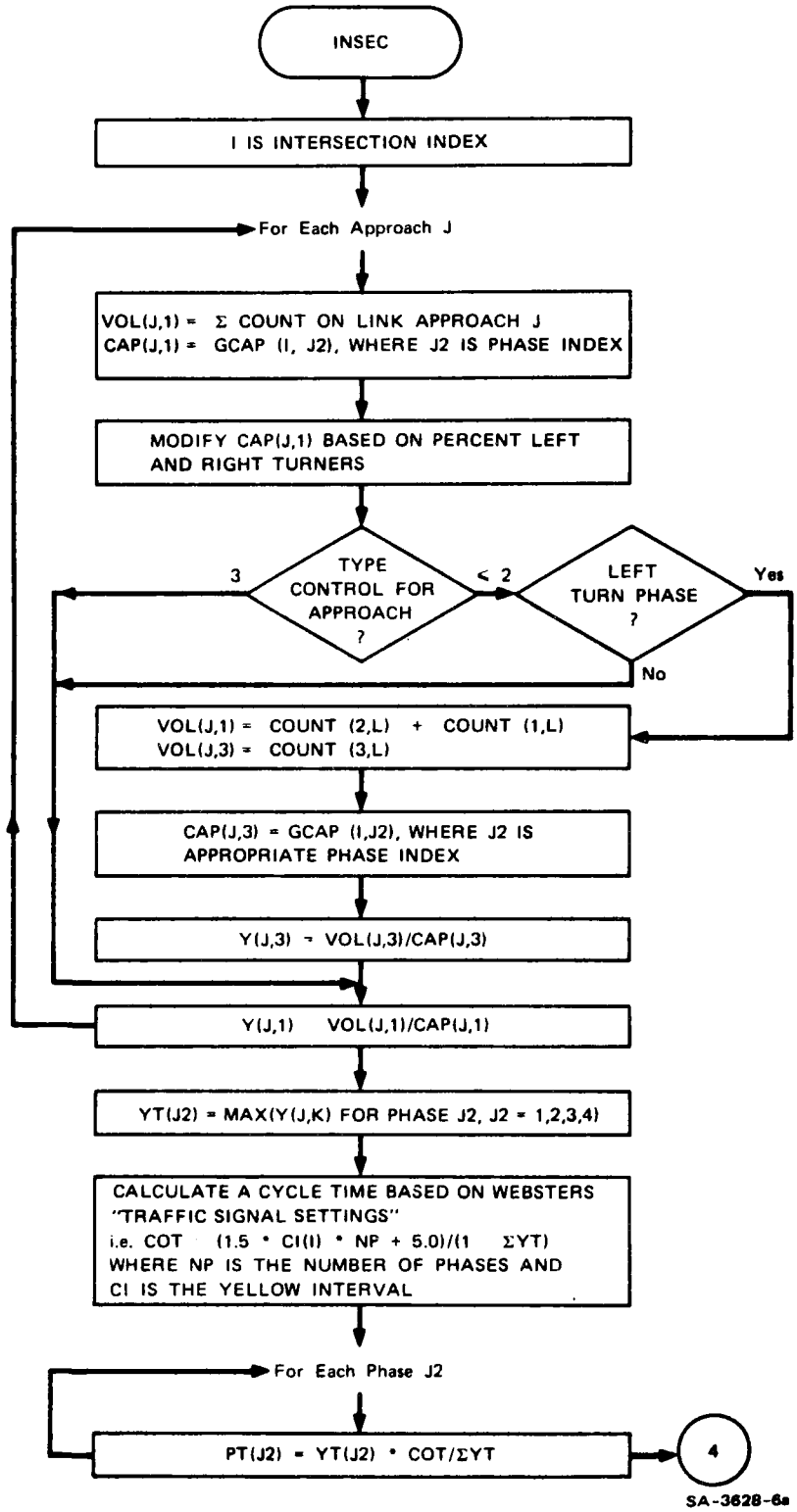

Figure 9. Signalized intersection model flow diagram.

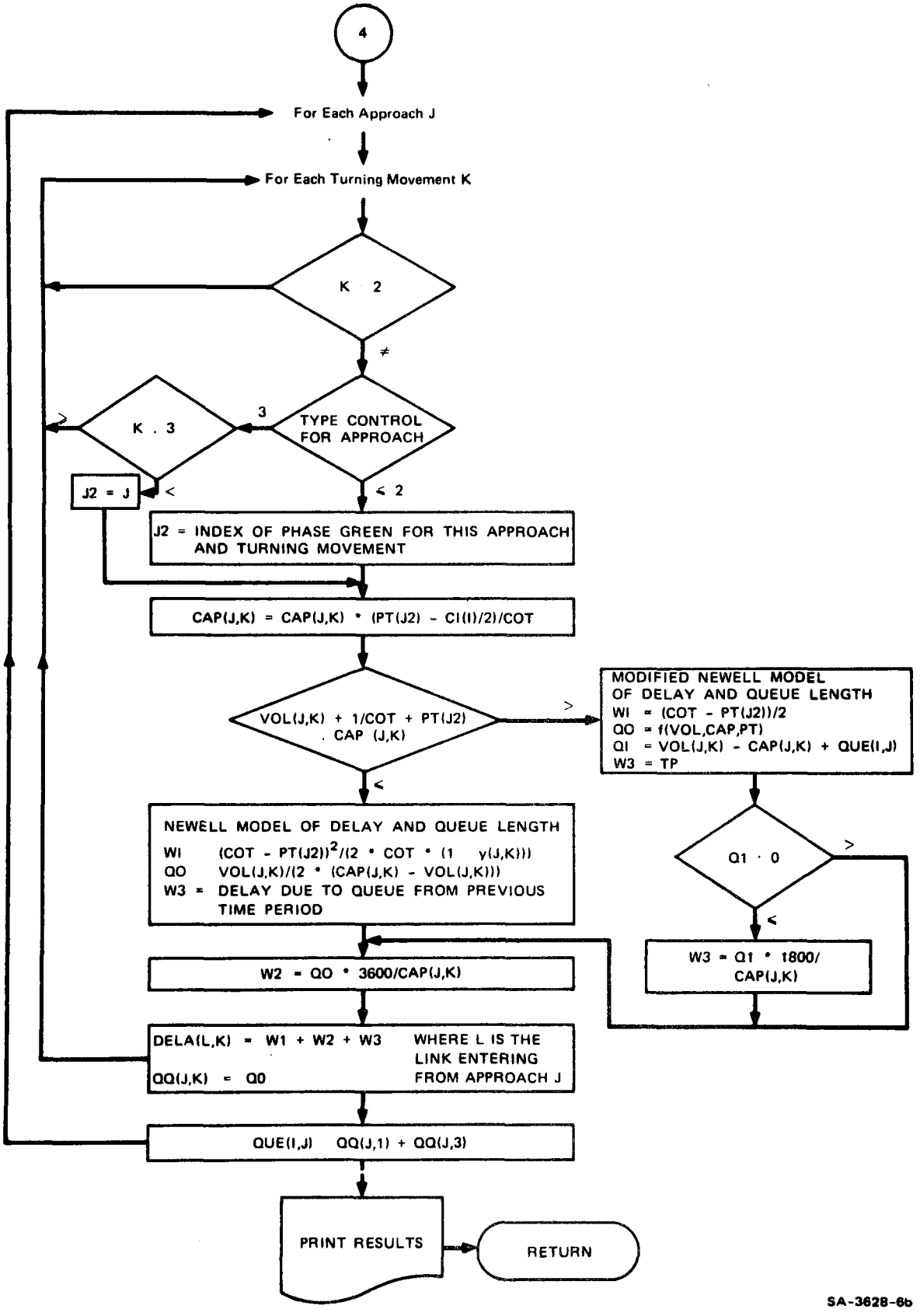

Figure 9 (continued). Signalized intersection model flow diagram.

phase. Phase lengths are determined based on volume to green capacities of each approach to an intersection.[3]

Delays and queue lengths are determined at an intersection on the basis of phase time, approach volume, and capacity. The referenced method of determining delay assumes a period of time during which traffic demand remains constant, and does not include a means of estimating queueing when volume exceeds capacity. For ISMAP a method was devised to provide continuous functions for delay and queue length when volumes approach or exceed the intersection approach capacity. This method establishes a constant rate of increase in queue length beyond the point where average delay at the intersection equals one cycle length. The queue is assumed to increase at a constant rate beyond this point until volume exceeds capacity, at which point the queue increases at the rate that volume exceeds capacity. A queue is dissipated at the rate that capacity exceeds volume until the average queue length, based on the volume to capacity ratio of the intersection approach, is reached.

The results of the model, stored in the DELA and QUE arrays, are delay and queue length for each approach to an intersection.

SUBROUTINE INSECU

This subroutine predicts the delay and queue length at unsignalized intersections in the network. Three types of intersections can be handled: uncontrolled, two-way stop, and four-way stop. If not specified as an input parameter, the INSECU routine determines an intersection capacity based on the Highway Capacity Manual.[9] For two-way stops and uncontrolled intersections, the HCM recommends that capacity be computed as if a signal were present and as if the signal split ratio equaled

$$\frac{\text{Volume}_1}{\text{Volume}_2} \times \frac{\text{Width}_2}{\text{Width}_1} \quad . \tag{19}$$

At four-way stops, capacity is a function of the number of lanes and the demand split among approaches, as shown in Tables 6.7 and 6.8 of the Highway Capacity Manual.[9] Queue length at an intersection is computed from classical queueing theory; i.e., QO = 1/(CAP/VOL) - 1. Average delay is a function of queue length times the time to process each vehicle.

SUBROUTINE PARKNG (EFFECTIVELY BYPASSED IN THE MODIFIED VERSION OF ISMAP)

This routine models parking zone flow and determines vehicle running time for a zone (see Figure 10 for flow chart). The running time of an average vehicle is considered to consist of three elements: (1) the time to travel from the edge of the zone to a stall or the time to travel from a stall to the edge of a zone (symbolized by TTA and TTD, respectively); (2) the delay experienced by arriving vehicles while waiting for vehicles to back out of stalls (DQ); (3) the average wait in a queue until a parked vehicle leaves the zone (AWQ).

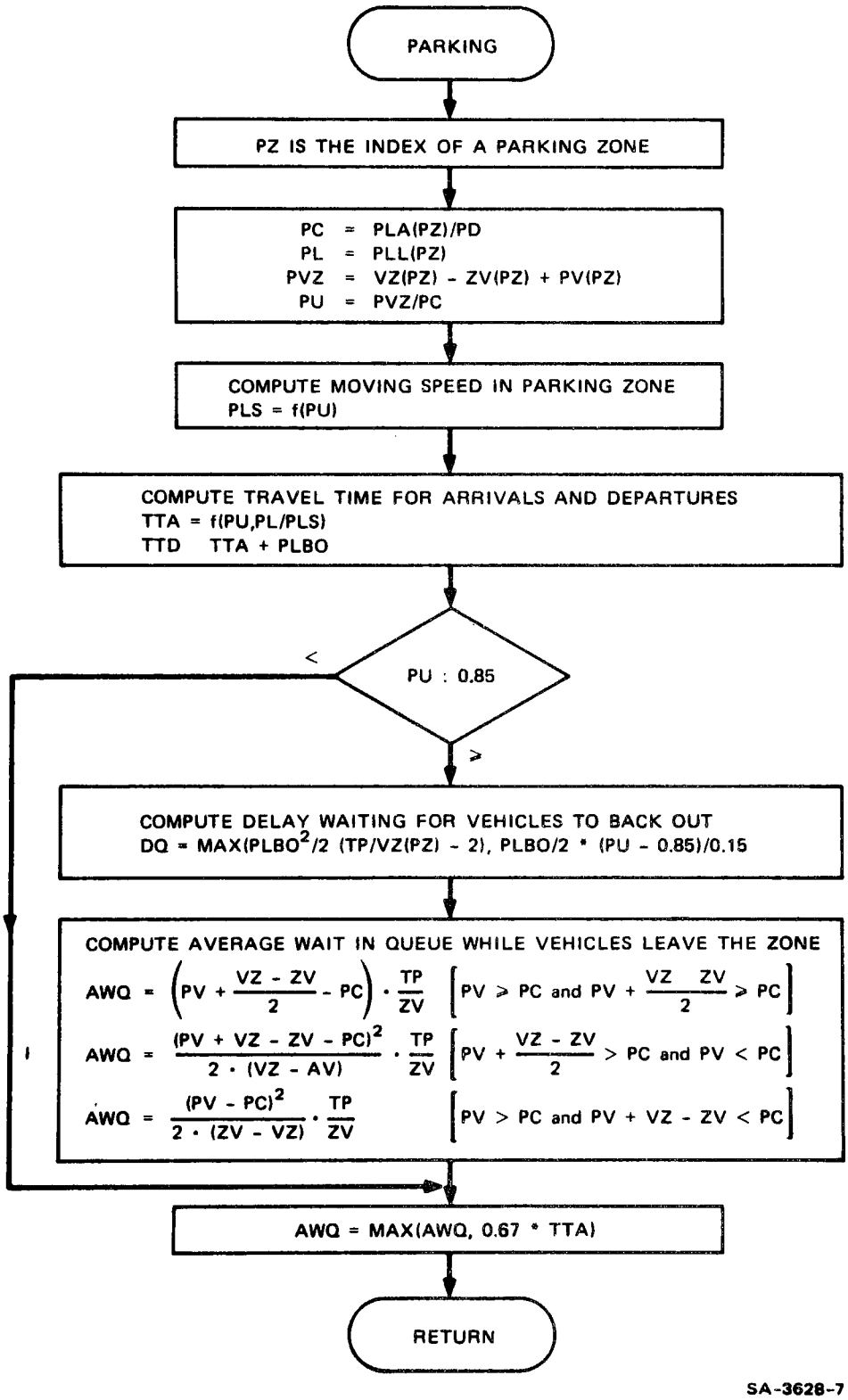

Figure 10. Parking model flow diagram.

TTA is computed on the basis of the length and speed of an average trip into the parking zone. It would appear that the length and speed of an average trip vary in direct and indirect proportion, respectively, to the utilization of a zone, and preliminary analysis of available data tends to support this supposition. TTD is the same as TTA, but with a time added to back out of a parking stall (PLBO).

DQ is the queued delay, due to interrupted flow lasting PLBO seconds, experienced by vehicles arriving in the parking zone. AWQ is a minimum value or is the time required to service each vehicle times an average queue length estimated from the utilization of the parking zone.

SUBROUTINE COORXY

To determine the location of steady speed, decelerating, queued, and accelerating vehicles on a link, it is necessary to know the direction of flow on the link (i.e., which end point traffic moves toward). This subroutine reorganizes the x,y coordinates of any link, L, so that X1(L), Y1(L) represent the upstream end of the link and X2(L), Y2(L) represent the downstream end of the link.

A second function of this routine is to convert the coordinates from feet to meters, since meters are the required units of distance in the dispersion submodel.

SUBROUTINE EMIT

Subroutine EMIT calculates the amount of CO produced on each traffic link (see Figure 4 for flow chart). The necessary input data come from the traffic analysis subroutine. These are: CAPL, COUNT, DIST, DQ, NLANE, NSTOPS, TT, and VEL. From this information, the model calculates the time and distance each vehicle devotes to constant velocity travel, deceleration, idling, and acceleration. The model then determines the amount of CO produced by one vehicle for 1 second for each cruise condition. Next, each of these emissions is assigned to that portion of the link where the emission occurred. The result is CO emissions as a function of position on each traffic link. The following are outputs of the EMIT subroutine for each traffic link: EMISS (I), I = 1,4 where I = 1 is the constant speed emissions, I = 2 is the decelerations emission, I = 3 is idling emissions, and I = 4 is the acceleration emissions, and DLINK (I), I = 1,4 is the distance over which each emission rate applies.

The emissions model also calculates parking zone emission rates based on total vehicle running time within the zone. Vehicles are assumed to emit CO within a zone at the idle emission rate, and the vehicle emissions are distributed evenly over an entire zone.

SUBROUTINE ACDC

The purpose of this subroutine is to calculate the total emission of CO by a vehicle accelerating from stop to a terminal velocity at a constant rate. Alternatively, the emissions generated by decelerating from a specific speed

to stop at a fixed rate can be determined. The integrated Model Emission equations were utilized to calculate the total CO emissions for the acceleration or deceleration mode.

SUBROUTINE CRUZ

This subroutine utilizes the Modal Emission Coefficients to calculate the CO emissions for vehicles traveling at a fixed rate of speed.

SUBROUTINE INITMM

This subroutine utilizes the composite vehicle age and mileage distribution calculated in SUP8 to calculate the vehicle distribution in the 20 modal model vehicle groupings. It also sets up the array of modal model coefficients.

SUBROUTINE SUP8

This subroutine calculates the average driving cycle emission factors and idle emission factors based on the parameters supplied to it. It, in turn, calls subroutines OUTPUT, EFCALX, BIGCFX, INITEX, TFCALX, SPFCLX, BEFGEN, GETCUM, EFALTX, CCEVAX, LDVIMX, ALUH, TRKOPC.

SUBROUTINE DISPER

This subroutine controls all dispersion calculations. It calls WIND, LSLOPE, STABLE, BASIC, GENGSC, and ZBORDR at the appropriate times (see Figure 11 for flow chart) and rounds off the coordinates of the link end points so that the center of a grid square cannot coincide with a link end point. As the final step before printing, DISPER divides the normalized CO concentrations by pi and the wind speed, converts the concentration to ppm, adds the background concentration, and multiplies by a factor that corrects the emission rate for the vehicle mix of 1971 to that of the input year and corrects for the ambient temperature and the fraction of hot and cold started vehicles.

SUBROUTINE WIND

Meteorological wind direction, measured clockwise from north, is converted to algebraic wind direction, measured counterclockwise from east. In addition, the angle from the y axis to the wind vector is found by WIND.

SUBROUTINE LSLOPE

The slope and y intercept of each link are found by this subroutine. If the slope of a link is infinite, both the slope and the y intercept are set equal to zero as a cue to indicate that the link is vertical.

SUBROUTINE STABLE

STABLE determines the stability class and therefore the dispersion parameters for each hour of the day, unless stability is input directly. The

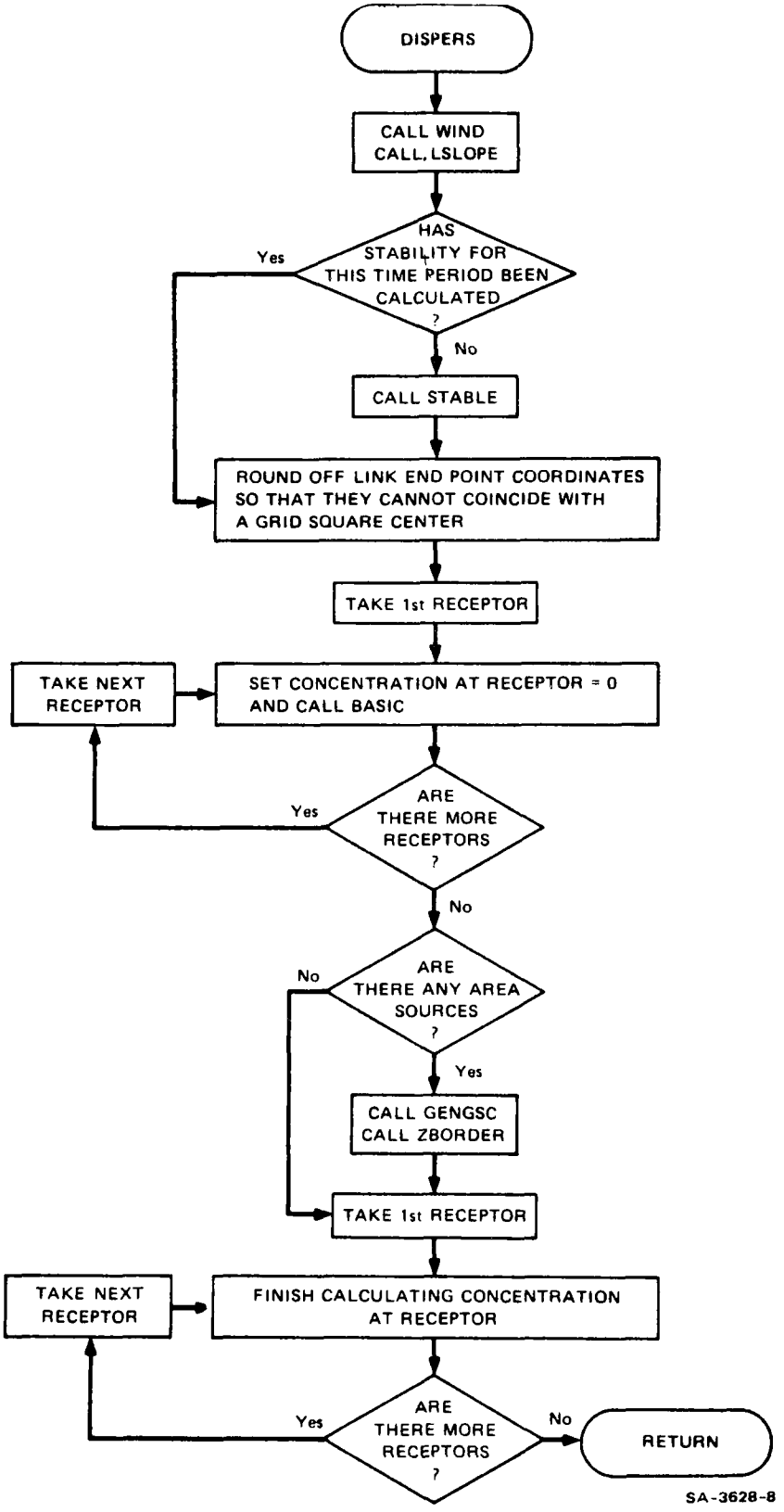

Figure 11. Subroutine DISPERS flow diagram.

amount of cloud cover is used in determining an insolation class, which, when coupled with a wind speed class, allows an appropriate stability to be chosen from a table. If the user chooses to predetermine and read in stability as input, this initial part of the subroutine is bypassed, and the time period need not be 1 hour. Wind speed is converted from knots to meters per second for use in the point-source equation, and the diffusion parameters appropriate to the stability class are located from tables and stored.

SUBROUTINE BASIC

This routine determines whether all or any part of a link is upwind of a receptor. If a link lies within a 90°, pie-shaped segment that originates at the receptor and is bisected by the wind vector, the link is considered to be upwind of the receptor. The subroutine checks this by first transforming the coordinates of the link end points into a system with the origin at the receptor and the positive y axis in the direction from which the wind is blowing. If the new y coordinates of both end points of the link are negative, the link is downwind of the receptor. If the y coordinate of one or both end points of a link is positive and the angle between the new x axis and a line connecting a positive y coordinate of a link point with the origin is greater than or equal to 45°, at least part of the link is upwind of the receptor. It is also possible that the link passes through the segment even though neither of its end points lies in the pie-shaped segment. For this to occur the transformed x-coordinate of one end point must be positive and the other negative. In this case, the point at which the link intersects the y axis is found. If the y coordinate of that point is positive, the link passes through the segment and part of it is upwind of the receptor. If all or part of a link lies upwind of a receptor, BASIC calls ORIGIN, CORTRN, and CONCA1 or CONCA2.

In this routine the calculation was also performed to determine whether a street canyon vortex will form at each street canyon receptor location based upon hourly wind speed and stability conditions.

SUBROUTINE ORIGIN

Here the coordinates of the point to be the origin in a new coordinate system are found. For each link the origin is taken as the point of intersection of the link (or its extension) and the wind vector. ORIGIN also finds the angle from the wind vector to the link.

SUBROUTINE CORTRN

This subroutine transforms the coordinates of the receptor and the link end points into a new system with the y axis along the direction of the link and the origin as found by ORIGIN. This transformation is done so that the same Gaussian point source equation can apply to all receptor-link configurations.

SUBROUTINE CONCA1

CONCA1 is used when the wind is not parallel to the link. It calls MISC and XSIGN and computes the normalized CO concentration at a receptor resulting from each point on the link. Subroutine BASIC determined that at least part of the link is upwind of the receptor. Now each point on the link is checked to see that it is upwind of the receptor by comparing the sign of its distance along the wind direction to the receptor with the sign found by XSIGN. If the signs are the same, ISMAP determines what driving mode the traffic is in at the point on the link, chooses the corresponding emission rate, computes the lateral and vertical standard deviations of plume concentration, and applies the point source equation. Then for each receptor all point source concentrations are summed to give the CO concentration at the receptor from a link, and this concentration is added to the concentrations from other links for that receptor.

SUBROUTINE XSIGN

Some wind vector-link-receptor configurations are such that only part of a link is upwind of a receptor. In this case, there is a point on the link that has zero distance to the receptor along the wind direction. On one side of this point, the distance from link to receptor along the direction of the wind is positive; on the other side it is negative. Due to the coordinate transformation performed in CORTRN, it is not known whether the points that are actually upwind will show positive or negative distances. Therefore, the sign of the distance along the wind direction from the receptor to a point known to be upwind of the receptor is found for use in determining which points on the link will contribute to the concentration.

The point used in the above determination depends on the orientation of the link and of the wind vector. When the wind vector and the link are not parallel, the coordinate system origin found by subroutine ORIGIN is used, unless it coincides with the receptor location. If it does coincide, the known point becomes the point of intersection of the wind vector passing through the receptor and the perpendicular to the wind that passes through a link end point. When the wind vector and the link are parallel and the receptor is located on an extension to the link, one of the link end points becomes the known point. Or, if the receptor is not located on an extension to the link, the known point used is the point of intersection of the wind vector passing through the receptor and the perpendicular to the link (or wind vector) that passes through a link end point.

When the quadrant from which the wind vector originates and the signs of the differences between the x coordinates and between the y coordinates in the original coordinate system of the receptor and the point chosen for the sign check are known, the sign of the distances from the receptor to those points that are upwind of the receptor can be determined. This sign is used in CONCA1 or CONCA2.

SUBROUTINE SPCING

This subroutine computes the distance to be taken between the point sources that represent the link, using the method presented in Section 2. Equation 12 is applied to determine the spacing. To compute the σ_y that appears in the equation, a downwind distance from point source to receptor is needed. If the wind vector that passes through the receptor (which is assumed to be perpendicular to the link) intersects the link, the downwind distance used in σ_y is simply the absolute value of the x coordinate of the receptor in the coordinate system generated by ORIGIN and CORTRN. If the wind vector intersects an extension to the link, the distance from the receptor to the closest link end point is used for computation of σ_y. Then Equation 12 is solved for the spacing, letting the initial horizontal diffusion in the equation equal the spacing. However, if the value found for the spacing is less than 3 m, the assumed initial horizontal diffusion was less than 3 m. Since this assumption is probably not realistic, Equation 12 is solved again for the spacing, using 3 m as initial horizontal diffusion. To be conservative, the model uses half of the spacing value solved for as the distance between point sources. The spacing solved for above is taken as the value of initial horizontal diffusion on the link unless it is greater than the half-width of the link or less than 3 m. When the spacing value falls outside these limits, initial horizontal diffusion is set equal to the limiting value.

SUBROUTINE MISC

Several miscellaneous calculations are performed in this subroutine. MISC calls subroutine SPCING. It calculates the length of the link and the number of point sources that will represent the link. The subroutine finds the coordinates of the points on the link that separate one mode of driving operation from another. CONCA1 or CONCA2 uses these points to determine which driving mode is being practiced at the location of each point source on the link. MISC also initializes for each link the parameter on which concentrations are stored and the parameter that holds the y coordinate of the points that represent the link.

SUBROUTINE CONCA2

Subroutine CONCA2 is used when the wind is parallel to the link. It performs the same functions as CONCA1, except that the point source equation is formulated for the case where the link and wind vector are parallel.

SUBROUTINE GENGSC

The emissions from each area source are assumed to be uniform over the area source. Therefore, ISMAP overlays a grid on the entire region being modeled and allocates the emissions of an area source equally to those grid squares that cover the source. The emissions from a grid square are assumed to be released from the center of the square. Subroutine GENGSC finds the maximum of both the x and y coordinates of all link end points to determine the area over which the grid will be overlayed. The number of grid squares in both the x and y directions are determined using the maximum found above

and the grid size that was input. Then, to conserve space in core memory, the coordinates of the centers of the grid squares are generated separately, to be paired later in ISMAP. The final step in the subroutine is to set the initial diffusion for area sources equal to the grid size.

SUBROUTINE ZBORDR

A zone (i.e., area source) may have up to six border links that form its sides. A zone may also have one nonlink side, but the border link information must be input such that the first and last border links are the links that do not connect. For each zone subroutine ZBORDR checks to see that one of the end points of a border link connects with an end point of the next link. If the specified border links do not connect, an error message is printed and computations for the zone are bypassed. If the links do connect, subroutine INTERS is called. This checking procedure and calling of INTERS continues until all the border links have been exhausted. If the open end points of the first and last links connect, the program calls subroutine ZNGSC; if the open ends do not connect, the last side of the zone, along with its slope and y intercept, are generated and ZNGSC is called. The entire procedure outlined above is repeated for each zone.

Other subroutines, called by ZNGSC, compute the contribution to the CO concentration at a receptor from each zone. After the routines have been called, ZBORDR multiplies the concentration contribution of each zone for each receptor by the zone emission rate and divides by the number of grid squares in the zone. These concentrations then are summed for each receptor. However, the results are still normalized by pi and the wind speed. In the modified version of ISMAP, the interior zones will be bordered by only two links.

SUBROUTINE INTERS

Since the overlay grid is made up of regularly spaced squares, the centers of the squares can be said to lie in vertical lines. There are as many of these lines as there are grid squares in the x direction. This subroutine finds points of intersection of each vertical line with the sides of a zone. Since the manner in which the grid square centers were generated and the manner by which the link end points were rounded off preclude the possibility of a vertical line passing through a zone border link end point, there will be an even number of points if intersection on each vertical line. The points of intersection are found for one zone at a time, and the number of points of intersection on each vertical line, as well as the y coordinate of each point of intersection, are stored. The x coordinates can be found from the index of the vertical line since the values are the same as those of the x coordinates of the grid square centers with the same index.

SUBROUTINE ZNGSC

The y coordinates of the points of intersection of a vertical line with the zone sides, found by INTERS, will not be necessarily in ascending or

descending numerical order. Subroutine ZNGSC places the intersection points on each vertical line in descending numerical order so that they may be paired. If an uneven number of points of intersection are found, an error message is printed and further computations for the zone are bypassed. Otherwise, each grid square center that lies on a vertical line is checked to see if it lies between any of the pairs of intersection points. If it does, the grid square center, and therefore the grid square, lies within the zone. When a grid square center is found to be located in a zone, it is counted, subroutine ZCHECK is called, and the x and y coordinates of the grid square center are passed to ZCHECK.

SUBROUTINE ZCHECK

This subroutine transforms the coordinates of a zone grid square center into a system with the origin at a receptor and the y axis along the direction of the wind. If the transformed y coordinate of the grid square center is negative, the grid center is downwind of the receptor. For a positive y coordinate, if the angle that a line passing through the origin and the grid center makes with the x axis is less than 45°, the grid center is assumed to be positioned such that its contribution to the concentration at the receptor is negligible. If the angle is greater than or equal to 45°, subroutine CONCA3 is called. The above procedure is repeated for all receptors.

SUBROUTINE CONCA3

The transformed y coordinate of the grid square center that was passed along from ZCHECK is the distance along the direction of the wind from the point source to the receptor. This distance and the diffusion parameters found in STABLE are used to calculate the lateral and vertical standard deviations of plume concentration. The x coordinate of the grid square center gives the crosswind distance from source to receptor. Using these values, CONCA3 calculates a normalized concentration at a receptor resulting from a grid square of a zone. This concentration is added to the contributions to the concentration at the receptor from other grid squares in the zone. The results of CONCA3 are passed back to ZBORDR.

SUBROUTINE STREET

If the calculation in subroutine BASIC indicates, based upon hourly wind speeds and atmospheric stabilities, that a street canyon vortex will form at a particular street canyon receptor location, then subroutine STREET will be called to calculate the hourly concentration contribution from the street canyon links, which will then be added to the remaining link contributions calculated by the line source dispersion model. The calculated street canyon concentration will depend upon whether the receptor is located on the windward or leeward side of the street.

SECTION 4

INPUT DATA SPECIFICATION

The first step in the preparation of input data for the modified version of ISMAP is to draw a map of the traffic network under study. The link endpoints and receptor locations must be specified by coordinates in a west to east (x) and south to north (y) coordinate system. Each link must be characterized by the number of lanes and the identification numbers for those links branching from (turning left, turning right and going straight) the link in question. For each intersection the signal type (if signalized) and phase times must be specified. Receptor locations that are potential street canyons must be so identified. For such receptors, the user must also specify the direction of the street with respect to north, the identification numbers of the links comprising the street, the width of the street (building to building) and the height of the buildings on the street.

The most critical inputs to the modified ISMAP are the assignment of the internal and external zones in terms of the links entering and leaving the zone and the number of vehicles attracted to or generated by the internal zones during a given hour. Vehicle trips from an external zone are directed toward internal zones based upon the trip attraction of the internal zones. In the same manner, vehicle trips generated by internal zones are apportioned to external zones, but trips cannot be generated between individual internal or external zones. Since parking lots are not considered in the modified version of ISMAP, the internal and external zones are physically indistinguishable, with each consisting of an entrance and exit link pair. As illustrated by our sample model application presented in Section 5, the names "internal" or "external" zones do not indicate the geographic location of these zones but only signify that vehicles are routed between these two types of zones.

The final input requirement for the model is the hourly meteorological data for use in the dispersion calculation. Hourly temperature is also input for calculation of cold and hot start vehicle emission correction factors.

In Table 2 the program input requirements are listed on a card by card basis along with column numbers, format specification, units, value limits and general description. The remainder of this section will be devoted to a detailed description of each input card.

TABLE 2. MODIFIED ISMAP BASIC INPUT INFORMATION CARDS 0 THROUGH 10 (Data Read in by Subroutine INPT)

Card	Column	Format	Symbol	Units	Value limits	Typical value (default value in parentheses)	Description
0 Header	1-6	I6	NYEAR	–	710101-to 991231	741004	Date in year, month, day sequence
	7-76	18A4	LHEAD	–	–	EXAMPLE CASE	Heading to precede output summary
	80	I1	NOPARK	–	1	1	Must be set equal to 1
1	1-2	I2	IC	–	–	–	Card type number
	3-5	I3	L	–	0* to 200	154 (previously stored value)	Number of cards of type 2
	6-10	I5	I	–	0* to 70	48 (previously stored value)	Number of cards of type 3
	11-15	I5	Z	–	0* to 50	23 (previously stored value)	Number of cards of type 4
	16-20	I5	ICN	–	0 or 1	1 (previously stored value)	Number of cards of type 5
	21-25	I5	IG	–	0* to 10	7 (previously stored value)	Number of cards of type 6
	26-30	I5	NV	–	0* to 3	3 (previously stored value)	Number of cards of type 7
	31-35	I5	KT	–	0 to 200	15 (previously stored value)	Number of cards of type 8
	36-40	I5	NR	–	0* to 200	26 (previously stored value)	Number of cards of type 9
	41-45	I5	NMET	–	0* to 24	1 (previously stored value)	Number of cards of type 10
	46-50	F5.0	TOD	hours	0 to 23	8	Hour of day
	51-55	F5.0	DOW	–	1 to 7	3	Day of week (Sun. = 1, Mon. = 2 . . . Sat. = 7)
	56-60	F5.0	TP	sec	900 to 7200	3600	Time period duration
	61-65	F5.0	TOTATT	vehicles	≥ 0	124	Total trips attracted to and generated by the indirect sources during time period specified
	66-70	F5.0	TOTGEN	vehicles	≥ 0	74	
	71-75	F5.0	PD	ft^2/veh	360	360	Parking density constant (number of ft^2 required to park each vehicle)
	76-80	I5	ITM	–	1 to 10	1 (1)	Number of iterations of the route selection process

*May not be 0 on first iteration of program.

(continued)

TABLE 2 (continued).

Card	Column	Format	Symbol	Units	Value limits	Typical value (default value in parentheses)	Description
2	1-2	I2	IO	-	2	2	Card type number
	3-5	I3	L	-	0 to 200	1	Link number
	6-10	I5	NLANE(L)	-	≥ 1	2	Number of lanes
	11-15	F5.0	X1(L)	ft	> 0	3751	x-coordinate of northernmost end point (easternmost coordinate if link runs E-W)
	16-20	F5.0	Y1(L)	ft	> 0	2363	y-coordinate of northernmost end point
	21-25	F5.0	X2(L)	ft	> 0	2247	x-coordinate of southernmost end point
	26-30	F5.0	Y2(L)	ft	> 0	2673	y-coordinate of southernmost end point (links should terminate at the intersection center)
	31-35	I5	LCAP(L)	veh/hr	≥ 1	-0 (1800)	Capacity of link, veh/hr
	36-40	F5.0	VEL(L)	ft/sec	1 to 147	58	Speed, ft/sec
	41-45	I5	LCON(L,1)	-	0 to 200	3	Link to link connections. The link that L connects to going straight
	46-50	I5	LCON(L,2)	-	0 to 200	26	The link L connects to going right
	51-55	I5	LCON(L,3)	-	0 to 200	27	The link that L connects to going left
	56-60	F5.0	HEIGHT(L)	m	≥ 0	0	Emission height

(continued)

TABLE 2 (continued).

Card	Column	Format	Symbol	Units	Value limits	Typical value (default value in parentheses)	Description
3	1-2	I2	IC	–	3	3	Card type number
	2-5	I3	I	–	0 to 70	1	Intersection number
	6-10	I5	LIN(I,J)	–	0 to 200	142	North link number
	11-15	I5	LIN(I,J)	–	0 to 200	4	East link number
	16-20	I5	LIN(I,J)	–	0 to 200	143	South link number
	21-25	I5	LIN(I,J)	–	0 to 200	1	West link number
	26-30	I5	ITYPC(I)	–	-2 to 5	2	Type control; -1 = 2 way stop; -2 = 4 way stop; 0 = no control; 2,3 = V/A control; 1 = fixed time control; 4 = type 2 N-S, type 3 E-W; 5 = type 3 N-S, type 2 E-W
							Type I ft phases / Type II actuated phases / Type III actuated phases
	31-35	I5	CYCL(I)	sec	≥ Σ PH(I,J)	200	Cycle / Max cycle / Max cycle
	36-39	F4.0	PH(I,J)	sec	≥ 0	10	N-S / Min N-S / Min N
	40-43	F4.0	PH(I,J)	sec	≥ 0	10	E-W / Min E-W / Min E
	44-47	F4.0	PH(I,J)	sec	≥ 0	0	N-S LT / Min N-S LT / Min S
	48-51	F4.0	PH(I,J)	sec	≥ 0	0	E-W LT / Min E-W LT / Min W
	52-55	F4.0	CI(I)	sec	≥ 0	3	Yellow time interval
	56-59	F4.0	GCAP(I,1)	veh/hr	≥ 0	3600 (1200 veh/hr/lane)	Capacity of one approach N (or S) for phase 1 per hr green
	60-63	F4.0	GCAP(I,2)	veh/hr	≥ 0	4000 (1200 veh/hr/lane)	Capacity of E (or W) phase 2 per hr green
	64-67	F4.0	GCAP(I,3)	veh/hr	≥ 0	0 (1200 veh/hr/lane)	Capacity of S or S-N phase 3 per hr green
	68-71	F4.0	GCAP(I,4)	veh/hr	≥ 0	0 (1200 veh/hr/lane)	Capacity of W or W-E phase 4 per hr green

(continued)

TABLE 2 (continued).

Card	Column	Format	Symbol	Units	Value limits	Typical value (default value in parentheses)	Description
4	1-2	I2	IC	–	4	4	Card type number
	3-4	A2	IEXT	–	01 - 99	5	Zone identification (interior zones: 01-50; exterior zones: 51-99)
	5-7	F3.0	ZATTR(Z)	veh/hr or %	≥ 0	2	Zone attractions in either (1) average volume (veh/hr) or (2) % of total trips attracted to all interior or all exterior zones
	8-10	F3.0	ZGENR(Z)	veh/hr or %	≥ 0	2	Zone generations in either (1) average volume (veh/hr) or (2) % of total trips generated to all the interior or to the exterior zones
	11-13	F3.1	ZVT(1,Z)	–	≥ 0.	0. (1.)	Fraction of zone attractions and generations that are automobiles
	14-16	F3.1	ZVT(2,Z)	–	≥ 0.	0.	Fraction of zone attractions and generations that are buses
	17-19	F3.1	ZVT(3,Z)	–	≥ 0.	0.	Fraction of zone attractions and generations that are other vehicle types
	20-24	F5.0	PVEH	veh	≥ 0	0	Initial number of parked vehicles in zone
	25-29	F5.0	PLC	veh or gm/sec	≥ 0	0	Parking lot capacity in vehicles (if left blank lot capacity is generated by the computer) or zone emissions for exterior zones in gm/s during time period
	30-31	I2	NZL	–	1 to 6	1	Number of sides (links) which define parking area (the first NZL bordering links define the parking lot area)
	32-35	I4	ZLINKS(1,Z)	–	0 to 200	112	Bordering link number 1 (a negative link number indicates no access to the zone from this link)
	36-39	I4	ZLINKS(2,Z)	–	0 to 200	111	Bordering link number 2
	.	.					.
	.	.					.
	.	.					.
	76-79	I4	ZLINKS(12,Z)	–	0 to 200	0	Bordering link number 12

(continued)

TABLE 2 (continued).

Card	Column	Format	Symbol	Units	Value limits	Typical value (default value in parentheses)	Description
5	1-2	I2	IC	-	5	5	Card type number
	3-6	F4.0	GRDSIZE	m	≥ 0.	10. (10.)	Grid size
	7-10	F4.0	BACKGRD	ppm	≥ 0.	0.	Background concentration at site
	11-14	F4.0	SLAT	deg of lat.	0. to 90.	0.	Latitude of site (needed only if stability is to be computed by ISMAP)
	15-18	F4.0	TUNFAC	-	0.1 to 3.	1.	Tuning factor
	19-22	F4.0	ZCONST	m	3. to 10.	10. (3.)	Initial vertical diffusion
	23-24	I2	IREG	-	1,2,3	1	Region: 1 = low 2 = calif., 3 = high
	25-28	F4.0	TEMP	°F	0 - 110	75.	Temperature
	29-33	F5.3	COLD		0 - 1	0.20	Proportion of cold starts
	34-38	F5.3	HOT		0 - 1	0.20	Proportion of hot starts
	39-43	F5.3	CCOLD		0 - 1	0.20	Proportion of catalyst cold starts
	44-48	F5.3	VMTMIX(1)	-	0. to 1.	0.75	Fraction of light-duty vehicles
	49-53	F5.3	VMTMIX(2)	-	0. to 1.	0.05	Fraction of light-duty trucks, less than 600 GVW
	54-58	F5.3	VMTMIX(3)	-	0. to 1.	0.05	Fraction of light-duty trucks, 6000 - 8500 GVW
	59-63	F5.3	VMTMIX(4)	-	0. to 1.	0.09	Fraction of heavy-duty gasoline-fueled vehicles
	64-68	F5.3	VMTMIX(5)	-	0. to 1.	0.06	Fraction of heavy-duty diesel-fueled vehicles
	69-73	F5.3	VMTMIX(6)	-	0. to 1.	0.005	Fraction of motorcycles
	74-75	I2	ALHFLG	-	0 or 1	0	Flag for optional air conditioning, loading humidity correction
	76	I1	TRKFLG	-	0 or 1	0	Flag for optional truck correction
	77	I1	IMFLG	-	0 or 1	0	Flag for optional I/M correction

(continued)

TABLE 2 (continued).

Card	Column	Format	Symbol	Units	Value limits	Typical value (default value in parentheses)	Description
					Omit if ALHFLG equals zero		
5A	1-10	-	-	-		-	Blank
	11-20	F10.0	AC	-	-	.30	Fraction of LDV and LDT VMT with air-conditioning
	21-30	F10.0	XLOAD(1)	-	-	.10	Fraction of LDV VMT with additional 500 lb load
	31-40	F10.0	XLOAD(2)	-	-	.10	Fraction of LDT1 VMT with additional 500 lb load
	41-50	F10.0	XLOAD(3)	-	-	.10	Fraction of LDT2 VMT with additional 500 lb load
	51-60	F10.0	TRAILR	-	-	.05	Fraction of LDV VMT with a trailer (1,000 pounds)
	61-70	F10.0	ABSHUM	gr/lb	-	75	Humidity (not currently applicable)
					Omit if TRKFLG equals zero		
5B	1-10	-	-	-	-	-	Blank
	11-20	F10.0	HGWGT	lbs		18,000	Average gas vehicle weight
	21-30	F10.0	HDWGT	lbs	19,000-55,000	45,000	Average diesel vehicle weight
	31-40	F10.0	HGCID	in^3	330-390	350	Average gas displacement
	41-50	F10.0	HDCID	in^3	540-600	580	Average diesel displacement
					Omit if IMFLG equals zero		
5C	1-10	-	-	-	-	-	Blank
	11-15	I5	ICYIM	-	-	82	Last two digits of implementation year of I/M
	16-20	I5	ISTRIN	-	-	30	Stringency
	21-25	I5	IMTFLG	-	0-1	0	Mechanic's training (1 = yes)
	26-30	I5	MODYR1	-	-	71	Begining and ending model years subject to inspection
	31-35	I5	MODYR2	-	-	81	
6	1-2	I2	IC	-	6	6	Card type number
	6-10	I5	LGATE(1,IG)	-	1 to 200	15	Link number of exit gate to parking lot
	11-15	I5	LGATE(2,IG)	-	1 to 200	16	Link number of approach to parking lot

(continued)

TABLE 2 (continued).

Card	Column	Format	Symbol	Units	Value limits	Typical value (default value in parentheses)	Description
7	1-2	I2	IC	–	7	7	Card type number
	3-5	I3	VT	–	1, 2, or 3	1	Vehicle type, auto (park) = 1; bus (stop and load) = 2; truck (stop and wait) = 3
	6-10	F5.3	VTP(VT)	–	$\geq$ 1.	1.4	Number of passengers in average vehicle
	11-15	F5.3	VTE(VT)	–			

(continued)

TABLE 2 (continued).

Card	Column	Format	Symbol	Units	Value limits	Typical value (default value in parentheses)	Description
8	1-2	I2	IC	–	8	8	Card type number
	3-5	I3	L	–	1 to 200	151	Link number on which count occurs ($1 \leq L \leq$ NLINK)
	6-10	F5.0	COUNT(1,L)	veh	$\geq 0.$	755.	Count going straight from link L through the intersection during time period
	11-15	F5.0	COUNT(2,L)	veh	$\geq 0.$	88.	Count going right from link L
	16-20	F5.0	COUNT(3,L)	veh	$\geq 0.$	107.	Count going left from link L
	21-25	I5	L2	–	1 to 200	152	Link number on which count occurs
	26-30	F5.0	COUNT(1,L2)	veh	$\geq 0.$	1064.	Count going straight from link L2
	31-35	F5.0	COUNT(2,L2)	veh	$\geq 0.$	0.	Count going right from link L2
	36-40	F5.0	COUNT(3,L2)	veh	$\geq 0.$	0.	Count going left from link L2
	41-45	I5	L3	–	1 to 200	153	Link number on which count occurs
	46-50	F5.0	COUNT(1,L3)	veh	$\geq 0.$	457.	Count going straight from link L3
	51-55	F5.0	COUNT(2,L3)	veh	$\geq 0.$	147.	Count going right from link L3
	56-60	F5.0	COUNT(3,L3)	veh	$\geq 0.$	170.	Count going left from link L3
	61-65	I5	L4	–	1 to 200	154	Link number on which count occurs
	66-70	F5.0	COUNT(1,L4)	veh	$\geq 0.$	429.	Count going straight from link L4
	71-75	F5.0	COUNT(2,L4)	veh	$\geq 0.$	0.	Count going right from link L4
	76-80	F5.0	COUNT(3,L4)	veh	$\geq 0.$	0.	Count going left from link L4

(continued)

TABLE 2 (continued).

Card	Column	Format	Symbol	Units	Value limits	Typical value (default value in parentheses)	Description
9	1-2	I2	IC	–	9	9	Card type number
	3-12	F10.5	XO(I)	ft	> 0.	2983.75	X-coordinate of receptor location
	13-22	F10.5	YO(I)	ft	> 0.	3774.25	Y-coordinate of receptor location
	23-25	I3	ISTR(I)	–	0, 1	0	0 = exclude street canyon model 1 = potential street canyon receptor
	26-30	F5.0	AST(I)	deg from N	0. to 180.	17.0	Direction of the street in degrees from north
	31-35	I5	NLKST(I)	–	0 to 2	2	Number of links influencing the street canyon receptor
	36-45	F10.0	WST(I)	ft	> 0	60.	Receptor street width
	46-55	F10.0	BUILDH(I)	ft	> 0	70.	Building height
	56-60	F5.0	RECHGT(I)	ft	> 0	5.	Receptor height
	61-66	LI3	ISTLIN(I,J), J = 1, 2		≥ 0	1, 2	Identification numbers for those links affecting receptor I
	67-70	I4	IRSIDE(I)		1 or 2	1	Side of the street on which receptor I is located with respect to the street heading specified by AST(I) (1 = right, 2 = left)

(continued)

TABLE 2 (continued).

Card	Column	Format	Symbol	Units	Value limits	Typical value (default value in parentheses)	Description
10	1-2	I2	IC	-	10	10	Card type number
	3-10	F8.2	THETA(I)	deg from N	0. to 360.	101.	Wind direction
	11-18	F8.2	WS(I)	knots	$\geq$ 1.94	3.89	Wind speed
	19-23	I5	ISTAB(I)	-	1 to 5	4	Stability class
	or		or				or
	19-23	I5	ICLD(I)	tenths	0 to 10	-	Cloud cover

HEADER CARD (CARD TYPE 0)

The first data card for each time period is a header card containing the date and a descriptor that identifies the run. The descriptor may consist of any 70 alphanumeric characters. Column 80 must contain a 1 for the modified version of ISMAP to be run correctly.

RUN DESCRIPTION (CARD TYPE 1)

For each time period the header card is followed by card type 1. This card denotes how many of each of the other card types (2 through 10) should be read for that time period. If the data on a card type does not change from one time period to the next, the card type 1 for the second time period should have a zero or spaces in the proper columns to indicate that the data have not changed and no new cards of a particular type should be read.

Card type 1 also contains the time of day, TOD, and the day of week, DOW. The input value for time of day should be the beginning of the period being simulated; e.g., when computing hourly concentrations, TOD is 8 for the period 0800 to 0900 LST. For day of week, 1 is Sunday, 2 is Monday, and so forth. The length of time being simulated with each iteration of ISMAP is specified as TP. This value can be meaningfully varied from 900 sec (15 minutes) to 7200 sec (2 hours). In general, traffic counts and demands and meteorological data are available as 1-hour summaries; for this reason the usual value for TP is 3600. The trips attracted to and generated by the interior zones during a time period of duration TP are read as TOTATT and TOTGEN. The input value for the parking density constant is ignored by the modified version of ISMAP. This variable is set equal to an extremely low value to avoid routing of vehicles between interior zones.

The number of iterations of the route selection process may be input using the ITM parameter. The default value for ITM is 1, which means that one iteration of the route selection process will be made in which four alternative routes will be selected between each external-internal and internal-external zone pair. Vehicles are routed onto these alternative routes based on travel time and gate capacity. When ITM is 2, the route selection process is carried out twice and half of the vehicles to be routed are applied to the four initially selected alternative routes, and half the vehicles are applied to the second four alternative routes. When ITM is greater than 1, vehicles can be routed around intersections that are predicted to be congested by earlier route selection iterations. A computer run time increase of about 60 percent can be expected when specifying ITM = 2 rather than ITM = 1, when only one receptor location (card type 9) has been specified.

LINK DESCRIPTION (CARD TYPE 2)

Each direction of flow on a street within the network is assigned a link number. A card type 2 is read for each link and includes the link number. The average number of lanes of traffic that make up the link are specified, and link capacity is determined when such capacity is not input (at 1200 veh/lane for speeds less than 70 ft/sec and 1800 veh/lane for speeds greater than 70 ft/sec).

Coordinates must be read that specify each link's end points in the horizontal plane. There are dire consequences if zones are defined by links that do not connect at their end points (see card type 4 description). Coordinates are input in feet from some arbitrary origin and must be positive.

Link capacity may be input or left to be computed by ISMAP. Usually a network is very insensitive to link capacity, intersection capacity being the controlling variable. The exception is a freeway or other limited access road, which might be included in the simulated network.

Speed, in ft/sec, on a link indicates the average speed that vehicles travel during low-use periods. This can usually be estimated as the speed limit on the link.

Link-to-link connections are specified for each link going straight, right, and left. When a link does not lead to another link, the connection should be input as zero or spaces.

An emission height must be input for elevated links. When modeling a facility located on relatively flat terrain, an emission height of zero is normally input for all traffic links. While a more realistic value of the emission height would be 0.5 m, this change would cause little effect upon the calculated concentration if an initial vertical diffusion of 3 m were specified. At present, depressed roadways are treated as surface emission sources, while roadways upwind of aerodynamic barriers may be assigned effective emission heights, as necessary.

INTERSECTION DESCRIPTION (CARD TYPE 3)

An intersection description is required for each intersection in the network. An intersection can be defined at each connection between two or more links. The link numbers of streets entering the intersection from the north, east, south, and west are input parameters. If there is no link approaching from some cardinal direction, spaces or zeros should be entered.

A type of control for each intersection (I) must be specified as follows:

ITYPC(I) = -2	indicates a 4-way arterial stop.
ITYPC(I) = -1	indicates a 2-way arterial stop. The approach with the largest volume is considered to be uncontrolled, while the cross traffic links are assumed to have stop signs.
ITYPC(I) = 0	indicates no control on any approach.
ITYPC(I) = 1	indicates a fixed time signal controller.

ITYPC(I) = 2 indicates a vehicle-actuated (VA) controller with possible separate left turning phases for north-south and east-west traffic.

ITYPC(I) = 3 indicates a vehicle-actuated controller with each intersection approach controlled by a separate signal phase. During a green phase, left-turning vehicles do not have to cross an opposing traffic flow since the opposing traffic experiences a red signal.

ITYPC(I) = 4 indicates type 2 control on the north-south approaches and type 3 control on the east-west approaches. The north-south approaches are controlled by a vehicle-actuated controller during phase 1 of the signal cycle. If phase 3 is specified as having other than zero phase length, left turns from the north-south directions are made during phase 3. If phase 3 is specified as having a zero phase length, then left-turning vehicles are assumed to move during phase 1 and experience opposing vehicular flow. Phases 2 and 4 are exclusive green phases for the east and west approaches, respectively.

ITYPC(I) = 5 indicates type 3 control on the north-south approaches and type 2 control on the east-west approaches. Phases 1 and 3 are exclusive green phases controlling the north and south approaches, respectively. The east-west approaches are controlled by a vehicle-actuated controller during phase 2 of the signal cycle. If phase 4 is specified as having other than zero phase length, left turns from the east-west directions are made during this phase. If phase 4 is specified as having a zero phase length, left-turning vehicles are assumed to move during phase 2 and to experience opposing vehicular flow.

The five controller types are intended to allow any fixed time or VA type control to be simulated. The controllers are simplified, however, and cannot be used to exactly replicate leading or lagging left-turn phases, or 8-phase-full-quad-left signal controllers. It is often possible to adjust the capacity of an intersection to compensate for the lack of a more sophisticated simulated signal controller, and produce meaningful results. An 8-phase-quad-left control is best simulated by a type 2 controller.

Each signal controller must have a maximum cycle length specified, as well as nonzero phase times for each operating signal phase. For example, a simple fixed-time signal might have a 60-sec cycle time and 30-sec north-south and 30-sec east-west phase lengths. If phase 3 and phase 4 times are set to zero, the ISMAP simulation assumes left turns are made through opposing

traffic during phases 1 and 2. When specifying VA control, each phase time represents a minimum phase length. Green time is apportioned among the phases in proportion to the largest demand capacity ratio on the controlled approaches. For example, a type 2 controlled intersection might have a maximum cycle length of 240 sec, minimum north-south phase 1 and east-west phase 2 green times of 30 sec, and minimum north-south left-turn phase 3 and east-west left-turn phase 4 green times of 10 sec. This signal could operate at an 80-sec cycle time under low demand conditions (30+30+10+10) and at a 240-sec cycle time under high demand conditions. (Actually, if there is no demand for the east-west left-turn phase 4, phase skippability will be assumed and a 70-sec minimum cycle time would be possible.)

A yellow clearance interval, CI(I), is specified for each controller. This interval is applied to each phase of each controller and decreases the effective green time of each signal phase by CI(I)/2.

Four approach capacities may be specified, one for each approach (or phase if the intersection is signalized). If an input capacity parameter is left blank, ISMAP will determine a capacity based on the number of lanes on the approach link and the type of intersection control. Capacity of a signalized intersection approach can be determined by selecting the capacity per hour of green based on the road width from Figures 6.6 through 6.13 in the Highway Capacity Manual.[9] The capacity so selected assumes 10 percent left-turning and 10 percent right-turning vehicles. ISMAP adjusts the input capacity based on the percentage of vehicles actually routed left and right in the modeled network. When a 3 or 4 phase and type 1 or 2 signal is specified, the 3rd phase is assumed to be the north-south left-turn phase and the 4th phase is assumed to be the east-west left-turn phase. The capacities input should be the maximum of the left-turning approaches during the 3rd or 4th phase. A type 3 intersection assumes the 3rd and 4th phases control the south and east approaches and the 3rd and 4th capacities are the capacities of these approaches.

Besides an adjustment for number of turning vehicles, ISMAP assumes there is a left-turn lane where left-turn phases are specified. This has the effect of increasing capacity to the through vehicles, as well as to the left-turning vehicles.

ZONE DESCRIPTION (CARD TYPE 4)

Vehicle routing takes place from exterior zones to interior zones with the facility and vice versa. In the original ISMAP interior zones were parking zones surrounded by links of the network. Exterior zones are usually assigned to each exterior street from which vehicles enter or leave the network. Interior zones are identified by numbers 01 to 50 while exterior zones are numbered 51 to 99.

Each zone identified will have an attraction factor (ZATTR) and a generation factor (ZGENR) assigned to it by the user. These factors may vary from hour to hour (as would be the case for an office building parking lot located within a shopping center) or may remain constant throughout the day. The

factors may be in units of percent of TOTATT and TOTGEN (input on card type 1) or in units of vehicles attracted and generated by the zone during a time period, but the units must be consistent on all zone cards. The factors are summarized within ISMAP for all exterior and all interior zones, and a percentage of the total is determined for each zone and is multiplied by TOTATT and TOTGEN to determine the vehicles attracted or generated. TOTATT is the number of trips attracted to a facility from exterior zones. Therefore, each ZATTR specified for an interior zone will represent a part of TOTATT, while each ZATTR specified for an exterior zone will represent a part of TOTGEN.

Vehicle trips to a zone may be in terms of autos, buses, and trucks. Autos are the only vehicles that will use parking capacity. The percent of each type of trip to each zone is an input parameter.

When a run is initialized on the computer, zero cars are assumed to be parked in each zone unless a parked vehicle value is input. Since the number of parked vehicles is saved from iteration to iteration, it is unnecessary to input parked vehicle values except on the first iteration of ISMAP.

A parking zone capacity (PLC) is input in a number of vehicles. This parameter, if left blank, will be calculated by ISMAP based on the parking zone area. It is important to input PLC when a building or other facility is present within the parking zone.

The number of sides that define each parking zone (NZL) must be input for each interior zone. A parking zone is defined as a polygon of no more than six sides. The links that surround the zone should be input (as ZLINKS) in an order such that each link immediately follows the link to which it is connected. When defining a zone, if the last link does not connect with the first link, the program will construct a side to close the zone. The constructed side should not be included in NZL. For the modified ISMAP, NZL must be input as 2 of less, and the links surrounding the zone, ZLINKS, must be identified as the two links entering and leaving the internal zone.

LOCAL AREA DESCRIPTION (CARD TYPE 5)

Local area descriptors appear on card type 5. The user must input the percentage of heavy duty vehicles in the total traffic, the altitude and latitude of the site, and the background CO concentrations at the site. The size of the grid into which area source emissions will be parceled also must be specified. This input parameter depends somewhat on the size of the area to be modeled. Grid sizes of 5 to 10 m are deemed appropriate for applications such as shopping centers. An input value that is too small will result in the generation of arrays requiring more computer storage than has been set aside. To determine the minimum grid size allowed, divide the maximum value of the link end point coordinates by 298.001. It should be noted that computer running costs will increase nonlinearly as the grid size decreases. A default value of 10 m will be used for the grid size if 0 is input. A tuning factor is input on this card, which can be used to adjust modeled emissions, based on measured emissions, or to adjust emissions based on different emission factors found in other countries.

Card type 5 also requires specification of a distance representing initial vertical diffusion. Suggested values for this parameter are in the range of 3 to 10 m. If a value less than 3 m is input, ISMAP will set the constant equal to 3 m.

Also input on this card are the region, temperature, cold and hot starts, and the fractions of the vehicle population of light-duty vehicles, light-duty trucks, heavy-duty gasoline fueled vehicles, heavy-duty diesel-fueled vehicles and motorcycles. The final input variables on this card are indicators of whether the supplemental correction factors are to be used. (These factors are identical to MOBILE1.) If used cards 5A, 5B and/or 5C must be supplied.)

GATE DESCRIPTION (CARD TYPE 6)

Each entrance and exit gate to an internal zone must be specified by a gate description card. Two links define a gate: one is the link by which vehicles leave the center, and the second is the link by which vehicles enter the center. The capacity for each gate is determined by the program based on input or assumed link capacities. Recall that vehicles are allocated to four alternate routes in inverse proportion to travel time and volume to capacity ration of the gate used. Thus, all trips between external and internal zones must be routed through a gate. If a route exists between an external and internal zone that does not go through a gate, the program will fail without any explanation.

VEHICLE CHARACTERISTICS (CARD TYPE 7)

This card is read for each type of vehicle entering the facility. The three possible types of vehicles are autos, buses, and trucks. Each vehicle should have a load factor associated with it, indicating the number of person trips to the facility that are associated with each vehicle.

VEHICLE COUNTS (CARD TYPE 8)

The purpose of this card is to specify the nonfacility vehicles that use the street network. The links and intersections outside the facility will handle vehicle trips to the facility as well as other trips. The Vehicle Count cards are used to apply the nonfacility-oriented trips to the street network. The counts for as many as four links can be included on each card. The count on a link is broken down into vehicles traveling straight, turning right, or turning left at the downstream intersection of the link with other links. It is convenient to include the counts of four approaches to an intersection on a card. The input parameters are:

- North approach link number (L) followed by the counts on this approach traveling straight, right, and left.

- East approach link number (L2) followed by the counts on this approach traveling straight, right, and left.

- South approach link number (L3) followed by the counts on this approach traveling straight, right, and left.

- West approach link number (L4) followed by the counts on this approach traveling straight, right, and left.

The facility-oriented vehicles are added to the Vehicle Counts during an ISMAP run to determine the total vehicles on each link of the network.

RECEPTOR DESCRIPTION (CARD TYPE 9)

The x- and y-coordinates of each receptor location are input on card type 9. The values for these coordinates are in feet and must be positive and based on the same origin used for link coordinates. If the receptor is located in a potential street canyon the user must also specify the relevant street canyon characteristics discussed earlier in this section.

METEOROLOGICAL VARIABLES (CARD TYPE 10)

Card type 10 contains the meteorological information necessary for dispersion calculations. The user has the option of specifying the stability class as input or allowing ISMAP to calculate stability class from cloud cover and wind speed data. However, the stability determination capability of ISMAP can only be used when the time period being simulated is 1 hour. If stability is to be input, one card type 10, with values for wind direction, wind speed, and stability class, is input with traffic and site descriptors for each time period (durations of from 900 to 7200 sec are possible). If ISMAP is to calculate stability class, 24 card type 10s containing observations of wind direction, wind speed, and cloud cover for 24 hours of the day are input with the traffic and site data for the first hour for which CO concentrations are to be computed. If stability is to be found by ISMAP, the meteorological data for all 24 hours of a day is input together, even if CO computations are to be made for only a few hours of the day. This is necessary for program efficiency. Since the stability class for each hour of the day will be found during dispersion calculations for the first hour, no card type 10 is input with traffic and site data for subsequent hours until a new day is to be run. If the wind is calm, wind speed should be input on card type 10 as 1.94 knots (1 m/sec) and the wind direction from the previous time period should be used.

SECTION 5

SAMPLE APPLICATION

For a sample application of the modified ISMAP model we have chosen the configuration of links, intersections, zones and receptors shown in Figure 12. For this particular example the traffic flow within the network is generated by means of vehicle attractions and generations for six interior zones and six exterior zones. When setting up a network such as this one, it must be remembered that vehicles cannot be routed between two interior zones or two exterior zones. For each of the interior zones a gate has been designated with associated incoming and outgoing links. A total of nine receptor locations are used in this simulation with the central receptor considered to be a street canyon receptor.

A listing of the model input cards for this sample case follows Figure 12. The format for each card is given by Table 2. The first card in the list is the leader card which gives the date for which the simulation is being carried out (not the date of the computer run), the run identities and a flag in column 80 which must be set equal to 1 for the modified version of ISMAP to be run properly. The following card (card 1) indicates that for the simulation there will be 48 links, 9 intersections, 12 zones, 1 type 5 card, 6 gates, 3 vehicle types, no VMT which is not due to zone attractions, 9 receptor locations and 1 hour's worth of meteorological data. Also input to the program on this card is the hour of the day (8), time period duration (3600 sec), total trips attracted to the interior zones (1800), total trips generated by the interior zones (300), parking density constant (360 ft^2 - not used) and the number of iterations to be used in the route selection process (3). For each link a type 2 card is required giving number of lanes, endpoint coordinates in feet, cruise speed on the link (ft/sec), link connections (going straight, turning right and turning left), and emission height (meters). For example, if we follow link 21 along the direction from the first pair of endpoint coordinates to the second, we meet links 23, 46 and 47. For each of the links used in this example the cruise speed was assumed to be 58 ft/sec (40 mi/hr) and the emission height set equal to zero. For each of the nine intersections a type 3 card is provided giving the identification numbers of links approaching the intersection; signalization type (fixed time for the example case); cycle time (200 sec); duration of North-South, East-West, North-South left turn and East-West left turn phases (75, 75, 25, 25 sec); and yellow time interval (3 sec). The default value of 1200 veh/hr/lane was used as the capacity of each phase. The card type 4 entries identify the interior zones (01-06) and the exterior zones and give the number of trips attracted to or generated by the zone as a fraction of total trips attracted to and generated by all interior and exterior zones. The percentage values

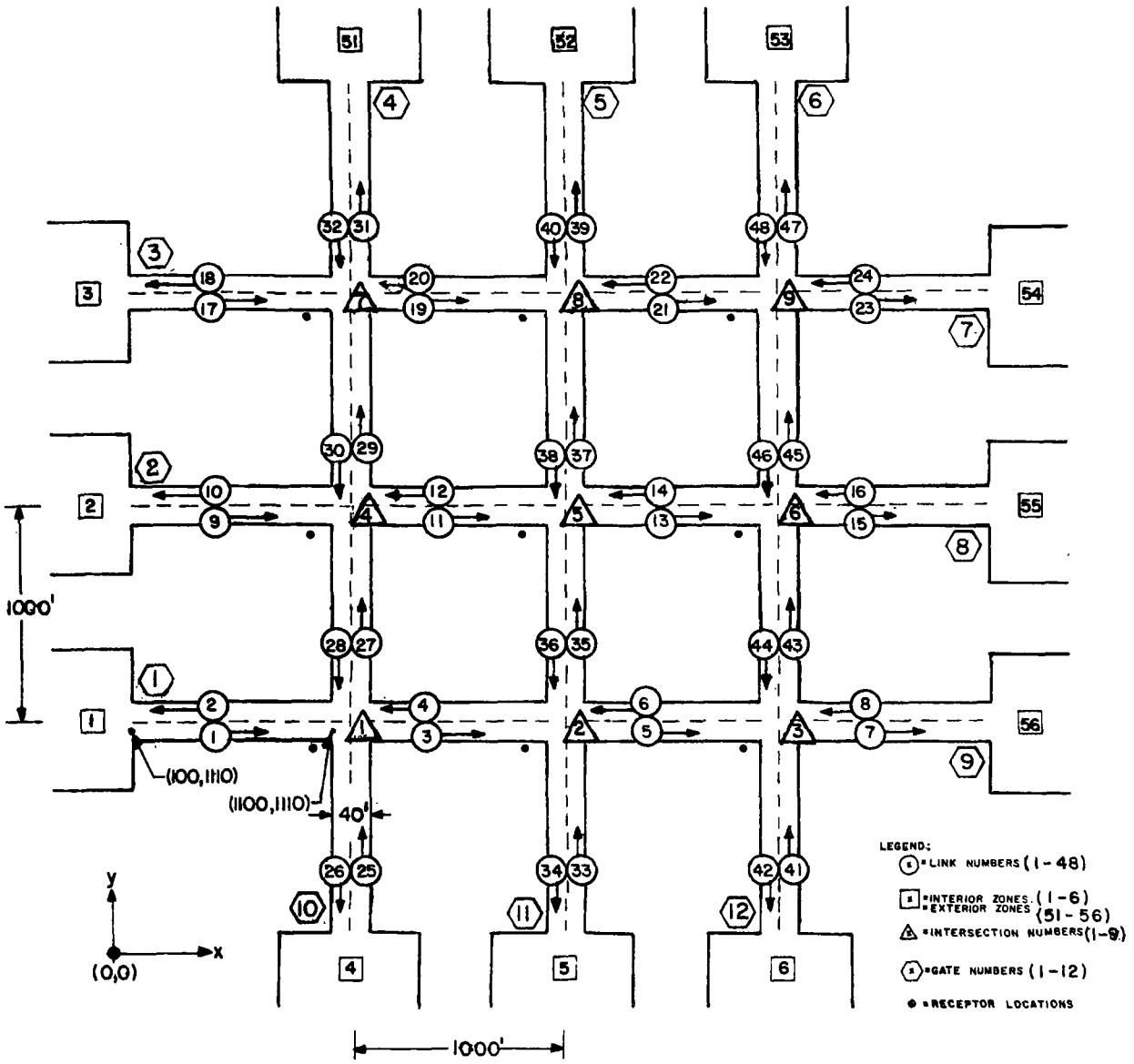

Figure 12. Diagram for test case of modified ISMAP.

```
WELT.L  ISMAPDATA
ELT007 SL73R1 07/20/79 18:44:56 (2,)
000001      000   770301EXAMPLE HOT SPOT CALCULATION WITH MODIFIED ISMAP                    1
000002      000    1 48    9   12    1    6    3    0    9    1   6.   3.3600.1800. 300. 360.    3
000003      000    2 01    21100.1110.0100.1110.        58.   03   26   57    0
000004      000    2 02    21100.1130.0100.1130.        58.   00   00   00    0
000005      000    2 03    22100.1110.1100.1110.        58.   05   34   35    0
000006      000    2 04    22100.1130.1100.1130.        58.   02   27   26    0
000007      000    2 05    23100.1110.2100.1110.        58.   07   42   43    0
000008      000    2 06    23100.1130.2100.1130.        58.   04   35   34    0
000009      000    2 07    24100.1110.3100.1110.        58.   00   00   00    0
000010      000    2 08    24100.1130.3100.1130.        58.   06   43   42    0
000011      000    2 09    21100.2110.0100.2110.        58.   11   28   29    0
000012      000    2 10    21100.2130.0100.2130.        58.   00   00   00    0
000013      000    2 11    22100.2110.1100.2110.        58.   13   36   37    0
000014      000    2 12    22100.2310.1100.2130.        58.   10   29   28    0
000015      000    2 13    23100.2110.2100.2110.        58.   15   44   46    0
000016      000    2 14    23100.2130.2100.2130.        58.   12   37   36    0
000017      000    2 15    24100.2110.3100.2110.        58.   00   00   00    0
000018      000    2 16    24100.2130.3100.2130.        58.   14   45   44    0
000019      000    2 17    21100.3110.0100.3110.        58.   19   30   31    0
000020      000    2 18    21100.3130.0100.3130.        58.   00   00   00    0
000021      000    2 19    22100.3110.1100.3110.        58.   21   38   39    0
000022      000    2 20    22100.3130.1100.3130.        58.   18   31   30    0
000023      000    2 21    23100.3110.2100.3110.        58.   23   46   47    0
000024      000    2 22    23100.3130.2100.3130.        58.   20   39   38    0
000025      000    2 23    24100.3110.3100.3110.        58.   00   00   00    0
000026      000    2 24    24100.3130.3100.3130.        58.   22   47   46    0
000027      000    2 25    21130.1100.1130.0100.        58.   27   03   02    0
000028      000    2 26    21110.1100.1110.0100.        58.   00   00   00    0
000029      000    2 27    21130.2100.1130.1100.        58.   29   11   10    0
000030      000    2 28    21110.2100.1110.1100.        58.   26   02   03    0
000031      000    2 29    21130.3100.1130.2100.        58.   31   19   18    0
000032      000    2 30    21110.3100.1110.2100.        58.   28   10   11    0
000033      000    2 31    21130.4100.1130.3100.        58.   00   00   00    0
000034      000    2 32    21110.4100.1110.3100.        58.   30   18   19    0
000035      000    2 33    22130.1100.2130.0100.        58.   35   05   04    0
000036      000    2 34    22110.1100.2110.0100.        58.   00   00   00    0
000037      000    2 35    22130.2100.2130.1100.        58.   37   13   12    0
000038      000    2 36    22110.2100.2110.1100.        58.   34   04   05    0
000039      000    2 37    22130.3100.2130.2100.        58.   39   21   20    0
000040      000    2 38    22110.3100.2110.2100.        58.   36   12   13    0
000041      000    2 39    22130.4100.2130.3100.        58.   00   00   00    0
000042      000    2 40    22110.4100.2110.3100.        58.   38   20   21    0
000043      000    2 41    23130.1100.3110.0100.        58.   43   07   06    0
000044      000    2 42    23110.1100.3110.0100.        58.   00   00   00    0
000045      000    2 43    23130.2100.3130.1100.        58.   45   51   14    0
000046      000    2 44    23110.2100.3110.1100.        58.   42   06   07    0
000047      000    2 45    23130.3100.3130.2100.        58.   47   23   22    0
000048      000    2 46    23110.3100.3110.2100.        58.   44   41   15    0
000049      000    2 47    23130.4100.3130.3100.        58.   00   00   00    0
000050      000    2 48    23110.4100.3110.3100.        58.   46   22   23    0
000051      000    3 01   28   04   25   01    1  200 75. 75. 25. 25.  3.
000052      000    3 02   36   06   33   03    1  200 75. 75. 25. 25.  3.
000053      000    3 03   44   08   41   05    1  200 75. 75. 25. 25.  3.
000054      000    3 04   30   12   27   09    1  200 75. 75. 25. 25.  3.
000055      000    3 05   38   14   35   11    1  200 75. 75. 25. 25.  3.
```

```
000056    000    3 06   46   16   43   13    1  200 75. 75. 25. 25.  3.
000057    000    3 07   32   20   29   17    1  200 75. 75. 25. 25.  3.
000058    000    3 08   40   22   37   19    1  200 75. 75. 25. 25.  3.
000059    000    3 09   48   24   45   21    1  200 75. 75. 25. 25.  3.
000060    000    45612.12.                    1  07  08
000061    000    45512.12.                    1  15  16
000062    000    45412.12.                    1  23  24
000063    000    45312.12.                    1  47  48
000064    000    45212.12.                    1  39  40
000065    000    45112.12.                    1  31  32
000066    000    40612.12.                    1  41  42
000067    000    40512.12.                    1  33  34
000068    000    40412.12.                    1  25  26
000069    000    4 312.12.                    1  17  18
000070    000    4 212.12.                    1   9  10
000071    000    4 112.12.                    1   1   2
000072    000    5 0.0 0.0 0.0 1.0 0.0 1 50. 0.20 0.20 0.20 0.75 0.05 0.05 0.09 0.06 0.00 000
000073    000    6      01   02
000074    000    6      09   10
000075    000    6      17   18
000076    000    6      41   42
000077    000    6      33   34
000078    000    6      25   26
000079    000    7  1 01.4
000080    000    7  2 15.0
000081    000    7  3 02.0
000082    000    9     1000.     1090.
000083    000    9     2000.     1090.
000084    000    9     3000.     1090.
000085    000    9     1000.     2090.
000086    000    9     2000.     2090.  1  90.    2       60.       70.   5. 11 12   1
000087    000    9     3000.     2090.
000088    000    9     1000.     3090.
000089    000    9     2000.     3090.
000090    000    9     3000.     3090.
000091    000   10    120.    5.00    4

END ELT.

@BRKPT PRINT$
```

770301 EXAMPLE HOT SPOT CALCULATION WITH MODIFIED ISMAP 1

```
PARKING LOT EMISSIONS NOT CONSIDERED
1 48    9   12    1    6    3    0    9    1    8.   3.3600.1800. 300. 360.    3
2  1    21100.1110. 100.1110.    0  58.    3   26   57   0.
2  2    21100.1130. 100.1130.    0  58.    0    0    0   0.
2  3    22100.1110.1100.1110.    0  58.    5   34   35   0.
2  4    22100.1130.1100.1130.    0  58.    2   27   26   0.
2  5    23100.1110.2100.1110.    0  58.    7   42   43   0.
2  6    23100.1130.2100.1130.    0  58.    4   35   34   0.
2  7    24100.1110.3100.1110.    0  58.    0    0    0   0.
2  8    24100.1130.3100.1130.    0  58.    6   43   42   0.
2  9    21100.2110. 100.2110.    0  58.   11   28   29   0.
2 10    21100.2130. 100.2130.    0  58.    0    0    0   0.
2 11    22100.2110.1100.2110.    0  58.   13   36   37   0.
2 12    22100.2310.1100.2130.    0  58.   10   29   28   0.
2 13    23100.2110.2100.2110.    0  58.   15   44   46   0.
2 14    23100.2130.2100.2130.    0  58.   12   37   36   0.
2 15    24100.2110.3100.2110.    0  58.    0    0    0   0.
2 16    24100.2130.3100.2130.    0  58.   14   45   44   0.
2 17    21100.3110. 100.3110.    0  58.   19   30   31   0.
2 18    21100.3130. 100.3130.    0  58.    0    0    0   0.
2 19    22100.3110.1100.3110.    0  58.   21   38   39   0.
2 20    22100.3130.1100.3130.    0  58.   18   31   30   0.
2 21    23100.3110.2100.3110.    0  58.   23   46   47   0.
2 22    23100.3130.2100.3130.    0  58.   20   39   38   0.
2 23    24100.3110.3100.3110.    0  58.    0    0    0   0.
2 24    24100.3130.3100.3130.    0  58.   22   47   46   0.
2 25    21130.1100.1130. 100.    0  58.   27    3    2   0.
2 26    21110.1100.1110. 100.    0  58.    0    0    0   0.
2 27    21130.2100.1130.1100.    0  58.   29   11   10   0.
2 28    21110.2100.1110.1100.    0  58.   26    2    3   0.
2 29    21130.3100.1130.2100.    0  58.   31   19   18   0.
2 30    21110.3100.1110.2100.    0  58.   28   10   11   0.
2 31    21130.4100.1130.3100.    0  58.    0    0    0   0.
2 32    21110.4100.1110.3100.    0  58.   30   18   19   0.
2 33    22130.1100.2130. 100.    0  58.   35    5    4   0.
2 34    22110.1100.2110. 100.    0  58.    0    0    0   0.
2 35    22130.2100.2130.1100.    0  58.   37   13   12   0.
2 36    22110.2100.2110.1100.    0  58.   34    4    5   0.
2 37    22130.3100.2130.2100.    0  58.   39   21   20   0.
2 38    22110.3100.2110.2100.    0  58.   36   12   13   0.
2 39    22130.4100.2130.3100.    0  58.    0    0    0   0.
2 40    22110.4100.2110.3100.    0  58.   38   20   21   0.
2 41    23130.1100.3110. 100.    0  58.   43    7    6   0.
2 42    23110.1100.3110. 100.    0  58.    0    0    0   0.
2 43    23130.2100.3130.1100.    0  58.   45   51   14   0.
2 44    23110.2100.3110.1100.    0  58.   42    6    7   0.
2 45    23130.3100.3130.2100.    0  58.   47   23   22   0.
2 46    23110.3100.3110.2100.    0  58.   44   41   15   0.
2 47    23130.4100.3130.3100.    0  58.    0    0    0   0.
2 48    23110.4100.3110.3100.    0  58.   46   22   23   0.
3  1   28    4   25    1    1  200 75. 75. 25. 25.  3.  0.  0.  0.  0.
3  2   36    6   33    3    1  200 75. 75. 25. 25.  3.  0.  0.  0.  0.
3  3   44    8   41    5    1  200 75. 75. 25. 25.  3.  0.  0.  0.  0.
3  4   30   12   27    9    1  200 75. 75. 25. 25.  3.  0.  0.  0.  0.
3  5   38   14   35   11    1  200 75. 75. 25. 25.  3.  0.  0.  0.  0.
```

```
3  6   46   16   43   13    1  200 75. 75. 25. 25.  3.  0.  0.  0.  0.
3  7   32   20   29   17    1  200 75. 75. 25. 25.  3.  0.  0.  0.  0.
3  8   40   22   37   19    1  200 75. 75. 25. 25.  3.  0.  0.  0.  0.
3  9   48   24   45   21    1  200 75. 75. 25. 25.  3.  0.  0.  0.  0.
4 5612.12.1.0 .0 .0   0.   0. 1   7   8
4 5512.12.1.0 .0 .0   0.   0. 1  15  16
4 5412.12.1.0 .0 .0   0.   0. 1  23  24
4 5312.12.1.0 .0 .0   0.   0. 1  47  48
4 5212.12.1.0 .0 .0   0.   0. 1  39  40
4 5112.12.1.0 .0 .0   0.   0. 1  31  32
4  112.12.1.0 .0 .0   0.   0. 1   1   2
4  212.12.1.0 .0 .0   0.   0. 1   9  10
4  312.12.1.0 .0 .0   0.   0. 1  17  18
4  412.12.1.0 .0 .0   0.   0. 1  25  26
4  512.12.1.0 .0 .0   0.   0. 1  33  34
4  612.12.1.0 .0 .0   0.   0. 1  41  42
5  0.   0.   0.   1.   0. 1 50. .200 .200 .200 .750 .050 .050 .090 .060 .000 000
6        1    2
6        9   10
6       17   18
6       41   42
6       33   34
6       25   26
 9 304.80000 332.23200  0   0.    0        0.        0.   0.  0  0  0
 9 609.60000 332.23200  0   0.    0        0.        0.   0.  0  0  0
 9 914.39999 332.23200  0   0.    0        0.        0.   0.  0  0  0
 9 304.80000 637.03200  0   0.    0        0.        0.   0.  0  0  0
 9 609.60000 637.03200  1  90.    2       18.       21.   2. 11 12  1
 9 914.39999 637.03200  0   0.    0        0.        0.   0.  0  0  0
 9 304.80000 941.83199  0   0.    0        0.        0.   0.  0  0  0
 9 609.60000 941.83199  0   0.    0        0.        0.   0.  0  0  0
 9 914.39999 941.83199  0   0.    0        0.        0.   0.  0  0  0
10  120.00    5.00    4
```

PARKING ZONE CAPACITIES AND LENGTHS NOT USED IN THE CALCULATION

PAFFIC SUBMODEL

INTERSECTION 1

	PHASE 1		PHASE 2		PHASE 3		PHASE 4	
TIME	75.		75.		25.		25.	
V/GC	.13803		.06007		.04167		.08333	
	NORTH-AP	SOUTH-AP	EAST-AP	WEST-AP	N-AP-LFT	S-AP-LFT	E-AP-LFT	W-AP-LFT
DELA	46.	40.	42.	40.	87.	77.	115.	77.
QUEUE	0.	0.	0.	0.	0.	0.	1.	0.
VOLUME	400.	50.	174.	50.	50.	0.	100.	0.
CAPACITY	1065.	1065.	1065.	1217.	141.	141.	141.	141.
V/GCAP	.13803	.01725	.06007	.01510	.04167	.00000	.08333	.00000

INTERSECTION 2

	PHASE 1	PHASE 2	PHASE 3	PHASE 4
TIME	75.	75.	25.	25.
V/GC	.10352	.06890	.00476	.05566

	NORTH-AP	SOUTH-AP	EAST-AP	WEST-AP	N-AP-LFT	S-AP-LFT	E-AP-LFT	W-AP-LFT
DELA	44.	40.	42.	40.	77.	77.	87.	93.
QUEUE	0.	0.	0.	0.	0.	0.	0.	0.
VOLUME	300.	44.	210.	61.	5.	6.	50.	67.
CAPACITY	1065.	1150.	1118.	1065.	141.	141.	141.	141.
V/GCAP	.10752	.01415	.06890	.02104	.00384	.00476	.04167	.05566

INTERSECTION 3

	PHASE 1	PHASE 2	PHASE 3	PHASE 4
TIME	75.	75.	25.	25.
V/GC	.07548	.08235	.04960	.04167

	NORTH-AP	SOUTH-AP	EAST-AP	WEST-AP	N-AP-LFT	S-AP-LFT	E-AP-LFT	W-AP-LFT
DELA	43.	40.	43.	39.	77.	90.	87.	77.
QUEUE	0.	0.	0.	0.	0.	0.	0.	0.
VOLUME	250.	71.	250.	21.	4.	60.	50.	0.
CAPACITY	1217.	1065.	1116.	1217.	141.	141.	141.	141.
V/GCAP	.07548	.02467	.08235	.00647	.00347	.04960	.04167	.00000

INTERSECTION 4

	PHASE 1	PHASE 2	PHASE 3	PHASE 4
TIME	75.	75.	25.	25.
V/GC	.22429	.01510	.04167	.01934

	NORTH-AP	SOUTH-AP	EAST-AP	WEST-AP	N-AP-LFT	S-AP-LFT	E-AP-LFT	W-AP-LFT
DELA	53.	40.	40.	39.	77.	87.	77.	81.
QUEUE	1.	0.	0.	0.	0.	0.	0.	0.
VOLUME	650.	46.	50.	27.	0.	50.	0.	23.
CAPACITY	1065.	1217.	1217.	1217.	141.	141.	141.	141.
V/GCAP	.22429	.01399	.01510	.00809	.00000	.04167	.00000	.01934

INTERSECTION 5

	PHASE 1	PHASE 2	PHASE 3	PHASE 4
TIME	75.	75.	25.	25.
V/GC	.05051	.01510	.04167	.12500

	NORTH-AP	SOUTH-AP	EAST-AP	WEST-AP	N-AP-LFT	S-AP-LFT	E-AP-LFT	W-AP-LFT
DELA	41.	41.	40.	39.	87.	77.	422.	77.
QUEUE	0.	0.	0.	0.	0.	0.	12.	0.
VOLUME	150.	146.	50.	27.	50.	0.	150.	0.
CAPACITY	1217.	1065.	1217.	1130.	141.	141.	141.	141.
V/GCAP	.04529	.05051	.01510	.00871	.04167	.00000	.12500	.00000

INTERSECTION 6

	PHASE 1	PHASE 2	PHASE 3	PHASE 4
TIME	75.	75.	25.	25.
V/GC	.08115	.08235	.00459	.05509

	NORTH-AP	SOUTH-AP	EAST-AP	WEST-AP	N-AP-LFT	S-AP-LFT	E-AP-LFT	W-AP-LFT
DELA	43.	40.	43.	41.	77.	77.	87.	92.
QUEUE	0.	0.	0.	0.	0.	0.	0.	0.
VOLUME	235.	97.	250.	94.	6.	0.	50.	66.
CAPACITY	1065.	1217.	1116.	1065.	141.	141.	141.	141.
V/GCAP	.08115	.02931	.08235	.03261	.00459	.00000	.04167	.05509

INTERSECTION 7

	PHASE 1	PHASE 2	PHASE 3	PHASE 4
TIME	75.	75.	25.	25.
V/GC	.08235	.07785	.04167	.37500

	NORTH-AP	SOUTH-AP	EAST-AP	WEST-AP	N-AP-LFT	S-AP-LFT	E-AP-LFT	W-AP-LFT
DELA	43.	40.	43.	40.	87.	77.	4252.	78.

QUEUE	0.	0.	0.	0.	0.	0.	112.	0.
VOLUME	250.	70.	256.	43.	50.	0.	450.	7.
CAPACITY	1116.	1065.	1206.	1217.	141.	141.	141.	141.
V/GCAP	.08235	.02399	.07785	.01288	.04167	.00000	.37500	.00613

INTERSECTION 8

	PHASE 1		PHASE 2		PHASE 3		PHASE 4	
TIME	75.		75.		25.		25.	
V/GC	.10352		.15097		.00459		.08333	
	NORTH-AP	SOUTH-AP	EAST-AP	WEST-AP	N-AP-LFT	S-AP-LFT	E-AP-LFT	W-AP-LFT
DELA	44.	40.	47.	41.	77.	77.	115.	80.
QUEUE	0.	0.	0.	0.	0.	0.	1.	0.
VOLUME	300.	52.	500.	107.	0.	6.	100.	18.
CAPACITY	1065.	1065.	1217.	1217.	141.	141.	141.	141.
V/GCAP	.10352	.01810	.15097	.03225	.00000	.00459	.08333	.01521

INTERSECTION 9

	PHASE 1		PHASE 2		PHASE 3		PHASE 4	
TIME	75.		75.		25.		25.	
V/GC	.10352		.07548		.08333		.04167	
	NORTH-AP	SOUTH-AP	EAST-AP	WEST-AP	N-AP-LFT	S-AP-LFT	E-AP-LFT	W-AP-LFT
DELA	44.	40.	43.	41.	77.	115.	87.	81.
QUEUE	0.	0.	0.	0.	0.	1.	0.	0.
VOLUME	300.	47.	250.	101.	0.	100.	50.	26.
CAPACITY	1065.	1065.	1217.	1065.	141.	141.	141.	141.
V/GCAP	.10352	.01624	.07548	.03486	.00000	.08333	.04167	.02205

INTERSECTION 1

	PHASE 1		PHASE 2		PHASE 3		PHASE 4	
TIME	75.		75.		25.		25.	
V/GC	.12077		.07291		.02083		.08333	
	NORTH-AP	SOUTH-AP	EAST-AP	WEST-AP	N-AP-LFT	S-AP-LFT	E-AP-LFT	W-AP-LFT
DELA	45.	40.	43.	40.	81.	77.	115.	77.
QUEUE	0.	0.	0.	0.	0.	0.	1.	0.
VOLUME	350.	50.	212.	50.	25.	0.	100.	0.
CAPACITY	1065.	1065.	1069.	1217.	141.	141.	141.	141.
V/GCAP	.12077	.01725	.07291	.01510	.02083	.00000	.08333	.00000

INTERSECTION 2

	PHASE 1		PHASE 2		PHASE 3		PHASE 4	
TIME	75.		75.		25.		25.	
V/GC	.08627		.08531		.02395		.08333	
	NORTH-AP	SOUTH-AP	EAST-AP	WEST-AP	N-AP-LFT	S-AP-LFT	E-AP-LFT	W-AP-LFT
DELA	43.	40.	43.	40.	82.	77.	115.	89.
QUEUE	0.	0.	0.	0.	0.	0.	1.	0.
VOLUME	250.	47.	268.	35.	29.	3.	100.	56.
CAPACITY	1065.	1115.	1154.	1065.	141.	141.	141.	141.
V/GCAP	.08627	.01553	.08531	.01204	.02395	.00238	.08333	.04643

INTERSECTION 3

	PHASE 1		PHASE 2		PHASE 3		PHASE 4	
TIME	75.		75.		25.		25.	
V/GC	.12077		.08235		.03565		.04167	
	NORTH-AP	SOUTH-AP	EAST-AP	WEST-AP	N-AP-LFT	S-AP-LFT	E-AP-LFT	W-AP-LFT
DELA	45.	41.	43.	40.	77.	85.	87.	77.
QUEUE	0.	0.	0.	0.	0.	0.	0.	0.
VOLUME	350.	109.	250.	43.	2.	43.	50.	5.

CAPACITY	1065.	1077.	1116.	1065.	141.	141.	141.	141.
V/GCAP	.12077	.03730	.08235	.01473	.00174	.03565	.04167	.00450

INTERSECTION 4

	PHASE 1		PHASE 2		PHASE 3		PHASE 4	
TIME	75.		75.		25.		25.	
V/GC	.13803		.06376		.04496		.08333	
	NORTH-AP	SOUTH-AP	EAST-AP	WEST-AP	N-AP-LFT	S-AP-LFT	E-AP-LFT	W-AP-LFT
DELA	46.	40.	42.	40.	88.	81.	115.	79.
QUEUE	0.	0.	0.	0.	0.	0.	1.	0.
VOLUME	400.	71.	192.	36.	54.	25.	100.	14.
CAPACITY	1065.	1068.	1106.	1217.	141.	141.	141.	141.
V/GCAP	.13803	.02459	.06376	.01082	.04496	.02083	.08333	.01180

INTERSECTION 5

	PHASE 1		PHASE 2		PHASE 3		PHASE 4	
TIME	75.		75.		25.		25.	
V/GC	.10352		.05492		.02083		.06250	
	NORTH-AP	SOUTH-AP	EAST-AP	WEST-AP	N-AP-LFT	S-AP-LFT	E-AP-LFT	W-AP-LFT
DELA	44.	41.	42.	41.	81.	77.	96.	78.
QUEUE	0.	0.	0.	0.	0.	0.	1.	0.
VOLUME	300.	123.	176.	104.	25.	4.	75.	6.
CAPACITY	1065.	1065.	1179.	1078.	141.	141.	141.	141.
V/GCAP	.10352	.04239	.05492	.03562	.02083	.00345	.06250	.00534

INTERSECTION 6

	PHASE 1		PHASE 2		PHASE 3		PHASE 4	
TIME	75.		75.		25.		25.	
V/GC	.11358		.07123		.04262		.07470	
	NORTH-AP	SOUTH-AP	EAST-AP	WEST-AP	N-AP-LFT	S-AP-LFT	E-AP-LFT	W-AP-LFT
DELA	45.	40.	42.	41.	77.	87.	96.	105.
QUEUE	0.	0.	0.	0.	0.	0.	1.	1.
VOLUME	329.	83.	225.	97.	3.	51.	75.	90.
CAPACITY	1065.	1217.	1161.	1065.	141.	141.	141.	141.
V/GCAP	.11358	.02516	.07123	.03356	.00229	.04262	.06250	.07470

INTERSECTION 7

	PHASE 1		PHASE 2		PHASE 3		PHASE 4	
TIME	75.		75.		25.		25.	
V/GC	.08984		.06157		.04167		.18750	
	NORTH-AP	SOUTH-AP	EAST-AP	WEST-AP	N-AP-LFT	S-AP-LFT	E-AP-LFT	W-AP-LFT
DELA	43.	40.	42.	40.	81.	87.	2452.	78.
QUEUE	0.	0.	0.	0.	0.	0.	65.	0.
VOLUME	275.	56.	203.	42.	25.	50.	225.	8.
CAPACITY	1125.	1065.	1210.	1170.	141.	141.	141.	141.
V/GCAP	.08984	.01949	.06157	.01327	.02083	.04167	.18750	.00647

INTERSECTION 8

	PHASE 1		PHASE 2		PHASE 3		PHASE 4	
TIME	75.		75.		25.		25.	
V/GC	.10352		.09058		.00230		.12500	
	NORTH-AP	SOUTH-AP	EAST-AP	WEST-AP	N-AP-LFT	S-AP-LFT	E-AP-LFT	W-AP-LFT
DELA	44.	40.	43.	40.	77.	77.	537.	79.
QUEUE	0.	0.	0.	0.	0.	0.	15.	0.
VOLUME	300.	54.	300.	67.	0.	3.	150.	13.
CAPACITY	1065.	1065.	1217.	1217.	141.	141.	141.	141.
V/GCAP	.10352	.01852	.09058	.02027	.00000	.00230	.12500	.01097

INTERSECTION 9

	PHASE 1			PHASE 2			PHASE 3			PHASE 4		
TIME	75.			75.			25.			25.		
V/GC	.10352			.06793			.04167			.06250		
	NORTH-AP	SOUTH-AP		EAST-AP	WEST-AP		N-AP-LFT	S-AP-LFT		E-AP-LFT	W-AP-LFT	
DELA	44.	40.		42.	40.		77.	87.		96.	80.	
QUEUE	0.	0.		0.	0.		0.	0.		1.	0.	
VOLUME	300.	58.		225.	65.		0.	50.		75.	19.	
CAPACITY	1065.	1065.		1217.	1065.		141.	141.		141.	141.	
V/GCAP	.10352	.02012		.06793	.02250		.00000	.04167		.06250	.01563	

COUNTS

LINK	THRU	RT	LT	TERM	LINK	THRU	RT	LT	TERM	LINK	THRU	RT	LT	TERM	LINK	THRU	RT	LT	TERM	LINK	THRU	RT	LT	TERM
1	50.	0.	0.	0.	2	0.	0.	0.	300.	3	27.	17.	51.	0.	4	183.	48.	133.	0.	5	16.	33.	8.	0.
6	263.	104.	117.	0.	7	0.	0.	0.	50.	8	217.	33.	50.	0.	9	38.	0.	12.	0.	10	0.	0.	0.	300.
11	60.	19.	11.	0.	12	183.	31.	83.	0.	13	46.	35.	65.	0.	14	142.	26.	83.	0.	15	0.	0.	0.	50.
16	217.	17.	67.	0.	17	39.	3.	8.	0.	18	0.	0.	0.	300.	19	68.	17.	11.	0.	20	217.	2.	150.	0.
21	23.	75.	17.	0.	22	233.	0.	100.	0.	23	0.	0.	0.	50.	24	167.	0.	133.	0.	25	38.	12.	0.	0.
26	0.	0.	0.	300.	27	54.	16.	17.	0.	28	167.	117.	33.	0.	29	40.	23.	33.	0.	30	233.	100.	36.	0.
31	0.	0.	0.	50.	32	217.	50.	33.	0.	33	37.	11.	2.	0.	34	0.	0.	0.	300.	35	68.	68.	56.	0.
36	167.	100.	19.	0.	37	39.	30.	35.	0.	38	183.	100.	17.	0.	39	0.	0.	0.	50.	40	183.	100.	17.	0.
41	69.	31.	51.	0.	42	0.	0.	0.	300.	43	77.	0.	34.	0.	44	217.	217.	3.	0.	45	33.	27.	33.	0.
46	335.	101.	4.	0.	47	0.	0.	0.	50.	48	167.	133.	0.	0.										

INTERSECTION 1

	PHASE 1			PHASE 2			PHASE 3			PHASE 4		
TIME	75.			75.			25.			25.		
V/GC	.09777			.07667			.02778			.11111		
	NORTH-AP	SOUTH-AP		EAST-AP	WEST-AP		N-AP-LFT	S-AP-LFT		E-AP-LFT	W-AP-LFT	
DELA	44.	40.		43.	40.		83.	77.		250.	77.	
QUEUE	0.	0.		0.	0.		0.	0.		6.	0.	
VOLUME	283.	50.		232.	50.		33.	0.		133.	0.	
CAPACITY	1065.	1096.		1111.	1217.		141.	141.		141.	141.	
V/GCAP	.09777	.01677		.07667	.01510		.02778	.00000		.11111	.00000	

INTERSECTION 2

	PHASE 1			PHASE 2			PHASE 3			PHASE 4		
TIME	75.			75.			25.			25.		
V/GC	.09202			.12570			.01596			.09722		
	NORTH-AP	SOUTH-AP		EAST-AP	WEST-AP		N-AP-LFT	S-AP-LFT		E-AP-LFT	W-AP-LFT	
DELA	44.	40.		46.	40.		80.	77.		146.	87.	
QUEUE	0.	0.		0.	0.		0.	0.		2.	0.	
VOLUME	267.	48.		367.	44.		19.	2.		117.	51.	
CAPACITY	1065.	1097.		1074.	1065.		141.	141.		141.	141.	
V/GCAP	.09202	.01611		.12570	.01516		.01596	.00159		.09722	.04282	

INTERSECTION 3

	PHASE 1			PHASE 2			PHASE 3			PHASE 4		
TIME	75.			75.			25.			25.		
V/GC	.14953			.07992			.04211			.04167		
	NORTH-AP	SOUTH-AP		EAST-AP	WEST-AP		N-AP-LFT	S-AP-LFT		E-AP-LFT	W-AP-LFT	
DELA	47.	41.		43.	40.		77.	87.		87.	78.	
QUEUE	0.	0.		0.	0.		0.	0.		0.	0.	
VOLUME	433.	100.		250.	50.		3.	51.		50.	8.	
CAPACITY	1065.	1065.		1150.	1065.		141.	141.		141.	141.	
V/GCAP	.14953	.03455		.07992	.01712		.00249	.04211		.04167	.00687	

INTERSECTION 4

	PHASE 1			PHASE 2		PHASE 3		PHASE 4
TIME	75.			75.		25.		25.
V/GC	.11502			.06900		.02997		.06944
	NORTH-AP	SOUTH-AP	EAST-AP	WEST-AP	N-AP-LFT	S-AP-LFT	E-AP-LFT	W-AP-LFT
DELA	45.	40.	42.	40.	83.	79.	101.	78.
QUEUE	0.	0.	0.	0.	0.	0.	1.	0.
VOLUME	333.	70.	215.	38.	36.	17.	83.	12.
CAPACITY	1065.	1102.	1143.	1217.	141.	141.	141.	141.
V/GCAP	.11502	.02330	.06900	.01157	.02997	.01389	.06944	.00973

INTERSECTION 5

	PHASE 1			PHASE 2		PHASE 3		PHASE 4
TIME	75.			75.		25.		25.
V/GC	.09777			.05399		.04676		.06944
	NORTH-AP	SOUTH-AP	EAST-AP	WEST-AP	N-AP-LFT	S-AP-LFT	E-AP-LFT	W-AP-LFT
DELA	44.	41.	42.	40.	79.	89.	101.	78.
QUEUE	0.	0.	0.	0.	0.	0.	1.	0.
VOLUME	283.	136.	167.	80.	17.	56.	83.	11.
CAPACITY	1065.	1065.	1140.	1095.	141.	141.	141.	141.
V/GCAP	.09777	.04690	.05399	.02669	.01389	.04676	.06944	.00890

INTERSECTION 6

	PHASE 1			PHASE 2		PHASE 3		PHASE 4
TIME	75.			75.		25.		25.
V/GC	.14547			.07261		.02841		.05556
	NORTH-AP	SOUTH-AP	EAST-AP	WEST-AP	N-AP-LFT	S-AP-LFT	E-AP-LFT	W-AP-LFT
DELA	47.	40.	42.	40.	77.	83.	93.	92.
QUEUE	0.	0.	0.	0.	0.	0.	0.	0.
VOLUME	435.	77.	233.	80.	4.	34.	67.	65.
CAPACITY	1100.	1217.	1181.	1065.	141.	141.	141.	141.
V/GCAP	.14547	.02320	.07261	.02777	.00372	.02841	.05556	.05412

INTERSECTION 7

	PHASE 1			PHASE 2		PHASE 3		PHASE 4
TIME	75.			75.		25.		25.
V/GC	.08734			.06621		.02778		.12500
	NORTH-AP	SOUTH-AP	EAST-AP	WEST-AP	N-AP-LFT	S-AP-LFT	E-AP-LFT	W-AP-LFT
DELA	43.	40.	42.	40.	83.	83.	652.	78.
QUEUE	0.	0.	0.	0.	0.	0.	18.	0.
VOLUME	267.	64.	219.	42.	33.	33.	150.	8.
CAPACITY	1122.	1065.	1213.	1185.	141.	141.	141.	141.
V/GCAP	.08734	.02193	.06621	.01305	.02778	.02778	.12500	.00659

INTERSECTION 8

	PHASE 1			PHASE 2		PHASE 3		PHASE 4
TIME	75.			75.		25.		25.
V/GC	.09777			.07045		.02931		.08333
	NORTH-AP	SOUTH-AP	EAST-AP	WEST-AP	N-AP-LFT	S-AP-LFT	E-AP-LFT	W-AP-LFT
DELA	44.	40.	42.	40.	79.	83.	115.	78.
QUEUE	0.	0.	0.	0.	0.	0.	1.	0.
VOLUME	283.	69.	233.	85.	17.	35.	100.	11.
CAPACITY	1065.	1065.	1217.	1117.	141.	141.	141.	141.
V/GCAP	.09777	.02368	.07045	.02784	.01389	.02931	.08333	.00955

INTERSECTION 9

	PHASE 1	PHASE 2	PHASE 3	PHASE 4
TIME	75.	75.	25.	25.

	NORTH-AP	SOUTH-AP	EAST-AP	WEST-AP	N-AP-LFT	S-AP-LFT	E-AP-LFT	W-AP-LFT
V/C		.10352		.05032		.02778		.11111
DELA	44.	40.	41.	41.	77.	83.	250.	79.
QUEUE	0.	0.	0.	0.	0.	0.	6.	0.
VOLUME	300.	60.	167.	98.	0.	33.	133.	17.
CAPACITY	1065.	1065.	1217.	1065.	141.	141.	141.	141.
V/CAP	.10352	.02077	.05032	.03378	.00000	.02778	.11111	.01402

LINK	TRAVEL TIMES	LINK	TRAVEL TIMES	LINK	TRAVEL TIMES	LINK	TRAVEL TIMES	LINK	TRAVEL TIMES	LINK	TRAVEL TIMES
1	57. 57. 94.	2	17. 17. 17.	3	57. 57. 105.	4	61. 61. 268.	5	57. 57. 95.	6	65. 65. 165.
7	17. 17. 17.	8	61. 61. 105.	9	57. 57. 96.	10	17. 17. 17.	11	58. 58. 96.	12	61. 61. 119.
13	58. 58. 110.	14	60. 60. 119.	15	17. 17. 17.	16	61. 61. 111.	17	57. 57. 95.	18	17. 17. 17.
19	58. 58. 96.	20	61. 61. 671.	21	58. 58. 97.	22	61. 61. 133.	23	17. 17. 17.	24	60. 60. 268.
25	57. 57. 94.	26	17. 17. 17.	27	58. 58. 97.	28	62. 62. 101.	29	58. 58. 100.	30	64. 64. 102.
31	17. 17. 17.	32	62. 62. 101.	33	57. 57. 94.	34	17. 17. 17.	35	59. 59. 107.	36	62. 62. 98.
37	58. 58. 101.	38	62. 62. 98.	39	17. 17. 17.	40	62. 62. 98.	41	58. 58. 105.	42	17. 17. 17.
43	58. 58. 101.	44	66. 66. 96.	45	58. 58. 100.	46	66. 66. 96.	47	17. 17. 17.	48	63. 63. 95.

TOTAL TRAVEL TIME ON NETWORK 189.(VEH-HRS)
TOTAL RUNNING TIME IN PARKING ZONES 0.(VEH-HRS)
TOTAL VEHICLE MILES TRAVELED ON NETWORK 1465.(VEH-MI)

TOTAL INTERSECTION DELAY ON NETWORK 150.(VEH-HRS)
TOTAL STOPS AT INTERSECTIONS 5642.(VEH)
TOTAL OF INTERSECTION AVERAGE QUEUE LENGTHS 1748.(FT)

EMISSIONS SUBMODEL

REGION = 1
YEAR= 77
TEMP,COLD,HOT,CCOLD= 50.000 .200 .200 .200
VTMIX= .750 .050 .050 .090 .060 .000

EMISSION CORRECTION

1.262603	1.365603	1.422399	1.440174	1.435166
1.421784	1.409968	1.404052	1.406090	1.443988
1.429426	1.447363	1.472840	1.487815	1.512240
1.547107	1.605758	1.715649	1.931629	2.043941

EMISSIONS

LINK	NSTOPS	QLENGTH	COUNT	SPEED	E AT SPEED	DISTANCE	E DECEL	DISTANCE	E IDLE	DISTANCE	E ACCEL	DISTANCE
1	31	.8564+01	.5000+02	.5740+02	.2203-03	.1014+03	.6212-03	.1004+03	.7550-01	.2616+01	.1708-02	.1004+03
2	0	.0000	.3000+03	.5438+02	.1363-02	.3048+03	.0000	.0000	.0000	.0000	.0000	.0000
3	73	.1188+02	.9532+02	.5685+02	.4220-03	.1042+03	.1364-02	.9850+02	.9915-01	.3622+01	.3926-02	.9850+02
4	287	.1382+03	.3651+03	.5359+02	.1674-02	.8763+02	.5361-02	.8753+02	.3839-01	.4211+02	.1559-01	.8753+02
5	38	.9045+01	.5784+02	.5730+02	.2551-03	.1019+03	.7463-03	.1001+03	.8212-01	.2757+01	.2079-02	.1001+03
6	375	.1082+03	.4839+03	.5215+02	.2231-02	.1060+03	.7028-02	.8290+02	.6968-01	.3298+02	.2044-01	.8290+02
7	0	.0000	.5000+02	.5740+02	.2203-03	.3048+03	.0000	.0000	.0000	.0000	.0000	.0000
8	214	.5043+02	.3000+03	.5438+02	.1363-02	.1092+03	.4090-02	.9012+02	.8471-01	.1537+02	.1163-01	.9012+02
9	34	.7232+01	.5000+02	.5740+02	.2203-03	.1018+03	.6600-03	.1004+03	.8726-01	.2204+01	.1852-02	.1004+03
10	0	.0000	.3000+03	.5438+02	.1363-02	.3048+03	.0000	.0000	.0000	.0000	.0000	.0000
11	60	.1450+02	.9022+02	.5691+02	.3992-03	.1029+03	.1174-02	.9872+02	.8108-01	.4420+01	.3279-02	.9872+02
12	222	.5010+02	.2980+03	.5440+02	.1353-02	.1140+03	.4182-02	.9020+02	.8255-01	.1527+02	.1200-01	.9020+02
13	111	.2049+02	.1454+03	.5624+02	.6475-03	.1057+03	.2076-02	.9642+02	.9011-01	.6245+01	.5974-02	.9642+02
14	188	.4094+02	.2508+03	.5497+02	.1131-02	.1081+03	.3535-02	.9210+02	.8246-01	.1248+02	.1015-01	.9210+02
15	0	.0000	.5000+02	.5740+02	.2203-03	.3048+03	.0000	.0000	.0000	.0000	.0000	.0000
16	218	.4952+02	.3000+03	.5438+02	.1363-02	.1095+03	.4141-02	.9012+02	.8494-01	.1509+02	.1182-01	.9012+02
17	32	.7663+01	.5000+02	.5740+02	.2203-03	.1016+03	.6342-03	.1004+03	.8372-01	.2336+01	.1756-02	.1004+03
18	0	.0000	.3000+03	.5438+02	.1363-02	.3048+03	.0000	.0000	.0000	.0000	.0000	.0000
19	64	.1545+02	.9609+02	.5684+02	.4255-03	.1032+03	.1251-02	.9847+02	.8124-01	.4708+01	.3497-02	.9847+02
20	295	.3038+03	.3685+03	.5355+02	.1690-02	.3740+02	.5479-02	.8740+02	.1865-01	.9261+02	.1599-01	.8740+02
21	77	.1826+02	.1147+03	.5661+02	.5090-03	.1038+03	.1501-02	.9769+02	.8277-01	.5567+01	.4204-02	.9769+02
22	251	.6093+02	.3333+03	.5397+02	.1521-02	.1087+03	.4753-02	.8879+02	.7640-01	.1857+02	.1369-01	.8879+02
23	0	.0000	.5000+02	.5740+02	.2203-03	.3048+03	.0000	.0000	.0000	.0000	.0000	.0000
24	240	.1251+03	.3000+03	.5438+02	.1363-02	.8644+02	.4421-02	.9012+02	.3305-01	.3812+02	.1288-01	.9012+02
25	31	.8635+01	.5000+02	.5740+02	.2203-03	.1013+03	.6212-03	.1004+03	.7525-01	.2632+01	.1708-02	.1004+03
26	0	.0000	.3000+03	.5438+02	.1363-02	.3048+03	.0000	.0000	.0000	.0000	.0000	.0000
27	58	.1315+02	.8649+02	.5695+02	.3826-03	.1030+03	.1131-02	.9887+02	.8509-01	.4008+01	.3166-02	.9887+02
28	225	.5615+02	.3167+03	.5417+02	.1442-02	.1088+03	.4342-02	.8945+02	.8292-01	.1712+02	.1235-01	.8945+02
29	69	.1337+02	.9689+02	.5683+02	.4291-03	.1038+03	.1319-02	.9844+02	.9226-01	.4077+01	.3741-02	.9844+02
30	267	.6729+02	.3693+03	.5354+02	.1694-02	.1096+03	.5123-02	.8736+02	.8322-01	.2051+02	.1464-01	.8736+02
31	0	.0000	.5000+02	.5740+02	.2203-03	.3048+03	.0000	.0000	.0000	.0000	.0000	.0000
32	211	.5222+02	.3000+03	.5438+02	.1363-02	.1086+03	.4051-02	.9012+02	.8305-01	.1592+02	.1149-01	.9012+02
33	31	.8403+01	.5000+02	.5740+02	.2203-03	.1014+03	.6212-03	.1004+03	.7725-01	.2561+01	.1708-02	.1004+03
34	0	.0000	.3000+03	.5438+02	.1363-02	.3048+03	.0000	.0000	.0000	.0000	.0000	.0000
35	140	.2926+02	.1920+03	.5568+02	.8599-03	.1069+03	.2656-02	.9450+02	.8726-01	.8919+01	.7577-02	.9450+02
36	200	.5149+02	.2858+03	.5455+02	.1296-02	.1077+03	.3847-02	.9069+02	.8087-01	.1569+02	.1089-01	.9069+02
37	74	.1445+02	.1038+03	.5675+02	.4600-03	.1041+03	.1414-02	.9815+02	.9177-01	.4404+01	.4011-02	.9815+02
38	210	.5484+02	.3000+03	.5438+02	.1363-02	.1078+03	.4039-02	.9012+02	.8056-01	.1671+02	.1144-01	.9012+02

```
39    0   .0000      .5000+02   .5740+02   .2203-03   .3048+03   .0000      .0000      .0000      .0000      .0000      .0000
40  210   .5464+02   .3000+03   .5438+02   .1363-02   .1078+03   .4039-02   .9012+02   .8056-01   .1671+02   .1144-01   .9012+02
41  110   .2165+02   .1507+03   .5618+02   .6713-03   .1058+03   .2086-02   .9620+02   .8951-01   .6659+01   .5950-02   .9620+02
42    0   .0000      .3000+03   .5438+02   .1363-02   .3048+03   .0000      .0000      .0000      .0000      .0000      .0000
43   79   .1570+02   .1109+03   .5666+02   .4921-03   .1043+03   .1511-02   .9785+02   .9028-01   .4786+01   .4283-02   .9785+02
44  320   .8824+02   .4363+03   .5273+02   .1997-02   .1084+03   .6105-02   .8474+02   .7999-01   .2689+02   .1753-01   .8474+02
45   67   .1279+02   .9352+02   .5687+02   .4140-03   .1037+03   .1279-02   .9858+02   .9289-01   .3894+01   .3630-02   .9858+02
46  321   .8808+02   .4398+03   .5269+02   .2014-02   .1087+03   .6134-02   .8460+02   .8025-01   .2685+02   .1760-01   .8460+02
47    0   .0000      .5000+02   .5740+02   .2203-03   .3048+03   .0000      .0000      .0000      .0000      .0000      .0000
48  209   .5741+02   .3000+03   .5438+02   .1363-02   .1071+03   .4026-02   .9012+02   .7779-01   .1750+02   .1139-01   .9012+02

SUBROUTINE STREET CALLED FOR RECEPTOR     5 AND LINK    11
M=     5 I=    11 W=      18.29 BHT=      21.34 RHT=       1.52 XLO=       2.00
M=     5 I=    11 DXT=    304.80 DYT=        .00 SL=     304.80
M=     5 I=    11 II=     1 DX(II)=     102.95 DY(II)=        .00
M=     5 I=    11 II=     2 DX(II)=      98.72 DY(II)=        .00
M=     5 I=    11 II=     3 DX(II)=       4.42 DY(II)=        .00
M=     5 I=    11 II=     4 DX(II)=      98.72 DY(II)=        .00
JJ=     1 M=     5 XX(JJ)=      .387+03
YY(JJ)=      .643+03 XO(M)=      .610+03 YO(M)=      .637+03
DTS=      .223+03
JJ=     2 M=     5 XX(JJ)=      .488+03
YY(JJ)=      .643+03 XO(M)=      .610+03 YO(M)=      .637+03
DTS=      .122+03
JJ=     3 M=     5 XX(JJ)=      .539+03
YY(JJ)=      .643+03 XO(M)=      .610+03 YO(M)=      .637+03
DTS=      .707+02
JJ=     4 M=     5 XX(JJ)=      .591+03
YY(JJ)=      .643+03 XO(M)=      .610+03 YO(M)=      .637+03
DTS=      .198+02
I=    11 JKEEP=     4 Q(I,JKEEP)=      .328-02
M=     5 I=    11 JKEEP=     4 CW=      .379-03 CL=      .663-03
M=     5 I=    11 EPSIL=      30.00

SUBROUTINE STREET CALLED FOR RECEPTOR     5 AND LINK    12
M=     5 I=    12 W=      18.29 BHT=      21.34 RHT=       1.52 XLO=       2.00
M=     5 I=    12 DXT=   -304.80 DYT=     -54.90 SL=     309.70
M=     5 I=    12 II=     1 DX(II)=    -112.22 DY(II)=     -20.21
M=     5 I=    12 II=     2 DX(II)=     -88.77 DY(II)=     -15.99
M=     5 I=    12 II=     3 DX(II)=     -15.03 DY(II)=      -2.71
M=     5 I=    12 II=     4 DX(II)=     -88.77 DY(II)=     -15.99
JJ=     1 M=     5 XX(JJ)=      .584+03
YY(JJ)=      .694+03 XO(M)=      .610+03 YO(M)=      .637+03
DTS=      .625+02
JJ=     2 M=     5 XX(JJ)=      .483+03
YY(JJ)=      .676+03 XO(M)=      .610+03 YO(M)=      .637+03
DTS=      .132+03
JJ=     3 M=     5 XX(JJ)=      .432+03
YY(JJ)=      .667+03 XO(M)=      .610+03 YO(M)=      .637+03
DTS=      .180+03
JJ=     4 M=     5 XX(JJ)=      .380+03
YY(JJ)=      .657+03 XO(M)=      .610+03 YO(M)=      .637+03
DTS=      .231+03
I=    12 JKEEP=     1 Q(I,JKEEP)=      .135-02
M=     5 I=    12 JKEEP=     1 CW=      .156-03 CL=      .273-03
M=     5 I=    12 EPSIL=      30.00
```

INDIRECT SOURCE MODEL OF AIR POLLUTION

EXAMPLE HOT SPOT CALCULATION WITH MODIFIED ISMAP DATE = 770301

NO. OF LINKS = 48 NO. OF ZONES = 12 NO. OF INTERSECTIONS = 9

HOUR = 8. WIND DIRECTION = 120. WIND SPEED = 2.6+00 STABILITY = 4

RECEPTOR	CO CONCENTRATION (PPM)								
1 - 9	1.63-01	1.84-01	3.22-01	5.54-01	1.28+00	3.06-01	5.39-01	5.22-01	2.70-01

NORMAL EXIT. EXECUTION TIME: 16466 MILLISECONDS.

$PRKPT PRINT$

need not add to 100 since they are renormalized within the program. In our example the total trips attracted to and generated by all interior zones are distributed equally among the interior zones. The same procedure was followed in the case of exterior zones. According to this configuration each interior zone will attract 300 trips and generate 50 while each exterior zone will generate 300 trips and attract 50. Again, it should be noted that the terms interior and exterior no longer necessarily refer to the relative locations of zones but simply to define two groups of zones between which travel can occur. Card type 5 contains site description information and vehicle population characteristics while the card 6 entries specify the links comprising the gates entering the six interior zones. Card 7 gives the average number of passengers per vehicle for three vehicle types: automobiles, buses and trucks. No card type 8 is shown in the input listing since all trips are assumed to be generated. The location of each of the nine receptors is given on a separate card 9. The fifth receptor is designated as a street canyon receptor with the appropriate street and building characteristics. On card 10 a single hour's meteorological data is specified.

The first portion of the program output is simply a listing of the model input data with the receptor coordinates converted from feet to meters. The next part of the output is the parking zone capacities and parking zone trip lengths which are not used in the modified ISMAP calculations. Results of the traffic calculations are presented for each of the three iterations on an intersection by intersection and phase by phase basis. For each phase the volume to green time capacity ratio, average delay (sec), average queue length (vehicles), volume (vehicles/hour), capacity (vehicles/hour) and controlling volume to hour of green capacity ratio. Before the listing of intersection and phase parameters for the final iteration, a listing of vehicle counts on each link are printed out for vehicles going straight (thru), turning right (rt), turning left (lt) or terminating at a zone (term). Following a listing of parking lot parameters which are not used by the modified version of the program, the link travel times for going straight, turning right and turning left are printed out followed by listing of overall network travel times, delays and average queue lengths. Emission rates (g/m/sec) for each link are then printed out for different travel modes as follows:

LINK - Link number.

NSTOPS - Number of vehicles that stop on this link during the simulated time period.

QLENGTH - Average queue length experienced by all stopping vehicles, starting at the downstream intersection, ft.

COUNT - Number of vehicles using the link.

SPEED - Average speed of vehicles when unaffected by starting or stopping movements at the intersection, ft/sec.

E AT SPEED - Emission rate of vehicles traveling at SPEED, in gm sec^{-1} m^{-1}.

DISTANCE - Distance over which the E AT SPEED emission rate applies, m.

E DECEL - Emission rate of vehicles while decelerating from SPEED to a stopped position, gm sec^{-1} m^{-1}.

DISTANCE - Distance over which the E DECEL rate applies, m.

E IDLE - Emission rate of idling vehicles, gm sec^{-1} m^{-1}.

DISTANCE - Distance over which the E IDLE rate applies, m.

E ACCEL - Emission rate of accelerating vehicles, gm sec^{-1} m^{-1}.

DISTANCE - Distance over which the E ACCEL rate applies, m.

Finally the hour of the day and meteorological data are entered.

The output from the program consists of (1) a display of the input parameters; (2) a listing of the parking zone capacities and the average trip lengths within each zone; (3) traffic computations for each intersection in the network; (4) a summary of the travel times on each link; (5) a summary tabulation for the entire network showing total network travel time, total travel time in parking zones (always 0), network vehicle-miles of travel, total intersection delay, number of vehicle stops occurring, and the total of the average queue length; (6) link emission data summary; and (7) a summary showing the number of links, zones and intersections considered, the meteorological parameters used, and the resulting CO concentrations for each receptor.

Various codes used in the output are defined below:

TIME is the computed signal phase time (sec).

V/GV is the ratio of the highest approach volume to the approach capacity per hour of green signal time, at an intersection.

NORTH-AP, SOUTH-AP, EAST-AP, WEST-AP are the designations of the intersection's north, south, east and west approaches, respectively.

N-AP-LFT, S-AP-LFT, E-AP-LFT, W-AP-LFT are the designations of the left-turn phases from the intersection's north, south, east, and west approaches, respectively.

DELA is the computed delay time.

QUEUE is the average number of vehicles queued per cycle.

VOLUME is the traffic volume in vehicles per hour.

CAPACITY is the number of vehicles that could pass over a link taking into account interruptions to flow from traffic signals, pedestrians, slow moving vehicles, etc.

V/GCAP is the ratio of the approach volume to the approach capacity per hour of green signal time, for a specific intersection approach.

SECTION 6

USE OF THE MODIFIED ISMAP MODEL IN CONJUNCTION WITH APRAC

Since the modified ISMAP is not designed to handle a large number of traffic links, the concentrations calculated for the model receptor locations will certainly not reflect the carbon monoxide contributions from the remainder of the urban area. If application of the model shows that concentrations close to the standard might be expected, then some means must be found to include the remaining links of the traffic network in the analysis. For those receptors located quite close to roadways or intersections the contribution of other link emissions to the carbon monoxide concentration is often small in comparison to the contribution of the roadway or intersection in question. On the other hand, for those receptors not immediately adjacent to the roadway or intersection, this contribution can become a much greater fraction of the total concentration especially in a large urban area and when a low mixing prevails for an extended period of time.

One method to obtain quantitative estimates of the contributions of links not included in the modified ISMAP is by use of the APRAC-1A[8,10,11,12] or APRAC-2[15] computer models, which are designed for the prediction of hourly CO concentrations at specific receptor locations based upon traffic link emissions and meteorological data. The concentration contribution at a particular receptor location from links other than the receptor street is calculated by means of a receptor oriented Gaussian plume model. For a given hour the model sets up a series of area sources upwind of the receptor point. Each of these area sources is assigned an emission strength based upon the link elements contained within each. The vertical dispersion of the CO plume which occurs between the area source and receptor will depend upon a vertical dispersion parameter which is in turn a function of windspeed, cloud cover, and time of day. The dilution factor of CO along the axis of the plume is proportional to the windspeed.

For the most part, the differences between APRAC-1A and APRAC-2 are insignificant when the models are used to estimate the intra-urban background or mesoscale component. Most users will find APRAC-1A preferable, in that it costs one-quarter to one-half as much as APRAC-2 to execute. The principal differences between the models and their importance in the application considered here are reviewed below:

- Emissions Calculation - In the APRAC-1A program emissions, E, are parameterized simply in the form of an emission factor power law as:

$$E = \alpha S^{-\beta}$$

where E = emission factor (gm/veh/mi)

α, β = constants for a particular year, model year distribution, and weighted annual travel distribution

S = average traffic speed (mi/hr)

The speed S is available for the emission factor calculation since each link in the data set is characterized by one of eight speed classes.

This formulation, compared to APRAC-2 is very inexpensive in terms of computer time. One minor complication is that the α and β constants must be calculated for the emission characteristics of the vehicle populations. This implies that only one value of cold starts and the other emission parameters are applied universally. A different year or time period would require new constants.

APRAC-2, on the other hand, explicitly calculates the emission factors according to Supplement 5 of AP-42. Different cold starts proportions can be applied to seperate locations in a region and different time periods. A unique speed on each link can be input or speed can be calculated using a capacity restraint model. Consequently, APRAC-2 can calculate emissions with much greater sensitivity. The effect of such an enhancement on the calculation of the mesoscale component is unknown. (ISMAP, of course, is sensitive to these parameters and they are considered in the calculations of the microscale component.) APRAC-2 can be updated to reflect the new motor vehicle emission factors.[16]

- Multiple Wind Field - APRAC-2 can accommodate wind measurements from multiple locations in the region; unique values of speed and directions are then interpolated for each reciptor. Where such data are available and show significant variation, the use of APRAC-2 should be considered.

- Local Source Models - APRAC-1A and APRAC-2 provide for the use of a street canyon sub-model; APRAC-2 also provides for the use of a intersection sub-model. Since ISMAP already treats these situations, these options should not be used again when APRAC is used to calculate the mesoscale component.

- Dispersion Model Options - The following options are available to the user of the APRAC-1A model:

  - Synoptic Calculation - Source receptor relationships are recalculated for each new hour of meteorological data. This option should be used if the program is being run to simulate only a limited number of hours (e.g., 24 hours).

  - Climatological Calculation - Source receptor relationships are calculated at the beginning of the program for a wide range of meteorological configurations and written on a direct access device. The appropriate normalized concentrations for each source-receptor pair are then reread for the meteorological configuration which prevails at a given hour. The climatological option should be used for the simulation of a large number of hours. The final decision as to whether the synoptic or climatological options should be used must also depend upon the relative cost of CPU time and direct access I/O at the facility where the model is being run.

  - Grid Point Calculation - Carbon monoxide concentrations are calculated at up to 625 receptor locations for a specified hour of the day. This option would not be especially useful in conjunction with the application of the IMM since the user would be more interested in the concentration at a small number of receptors for more than a single hour.

  The APRAC-2 model has retained these options; however, the choice is implicit depending on the input data supplied to APRAC-2. APRAC-2 limits a user to 10 receptors for a 24-hour execution and one receptor for a multiday execution.

- Traffic Volume Data - The basic traffic input for the APRAC-1A model is the number of vehicles per day for an array of links whose endpoint coordinates are specified on a rectangular system in units of 0.01 miles. To account for the curvature of a particular link, the actual length of the roadway between the endpoint is input. Each link is also assigned an appropriate speed classification. Nonlink traffic volumes may be assigned to rid squares superimposed upon the traffic network. The daily traffic flow volume is apportioned for each hour of the day according to hourly traffic flow distributions specific to weekdays, Saturdays, and Sundays. There is also the option in the program of selecting a given hour of the day as a peak hour.

APRAC-2 treats link based traffic data in a similar manner. In addition, APRAC-2 has an option for accepting FHWA type binary link file via a preprocessor. Nonlink traffic volume can be assigned the same way as APRAC-1A or can automatically be computed by the mode. Daily traffic flow is apportioned for each hour according to diurnal distributions as in APRAC-1A. However, instead of diurnal factors for five facility types as in APRAC-1A, APRAC 2 requires diurnal distributions for two facility types, for two directions, for 5 locals (i.e., CBD, commercial, residential, industrial and rural) - a total of 20. There is no peak hour factor.

Appendix A

PROGRAM LISTING

```
C          DATA SET PTISMAP7   AT LEVEL 016 AS OF 07/13/78                MAI00010
C     ***** PTISMAP  INDIRECT SOURCE MODEL FOR AIR POLLUTION *****        MAI00020
C                                                                         MAI00030
C     MODIFIED ISMAP VERSION 2                                            MAI00040
C                                                                         MAI00050
C     PREPARED BY GCA/TECHNOLOGY DIVISION                                 MAI00060
C     BURLINGTON ROAD, BEDFORD MASSACHUSETTS                              MAI00070
C     (617) 275-9000                                                      MAI00080
C                                                                         MAI00090
C                                                                         MAI00100
C     VERSION 2 PREPARED BY FRANK BENESH INCORPORATING THE                MAI00110
C     NEW MOTOR VEHICLE EMISSION FACTORS AND MODAL MODEL                  MAI00120
C     COEFFICIENTS AND DETERIORATION RATES                                MAI00130
C                                                                         MAI00140
C     VERSION 1 PREPARED BY MICHAEL T. MILLS INCORPORATING THE            MAI00150
C     STREET CANYON MODEL AND DISABLING THE PARKING SUBROUTINE            MAI00160
C                                                                         MAI00170
C     PREPARED UNDER CONTRACT WITH THE U.S. E.P.A.                        MAI00180
C     SOURCE RECEPTOR ANALYSIS BRANCH                                     MAI00190
C     RESEARCH TRIANGLE PARK, NORTH CAROLINA                              MAI00200
C     GEORGE SCHEVE, PROJECT OFFICER                                      MAI00210
C                                                                         MAI00220
C*****************************************************************       MAI00230
C                                                                         MAI00240
      COMMON / COMM/ I,Z,L,J, K, TOD, DOW, TP, TOTATT, TTPZ,TOTGEN,ITM    MAI00250
     1,NYEAR,LHEAD(18), TUNFAC                                            MAI00260
      INTEGER Z                                                           MAI00270
      COMMON / LINK/ NLINK, NLANE(200), X1(200), Y1(200), X2(200),Y2(200MAI00280
     1),LCAP(200),DIST(200), VFL(200), LCON(200,3)                        MAI00290
     2,NSTOPS(200)                                                        MAI00300
      COMMON / INTRST/ NINS, LIN(70,4),ITYPC (70), CYCL(70), PH(70,4),    MAI00310
     1 CI(70), GCAP(70,4), QUE(70,4)                                      MAI00320
      COMMON /ZONES/ NZONES, NZLINS(50), ZLINKS(12,50), ZATTR(50),        MAI00330
     1ZGENR (50), NEXT, ZAT(2), ZGT(2), ZVT(3,50), ZNAME(50)              MAI00340
      INTEGER ZLINKS,ZNAME                                                MAI00350
      COMMON / GATE/ NGATE, LGATE(2,10), VCR(10, 2,4)                     MAI00360
      COMMON/VOLUME/ COUNT(4,200)                                         MAI00370
      COMMON /PARKZ/ PZ,   PV(50),PLA(50), VZ(50), ZV(50),PD,PLS,PLBO     MAI00380
     1,  PNOS(50) , PLL(50)                                               MAI00390
      INTEGER PNOS                                                        MAI00400
```

```
      INTEGEP PZ,PZ1                                                    MAI00410
      COMMON /VEHTYP/ VT,VTP(3), VTE(3), VTMULT(2), SZVTP(3,2)          MAI00420
      COMMON/EXCLUD/NOPARK                                              MAI00430
      COMMON TT(200,3),DFLA(200,3),PRT(50),QUEUE(200)                   MAI00440
     1, EMISS(200,4)                                                    MAI00450
      DATA IST/0/                                                       MAI00460
      DO 12 II3=1,200                                                   MAI00470
      DO 12 II4=1,3                                                     MAI00480
      LCON(II3,II4)=0                                                   MAI00490
   12 TT(II3,II4)=0.0                                                   MAI00500
    1 DO 11 J1=1,4                                                      MAI00510
      DO 11 J2=1,200                                                    MAI00520
   11 COUNT(J1,J2)=0.                                                   MAI00530
      CALL INPT                                                         MAI00540
C  TEST FOR FIRST TIME THRU AND INITIALIZE IF TRUE                      MAI00550
 14   IF (IST .GE. 1) GO TO 3                                           MAI00560
      IST = 1                                                           MAI00570
      CALL ZAREA                                                        MAI00580
C                                                                       MAI00590
C  INITIALIZE VOLUME TO CAPACITY RATIO INVERSES                         MAI00600
      DO 2   IG = 1,NGATE                                               MAI00610
      DO 2 K=1,2                                                        MAI00620
      DO 2   J  = 1,4                                                   MAI00630
 2    VCR(IG,K,J) = 2.0                                                 MAI00640
C                                                                       MAI00650
C  INITIALIZE LINK TO LINK TRAVEL TIMES                                 MAI00660
      DO 22  L  = 1,NLINK                                               MAI00670
      DIST(L)   = SQRT((X1(L)-X2(L))**2+(Y1(L)-Y2(L))**2)               MAI00680
      TTIME = DIST(L)/VEL(L)/NLANE(L)                                   MAI00690
      DO 22  J  = 1,3                                                   MAI00700
 22   TT (L,J) = TTIME                                                  MAI00710
C  INITIALIZE PARKING LOT LENGTH  (THE LENGTH OF THE LONGEST SIDE IS    MAI00720
C    STORED IN PLL(Z)                                                   MAI00730
      IZ = NEXT + 1                                                     MAI00740
      DO 24 Z= IZ,NZONES                                                MAI00750
      IF(PLL(Z) .LT. 0.0) PLL(Z) = -(PLL(Z) + PLA(Z)/PLL(Z) )           MAI00760
      PLL(Z) = AMAX1(PLL(Z),1.0)                                        MAI00770
 24   CONTINUE                                                          MAI00780
C                                                                       MAI00790
C  ROUTE VEHICLES SPECIFIED AS TOTALT (TOTAL TRIPS ATTRACTED AND GENERATMAI00800
```

```
C   ED) ONTO THE LINKS OF THE NETWORK                                    MAI00810
C                                                                        MAI00820
C   MODIFY ZONE ATTRACTIONS TO ACCOUNT FOR OVER CAPACITY DEMAND          MAI00830
      WRITE (6,28)                                                       MAI00840
   28 FORMAT('TRAFFIC SUBMODEL'/'0----------------'//)                   MAI00850
 3    N1 = NEXT +1                                                       MAI00860
      DO 33  Z = 1, NZONES                                               MAI00870
      IF (Z .GT. NEXT) GO TO 32                                          MAI00880
      VZ(Z) = ZATTR(Z)/ZAT(1)*TOTGEN                                     MAI00890
      ZV(Z) = ZGENR(Z)/ZGT(1)*TOTATT                                     MAI00900
      GO TO 325                                                          MAI00910
 32   VZ(Z) = ZATTR(Z)/ZAT(2) *TOTATT                                    MAI00920
      ZV(Z) = ZGENR(Z)/ZGT(2)*TOTGEN                                     MAI00930
 325  VTMULT(1) =(ZVT(2,Z) +ZVT(3,Z))/(ZVT(1,Z)+ZVT(2,Z)+ZVT(3,Z))      MAI00940
      VTMULT(2) = VTMULT(1) *ZV(Z)                                       MAI00950
      VTMULT(1) = VTMULT(1)*VZ(Z)                                        MAI00960
C STORE THE ZONE ATTRACTIONS AND GENERATION TEMPORARILY                  MAI00970
      QUEUE(Z) = VTMULT(1)                                               MAI00980
      QUEUE(Z+50) = VTMULT(2)                                            MAI00990
C     PRINT 935, VZ(Z),ZV(Z)                                             MAI01000
      VZ(Z) = VZ(Z) - VTMULT(1)                                          MAI01010
      ZV(Z) = ZV(Z) - VTMULT(2)                                          MAI01020
C     PRINT 935, VZ(Z),ZV(Z)                                             MAI01030
 33   CONTINUE                                                           MAI01040
      DO 352 J=1,NZONES                                                  MAI01050
      IZFLG = 0                                                          MAI01060
      DO 35  Z =N1, NZONES                                               MAI01070
C   TEST FOR OVER CAPACITY DEMAND                                        MAI01080
      IF ( VZ(Z) - ZV(Z) .LE. PLA(Z)/PD  - PV (Z)) GO TO 35              MAI01090
C   FIND          ZONE TO ROUTE VEHCLS WITH LARGEST ATTRACTION AND NOT   MAI01100
C   OVER CAPACITY                                                        MAI01110
      IZMAX = 0.0                                                        MAI01120
      ZMAX = 0.0                                                         MAI01130
      DO 34  IZ =N1, NZONES                                              MAI01140
      IF ( VZ(IZ)-ZV(IZ) .GE. PLA(IZ)/PD -PV(IZ)) GO TO 34               MAI01150
C    IDENTIFY THE ZONE ATTRACTING THE MOST TRIPS                         MAI01160
      IF(IZMAX .GE. VZ(IZ)) GO TO 331                                    MAI01170
      IZMAX = VZ(IZ)                                                     MAI01180
      IZ2 = IZ                                                           MAI01190
C   SEARCH FOR A COMMON LINK BETWEEN IZ AND Z.                           MAI01200
```

```
 331  DO 333 IZ3 = 1,12                                                MAI01210
      IF(ZLINKS(IZ3,IZ) .EQ. 0) GO TO 333                              MAI01220
      DO 332 IZ4 = 1,12                                                MAI01230
      IF(ZLINKS(IZ4,Z) .EQ. 0) GO TO 332                               MAI01240
      IF(IABS(ZLINKS(IZ3,IZ)).EQ. IABS(ZLINKS(IZ4,Z))) GO TO 335       MAI01250
 332  CONTINUE                                                         MAI01260
 333  CONTINUE                                                         MAI01270
      GO TO 34                                                         MAI01280
 335  IF( ZMAX .GT. PLA(IZ)/PD-PV(IZ)-(VZ(IZ)-ZV(IZ)))GO TO 34         MAI01290
      ZMAX = ( PLA(IZ) / PD - PV(IZ) - (VZ(IZ) - ZV(IZ) ))             MAI01300
      IZ1 = IZ                                                         MAI01310
  34  CONTINUE                                                         MAI01320
      IF(IZMAX .EQ. 0.0) GO TO 353                                     MAI01330
      IF(ZMAX .EQ. 0.0) IZ1 = IZ2                                      MAI01340
      VZC     =PLA(Z)/PD -PV(Z) + ZV(Z)                                MAI01350
      VZ (IZ1)= VZ(IZ1) + VZ(Z) - VZC                                  MAI01360
      VZ(Z) = VZC                                                      MAI01370
      IF(VZC .GT. 0.0) IZFLG = 1                                       MAI01380
 35   CONTINUE                                                         MAI01390
      IF(IZFLG .LE. 0) GO TO 353                                       MAI01400
 352  CONTINUE                                                         MAI01410
 353  CONTINUE                                                         MAI01420
C RESTORE BUS AND OTHER VEHICLES TO TRIP DEMANDS                       MAI01430
      DO 36 IZ = 1,NZONES                                              MAI01440
      VZ(IZ) = QUEUE(IZ) +VZ(IZ)                                       MAI01450
 36   ZV(IZ) = QUEUE(IZ+50) +ZV(IZ)                                    MAI01460
C     PRINT 935, (VZ(IZ),ZV(IZ),IZ = 1,NZONES)                         MAI01470
C935  FORMAT(* ISMAP35* 10F10.5)                                       MAI01480
C                                                                      MAI01490
C  ROUTE THE VEHICLES ONTO LINKS AND ADD TO COUNT ARRAY                MAI01500
 38   CALL ROUTEV                                                      MAI01510
      PRINT 938, (L,(COUNT(J,L),J=1,4),L=1,NLINK)                      MAI01520
 938  FORMAT (7H-COUNTS/5(26H LINK  THRU  RT  LT  TERM  )/5(I4,1X,4F5.0,MAI01530
     11X))                                                             MAI01540
C  CONVERT VZ AND ZV BACK TO AUTOMOBILE ATTRACTIONS AND GENERATIONS    MAI01550
      DO 39 IZ = 1, NZONES                                             MAI01560
       VTMULT(2) = ZVT(1,IZ)/(ZVT(1,IZ)+ ZVT(2,IZ)+ZVT(3,IZ)      )    MAI01570
      VZ(IZ) = VZ(IZ) *VTMULT(2)                                       MAI01580
 39   ZV(IZ) = ZV(IZ) * VTMULT(2)                                      MAI01590
      DO 40 L = 1, NLINK                                               MAI01600
```

```
      DO 395 K=1,3                                                      MAI01610
      TT(L,K)=0.0                                                       MAI01620
  395 DELA(L,K)=0.0                                                     MAI01630
 40   QUEUE(L) = 0.0                                                    MAI01640
C  FIND DELAY AND QUEUE LENGTHS AT INTERSECITONS                        MAI01650
      DO  45 I=1,NINS                                                   MAI01660
      IF (ITYPC(I) .GT. 0) GO TO 43                                     MAI01670
      CALL INSECU                                                       MAI01680
      GO TO 45                                                          MAI01690
 43   CALL INSEC                                                        MAI01700
 45   CONTINUE                                                          MAI01710
      DC 465 L=1,NLINK                                                  MAI01720
C  DETERMINE AVERAGE QUEUE LENGTH IN FEET ON EACH LINK AT 30 FT/VEH.    MAI01730
      QUEUE(L) = QUEUE(L)*30./NLANE(L)                                  MAI01740
      QUEUE(L)=AMIN1(QUEUE(L),DIST(L))                                  MAI01750
      IF(VEL(L) .LE. 0) GO TO 465                                       MAI01760
C   MODIFY VELOCITY VASED ON A SIMPLE LINK MODEL                        MAI01770
      SC = (COUNT(1,L)+COUNT(2,L)+COUNT(3,L))*3600/TP                   MAI01780
      CAPL = 1200.                                                      MAI01790
      IF(VEL(L) .GT. 70.0) CAPL = 1800.                                 MAI01800
      V = VEL(L)*(1.0-.5*SC/(NLANE(L)*CAPL))                            MAI01810
      DO 46 K=1,3                                                       MAI01820
 46   TT(L,K) = DIST(L)/V + DELA(L,K)                                   MAI01830
 465  CONTINUE                                                          MAI01840
 47   CONTINUE                                                          MAI01850
C                                                                       MAI01860
C  FIND THE RUNNING TIMES IN THE PARKING ZONES                          MAI01870
      PZ1 = NEXT +1                                                     MAI01880
      DC 57 PZ=PZ1,NZONES                                               MAI01890
      CALL PARKNG                                                       MAI01900
      PRT(PZ)=TTPZ                                                      MAI01910
      IF(NOPARK.EQ.1) PRT(PZ)=1.0E-08                                   MAI01920
   57 CONTINUE                                                          MAI01930
C                                                                       MAI01940
C  WRITE OUT RESULTS                                                    MAI01950
      WRITE (6, 961) (L,(TT(L,K),K=1,3),L =1,NLINK)                     MAI01960
 961  FORMAT (1H0, 6(20HLINK  TRAVEL TIMES  )/(1X,6(I3,2X,3F5.0    )))  MAI01970
      SUMTT=0.0                                                         MAI01980
      SVMT=0.0                                                          MAI01990
      SDELA=0.0                                                         MAI02000
```

```
      SNSTOP=0.0                                                        MAI02010
      SQI=0.0                                                           MAI02020
      SRT=0.0                                                           MAI02030
      DO 63 L=1,NLINK                                                   MAI02040
      DO 62 K=1,3                                                       MAI02050
      SUMTT = TT(L,K)*COUNT(K,L)/3600. + SUMTT                          MAI02060
      SVMT = COUNT(K,L)*DIST(L)/5280.0 + SVMT                           MAI02070
 62   SDELA = DELA(L,K)*COUNT(K,L)/3600.+SDELA                          MAI02080
      SNSTOP = NSTOPS(L) + SNSTOP                                       MAI02090
 63   SQI = QUEUE(L) + SQI                                              MAI02100
      DO 64 PZ = PZ1 , NZONES                                           MAI02110
 64   SRT = PRT(PZ)/3600. + SRT                                         MAI02120
      PRINT 962, SUMTT, SRT, SVMT                                       MAI02130
 962  FORMAT (29H-TOTAL TRAVEL TIME ON NETWORK, F21.0,9H(VEH-HRS) /     MAI02140
     1 36H TOTAL RUNNING TIME IN PARKING ZONES, F14.0, 9H(VEH-HRS)/     MAI02150
     2 40H TOTAL VEHICLE MILES TRAVELED ON NETWORK, F10.0, 8H(VEH-MI))  MAI02160
      PRINT 963, SDELA,SNSTOP,SQI                                       MAI02170
 963  FORMAT (36H0TOTAL INTERSECTION DELAY ON NETWORK,F14.0,9H(VEH-HRS)/MAI02180
     1 29H TOTAL STOPS AT INTERSECTIONS, F21.0, 5H(VEH)/                MAI02190
     2 44H TOTAL OF INTERSECTION AVERAGE QUEUE LENGTHS,F6.0,5H(FT) )    MAI02200
      IF (IST .GT. 1) GO TO 7                                           MAI02210
      CALL COORXY                                                       MAI02220
C                                                                       MAI02230
C  EMISSIONS MODEL                                                      MAI02240
C                                                                       MAI02250
 7    CALL EMIT                                                         MAI02260
C   ADD EMISSIONS FROM 1 LINK ZONES TO THE PPPROPRIATE LINK             MAI02270
      IST = IST + 1                                                     MAI02280
      DO 75 IZ = 1,NZONES                                               MAI02290
      IF(PNOS(IZ) .NE. 1) GO TO 75                                      MAI02300
      L = ZLINKS(1,IZ)                                                  MAI02310
      ADEMIS = PRT(IZ)/(DIST(L)*0.3048)                                 MAI02320
      DO 73 J=1,4                                                       MAI02330
 73   EMISS(L,J) = EMISS(L,J) + ADEMIS                                  MAI02340
 75   CONTINUE                                                          MAI02350
C  APPLY TUNNINE FACTOR TO ALL EMISSIONS                                MAI02360
      DO 77 L=1,NLINK                                                   MAI02370
      DO 77 J=1,4                                                       MAI02380
 77   EMISS(L,J) = EMISS(L,J) * TUNFAC                                  MAI02390
C                                                                       MAI02400
```

```
C   DISPERSION MODEL                                                    MAI02410
C                                                                       MAI02420
 8      CALL DISPER                                                     MAI02430
C                                                                       MAI02440
C       PRINT RESULTS                                                   MAI02450
        CALL PRNTOU                                                     MAI02460
        DO 88 PZ = PZ1,NZONES                                           MAI02470
 88     PV(PZ) = AMAX1(PV(PZ)+VZ(PZ) - ZV(PZ), 0.0)                     MAI02480
        GO TO 1                                                         MAI02490
        END                                                             MAI02500
```

```
      SUBROUTINE INPT                                                   INP00010
      COMMON / COMM/ I,Z,L,J, K, TOD, DOW, TP, TOTATT, TTPZ,TOTGEN,ITM  INP00020
     1,NYEAR,LHEAD(18), TUNFAC                                          INP00030
      INTEGER Z,VT                                                      INP00040
      COMMON /LINK/ NLINK, NLANE(200), X1(200), Y1(200), X2(200), Y2(200INP00050
     1), LCAP(200), DIST(200), VEL(200), LCON(200,3)                    INP00060
     2,NSTOPS(200)                                                      INP00070
      COMMON /INTRST/ NINS, LIN(70,4),ITYPC(70),ICYCL(70),PH(70,4),     INP00080
     1 CI(70), GCAP(70,4), QUE(70,4)                                    INP00090
      COMMON / ZONES/ NZONES, NZLINK(50), ZLINKS(12,50), ZATTR(50),     INP00100
     1ZGENR (50), NEXT, ZAT(2), ZGT(2), ZVT(3,50), ZNAME(50)            INP00110
      INTEGER ZNAME                                                     INP00120
      INTEGER ZLINKS                                                    INP00130
      COMMON / GATE/ NGATE, LGATE(2,10), VCR(10, 2,4)                   INP00140
      COMMON / VOLUME/ COUNT(4,200)                                     INP00150
      COMMON / PARKZ/ PZ, PV(50),PLA(50), VZ(50), ZV(50),PD ,PLS,PLBO   INP00160
     1, PNOS(50), PLL(50)                                               INP00170
      INTEGER PNOS                                                      INP00180
      COMMON /VEHTYP/ VT, VTP(3), VTE(3), VTMULT(2),SZVTP(3,2)          INP00190
      COMMON/DISP/XO(200),YO(200),NUMREC,THETA(24),WS(24),YCOEFF(24)    INP00200
     1,YEXP(24),YCONSL,YCONSZ,ZCOEFF(24),ZEXP(24),ZCONST,GRDSIZ,BCKG    INP00210
     2RD,MIX,ISTAB(24),SLAT,ICLD(24),NMET,NY,ITIME,HEIGHT(200)          INP00220
      COMMON/EXCLUD/NOPARK                                              INP00230
      COMMON/STCAN/ISTR(200),AST(200),NLKST(200),WST(200),              INP00240
     &BUILDH(200),RECHGT(200),ISTLIN(200,2),IRSIDE(200)                 INP00250
      COMMON/JUNK1/AC,XLOAD(3),TRAILR,ABSHUM                            INP00260
      COMMON/IMCOM/ICYIM,ISTRIN,IMTFLG,MODYR1,MODYR2                    INP00270
      COMMON/JUNK2/HGWGT,HDWGT,HGCID,HDCID                              INP00280
      COMMON/ELINK/IREG,TEMP,COLD,HOT,CCOLD,VMTMIX(6),                  INP00290
     & ALHFLG,TRKFLG,IMFLG                                              INP00300
      INTEGER ALHFLG,TRKFLG                                             INP00310
      READ(5,910,END=99)NYEAR,LHEAD,NOPARK                              INP00320
  910 FORMAT(I6,18A4,T80,I1)                                            INP00330
      PRINT 911,NYEAR,LHEAD,NOPARK                                      INP00340
  911 FORMAT(1H1,20X,I10,5X,18A4,I4)                                    INP00350
      IF(NOPARK.EQ.1)WRITE(6,915)                                       INP00360
  915 FORMAT(/,1X,'PARKING LOT EMISSIONS NOT CONSIDERED')               INP00370
      READ(5,912,END=99) IC,L,I,Z,ICN,IG,NV,KT,NR,NMET,TOD,DOW,TP,      INP00380
     1TOTATT,TOTGEN,PD,ITM                                              INP00390
      PRINT 912,IC,L,I,Z,ICN,IG,NV,KT,NR,NMET,TOD,DOW,TP,TOTATT,TOTGEN,PINP00400
```

```
     1D,ITM                                                            INP00410
 912  FORMAT(I2,I3,8I5,6F5.0,I5)                                        INP00420
      IF(NOPARK.EQ.1) PD=1.0E-9                                         INP00430
 2    IF(L .EQ. 0) GO TO 3                                              INP00440
      NLINK = L                                                         INP00450
      DO 25 L1=1,NLINK                                                  INP00460
      READ(5,922)  IC, L, (NLANE(L), X1(L),Y1(L),X2(L),Y2(L), LCAP(L)   INP00470
     1,VEL(L),(LCON(L,J),J=1,3),HEIGHT(L))                              INP00480
 25   PRINT 922,IC, L, (NLANE(L), X1(L),Y1(L),X2(L),Y2(L), LCAP(L),VEL  INP00490
     1(L),(LCON(L,J),J=1,3),HEIGHT(L))                                  INP00500
 922  FORMAT(I2,I3,I5,4F5.0,I5,F5.0,3I5,F5.0)                           INP00510
 3    IF(I .EQ. 0) GO TO 4                                              INP00520
      NINS = I                                                          INP00530
      READ(5,933) (IC,I,(LIN(I,J),J=1,4),ITYPC(I),ICYCL(I),(PH(I,K)     INP00540
     1, K=1,4),CI(I),(GCAP(I,M),M=1,4) ,I1=1,NINS)                      INP00550
      PRINT 933, (IC,I,( (LIN (I,J),J=1,4), ITYPC(I), ICYCL(I),(PH(I,K) INP00560
     1, K=1,4),CI(I),(GCAP(I,M),M=1,4)),I =1,NINS)                      INP00570
  933 FORMAT(I2,I3,6I5,9F4.0)                                           INP00580
 4    IF(Z .EQ. 0) GO TO 5                                              INP00590
      NZONES = Z                                                        INP00600
      Z=1                                                               INP00610
      IZ=NZONES                                                         INP00620
      DO 475 KZ = 1,NZONES                                              INP00630
      READ(5,944) (IC,IEXT, ZATTR(IZ),ZGENR(IZ),(ZVT(J,IZ),J=1,3),      INP00640
     1PVEH,PLC, NZL, (ZLINKS(K,IZ),K=1,12))                             INP00650
      IF(IC .NE. 4) CALL EXIT                                           INP00660
 944  FORMAT (I2,I2,2F3.0, 3F3.1, 2F5.0, I2, 12I4)                      INP00670
C  THIS TEST IS PECULIAR TO THE CDC 6400 COLLATING SEQUENCE.            INP00680
      IF (IEXT .GT. 49) GO TO 44                                        INP00690
C  STORE AN INTERNAL ZONES DATA                                         INP00700
      JZ = IZ                                                           INP00710
      IZ = IZ - 1                                                       INP00720
      GO TO 47                                                          INP00730
C  STORE AN EXTERNAL ZONES DATA                                         INP00740
 44   NEXT = Z                                                          INP00750
      ZATTR( Z) = ZATTR(IZ)                                             INP00760
      ZGENR( Z) = ZGENR(IZ)                                             INP00770
      ZVTSUM = 0.0                                                      INP00780
      DO 45 K= 1,12                                                     INP00790
 45   ZLINKS(K,Z) = ZLINKS(K, IZ)                                       INP00800
```

```
      DO 46 J = 1,3                                                     INP00810
46    ZVT(J, Z) = ZVT(J, IZ)                                            INP00820
      JZ = Z                                                            INP00830
      Z = Z+1                                                           INP00840
47    DO 472 K=1,12                                                     INP00850
      IF (ZLINKS(K,JZ) .NE. 0) GO TO 472                                INP00860
      NZLINK (JZ) = K-1                                                 INP00870
      GO TO 473                                                         INP00880
472   CONTINUE                                                          INP00890
      NZLINK(JZ) = 12                                                   INP00900
473   IF(PLC .NE. 0.0) PLA(JZ) = PLC*PD                                 INP00910
      IF(PVEH .NE. 0.0) PV(JZ) = PVEH                                   INP00920
      PNOS(JZ) = NZL                                                    INP00930
      ZNAME(JZ) = IEXT                                                  INP00940
      IF(ZVT(1,JZ)+ ZVT(2,JZ) + ZVT(3,JZ) .LE. 0.) ZVT(1,JZ) = 1.0      INP00950
475   CONTINUE                                                          INP00960
      ZAT(1)=0.0                                                        INP00970
      ZAT(2)=0.0                                                        INP00980
      ZGT(1)=0.0                                                        INP00990
      ZGT(2)=0.0                                                        INP01000
      DO 49 Z = 1,NZONES                                                INP01010
      IZ = Z                                                            INP01020
      KZ = NZLINK(IZ)                                                   INP01030
      IF (Z .GT. NEXT) GO TO 48                                         INP01040
      ZAT(1) = ZAT(1) + ZATTR(Z)                                        INP01050
      ZGT(1) = ZGT(1) + ZGENR(Z)                                        INP01060
      GO TO 49                                                          INP01070
48    ZAT(2) = ZAT(2) + ZATTR(Z)                                        INP01080
      ZGT(2) = ZGT(2) + ZGENR(Z)                                        INP01090
49    PRINT 944, IC, ZNAME(IZ), ZATTR(IZ), ZGENR(IZ), (ZVT(J,IZ),J=1,3) INP01100
     1, PV(IZ), PLA(IZ), PNOS(IZ)   , (ZLINKS(K,IZ),K=1,KZ)             INP01110
5     IF ( ICN .EQ. 0) GO TO 6                                          INP01120
      READ(5,955) IC,GRDSIZ,BCKGRD,SLAT,TUNFAC,ZCONST,IREG,TEMP,        INP01130
     1 COLD,HOT,CCOLD,(VMTMIX(J1),J1=1,6),ALHFLG,TRKFLG,IMFLG           INP01140
 955  FORMAT(I2,5F4.0,I2,F4.0,9F5.3,1X,3I1)                             INP01150
      IF(TUNFAC.EQ.0) TUNFAC=1                                          INP01160
      WRITE(6,955)IC,GRDSIZ,BCKGRD,SLAT,TUNFAC,ZCONST,IREG,TEMP,        INP01170
     1 COLD,HOT,CCOLD,(VMTMIX(J1),J1=1,6),ALHFLG,TRKFLG,IMFLG           INP01180
      IF(ALHFLG.EQ.1) READ(5,956)AC,(XLOAD(J1),J1=1,3),TRAILR,ABSHUM    INP01190
 956  FORMAT(10X,6F10.0)                                                INP01200
```

```
      IF(ALHFLG.EQ.1) WRITE(6,956)AC,(XLOAD(J1),J1=1,3),TRAILR,ABSHUM    INP01210
      IF(TRKFLG.EQ.1)READ(5,956)HGWGT,HDWGT,HGCID,HDCID                  INP01220
      IF(TRKFLG.EQ.1)WRITE(6,956)HGWGT,HDWGT,HGCID,HDCID                 INP01230
      IF(IMFLG.EQ.1) READ(5,957)ICYIM,ISTRIN,IMTFLG,MODYR1,MODYR2        INP01240
      IF(IMFLG.EQ.1)WRITE(6,957)ICYIM,ISTRIN,IMTFLG,MODYR1,MODYR2        INP01250
  957 FORMAT(10X,5I5)                                                    INP01260
  6   IF(IG.EQ.0) GO TO 7                                                INP01270
      NGATE = IG                                                         INP01280
      READ(5,966) (IC, (LGATE(J,IG),J =1,2),IG=1,NGATE )                 INP01290
      PRINT 966,(IC, (LGATE(J,IG),J =1,2),IG=1,NGATE )                   INP01300
 966  FORMAT ( I2, 3X, 2I5)                                              INP01310
 7    IF(NV .EQ. 0) GO TO 76                                             INP01320
      NVETP = NV                                                         INP01330
      READ(5,97) (IC, VT, VTP(VT), VTE(VT),K=1,NV)                       INP01340
 97   FORMAT (I2,I3, 2F5.3)                                              INP01350
 76   CONTINUE                                                           INP01360
 8    IF(KT.EQ. 0) GO TO 89                                              INP01370
      NCOUNT = KT                                                        INP01380
      DO 85 K=1,NCOUNT                                                   INP01390
      READ(5,988) IC,L,(COUNT(J1, L),J1=1,3), L2,(COUNT(J2,L2),J2=1,3),  INP01400
     1 L3, (COUNT(J3,L3), J3=1,3), L4,(COUNT(J4,L4), J4= 1,3)            INP01410
 85   PRINT 988,IC,L,(COUNT(J1,L),J1=1,3), L2,(COUNT(J2,L2),J2=1,3),     INP01420
     1 L3, (COUNT(J3,L3), J3=1,3), L4,(COUNT(J4,L4), J4= 1,3)            INP01430
 988  FORMAT ( I2,I3, 3(F5.0), 3(I5, 3(F5.0)))                           INP01440
 89   IF (NR .LE. 0) GO TO 8910                                          INP01450
      NUMREC = NR                                                        INP01460
      DO 90 I=1,NR                                                       INP01470
      DO 8010 J=1,2                                                      INP01480
      ISTLIN(I,J)=0                                                      INP01490
8010  CONTINUE                                                           INP01500
      READ(5,989)IC,XO(I),YO(I),ISTR(I),AST(I),NLDUM,WST(I),             INP01510
     &BUILDH(I),RECHGT(I),ISTLIN(I,1),ISTLIN(I,2),IRSIDE(I)              INP01520
989   FORMAT(I2,2F10.5,I3,F5.0,I5,2F10.0,F5.0,2I3,I4)                    INP01530
      NLKST(I)=NLDUM                                                     INP01540
      XO(I)=XO(I)*0.3048                                                 INP01550
      YO(I)=YO(I)*0.3048                                                 INP01560
      WST(I)=WST(I)*0.3048                                               INP01570
      BUILDH(I)=BUILDH(I)*0.3048                                         INP01580
   90 RECHGT(I)=RECHGT(I)*0.3048                                         INP01590
      IF (IC .NE. 9) PRINT 9391                                          INP01600
```

```
 9891 FORMAT(1X,' CARD TYPE 9 CHECK ERROR ')                          INP01610
      DO 9893 I=1,NR                                                  INP01620
      PRINT 9892 ,IC,XO(I),YO(I),ISTR(I),AST(I),NLKST(I),             INP01630
     &WST(I),BUILDH(I),RECHGT(I),ISTLIN(I,1),ISTLIN(I,2),IRSIDE(I)    INP01640
 9892 FORMAT(1X,I2,2F10.5,I3,F5.0,I5,2F10.0,F5.0,2I3,I4)              INP01650
 9893 CONTINUE                                                        INP01660
 8910 IF(NMET.EQ.0.AND.IMET.EQ.24) GO TO 98                           INP01670
      IF(NMET.EQ.0.AND.IMET.EQ.1) CALL EXIT                           INP01680
      IMET=NMET                                                       INP01690
      IF(NMET.EQ.1) GO TO 9901                                        INP01700
      DO 9989 I=1,24                                                  INP01710
      READ(5,9899)IC,THETA(I),WS(I),ICLD(I)                           INP01720
 9899 FORMAT(I2,2F8.2,I5,F7.2,2F10.2)                                 INP01730
 9989 WRITE(6,9897)IC,THETA(I),WS(I),ICLD(I)                          INP01740
 9897 FORMAT(1X,I2,2F8.2,I5,F7.2,2F10.2)                              INP01750
      GO TO 98                                                        INP01760
 9901 ITIME=TOD+1                                                     INP01770
      READ(5,9899)IC,THETA(ITIME),WS(ITIME),ISTAB(ITIME)              INP01780
      PRINT 9897, IC,THETA(ITIME),WS(ITIME),ISTAB(ITIME)              INP01790
 98   RETURN                                                          INP01800
 99   STOP                                                            INP01810
      END                                                             INP01820
```

```
      SUBROUTINE INSEC                                                  INS00010
C  THIS PROGRAM COMPUTES DELAY AND QUEUE LENGTH AT AN INTERSECTION.     INS00020
C  IT IS BASED ON GORDON NEWLLS INTERSECTION MODEL.  THREE TYPES OF     INS00030
C  SIGNALS MAY BE SPECIFIED 1)FIXED TIME 2)ACTUATED WITH LEFT THURN     INS00040
C  PHASES 3)ACTUATED WITH SEPEPATE PHASES FOR EACH APPROACH.  THIS MODELINS00050
C  WILL HANDLE OVER CAPACITY INTERSECTIONS, AN EXTENSION TO NEWELLS     INS00060
C  MODEL, BY ASSUMING CAPACITY FLOW WHEN THE QUEUE LENGTH AT THE END    INS00070
C  OF GREEN EQUALS HALF THE ARRIVALS ON RED PLUS THE ARRIVALS ON GREEN  INS00080
C  CYCLE AND PHASE                                                      INS00090
C  LENGTHS ARE COMPUTED BASED ON F.V.WEBSTER "TRAFFIC SIGNAL SETTINGS"  INS00100
      COMMON / COMM/ I,Z,L,J, K, TOD, DOW, TP, TOTATT, TTPZ,TOTGEN      INS00110
      COMMON /LINK/  NLINK, NLANE(200), X1(200), Y1(200),X2(200),Y2(200)INS00120
     1, LCAP(200), DIST(200), VEL(200), LCON (200,3)                    INS00130
     2,NSTOPS(200)                                                      INS00140
      COMMON /INTRST/NI, LIN(70,4),ITYPC(70),ICYCL(70), PH(70,4), CI(70)INS00150
     1 , GCAP(70,4),QUE(70,4)                                           INS00160
      COMMON /ZONES/    NZONES,NZLINK(50), ZLINKS(12,50), ZATTP(50),    INS00170
     1ZGENR (50), NEXT, ZAT(2), ZGT(2), ZVT(3,50), ZNAME(50)            INS00180
      COMMON / GATE/ NGATE, LGATE(2,10), VCR(10, 2,4)                   INS00190
      COMMON /VOLUME/ COUNT(4,200)                                      INS00200
      COMMON /PARKZ/ PZ,       PV(50), PLA(50), VZ(50), ZV(50)          INS00210
      REAL MINR                                                         INS00220
      COMMON TT(200,3), DELA(200,3), PRT(50), QUEUE(200)                INS00230
      COMMON                PT(4,3),VOL(4,3),Y(4,3),CAP(4,3),YT(4)      INS00240
     1,DELAI(70,8),QQ(4,3)                                              INS00250
      REAL*8 NPHASE(9)                                                  INS00260
      DATA NPHASE/ 8HNORTH-AP, 8H EAST-AP, 8HSOUTH-AP,                  INS00270
     1 8H WEST-AP, 8HN-AP-LFT, 8HS-AP-LFT, 8HE-AP-LFT,                  INS00280
     2 8HW-AP-LFT, 8H          /                                        INS00290
C  FOR EACH SIGNAL PHASE DETERMINE A PHASE LENTH                        INS00300
 1    NP = 0                                                            INS00310
      MINR = 1.0                                                        INS00320
      SY = 0                                                            INS00330
C   INITIALIZE VOLUMES ON EACH APPROACH                                 INS00340
      DO 15 J=1,4                                                       INS00350
      VOL(J,1) = 0.0                                                    INS00360
      L = LIN(I,J)                                                      INS00370
      IF(L.GT.0) VOL(J,1)=(COUNT(3,L)+COUNT(2,L)+COUNT(1,L))*3600./TP   INS00380
      VOL(J,2)=0.0                                                      INS00390
      VOL(J,3)=0.0                                                      INS00400
```

```
      CAP(J,2)=0.0                                                      INS00410
      CAP(J,3)=0.0                                                      INS00420
      DELAI(I,J)=0.0                                                    INS00430
      DELAI(I,J+4)=0.0                                                  INS00440
      QQ(J,1)=0.0                                                       INS00450
      QQ(J,2)=0.0                                                       INS00460
      QQ(J,3)=0.0                                                       INS00470
 15   CONTINUE                                                          INS00480
C  FIRST DETERMINE THE VOLUME AND CAPACITY OF EACH APPROACH.            INS00490
 2    J1 = 1                                                            INS00500
C   J IS INDEX OF APPROACH DIRECTION                                    INS00510
C  J2 IS INDEX OF PHASE CONTROLLING THIS APPROACH                       INS00520
      DO 25 J = 1,4                                                     INS00530
      CAP(J,1) = 0.5                                                    INS00540
      L = LIN(I,J)                                                      INS00550
      IF(L .EQ. 0) GO TO 24                                             INS00560
C  SET J1,AN INDEX TO TEST WHETHER THERE IS  MORE THAN ONE PHASE (OR    INS00570
C  SUB-PHASE CONTROLING THE N-S OR E-W APPROACHES                       INS00580
      J2 = J1                                                           INS00590
C   SET INDEX J3 TO IDENTIFY OPPOSING TRAFFIC                           INS00600
C      IE OPPOSING APPROACH                                             INS00610
      J3 = J1                                                           INS00620
      IF(J .LE. 2) J3 = J+2                                             INS00630
      IF(ITYPC(I) .EQ. 3) GO TO 21                                      INS00640
      IF(ITYPC(I) .EQ. 4 .AND. J1 .EQ. 2) GO TO 21                      INS00650
      IF(ITYPC(I) .EQ. 5 .AND. J1 .EQ. 1) GO TO 21                      INS00660
      CAP (J,1) = GCAP(I,J2)                                            INS00670
      IF (CAP(J,1) .EQ. 0.0) CAP(J,1) = NLANE (L)*1200.                 INS00680
      IF(VOL(J,1) .LE. 0. ) GO TO 24                                    INS00690
C  CONTROL IS BY A TWO PHASE SIGNAL WITH 2 POSSIBLE MINOR MOVEMENTS     INS00700
      IF( PH(I,J2+2) .NE. 0.) GO TO 22                                  INS00710
      IF(VOL(J3,1) .EQ. 0.0) GO TO 225                                  INS00720
C  MODIFY THE CAPACITIES DUE TO LEFT TURNING VEHICLES (SEE TABLE6.5 HCM)INS00730
 200  PCNTRN = AMIN1(COUNT(3,L)/VOL(J,1)*3600./TP, 0.3)                INS00740
      BASEPT = 0.1                                                      INS00750
      DIF = 0.5                                                         INS00760
      BPVAL = 0.0                                                       INS00770
      IF(NLANE(L) .GT. 2) GO TO 208                                     INS00780
      IF(NLANE(L) .EQ. 2) GO TO 205                                     INS00790
      IF(PCNTRN  .LT. 0.2) GO TO 201                                    INS00800
```

```
      BASEPT = 0.2                                                      INS00810
      BPVAL = 0.15                                                      INS00820
      GO TO 208                                                         INS00830
 201   IF(PCNTRN .LT. 0.15) GO TO 203                                   INS00840
      BASEPT = 0.15                                                     INS00850
      BPVAL = 0.1                                                       INS00860
      GO TO 207                                                         INS00870
 203  DIF = 2.0                                                         INS00880
      IF(PCNTRN .LT. 0.1) DIF = 3.0                                     INS00890
      GO TO 208                                                         INS00900
 205   IF(PCNTRN .LT. 0.2) GO TO 207                                    INS00910
      BASEPT = 0.2                                                      INS00920
      BPVAL = 0.1                                                       INS00930
      GO TO 208                                                         INS00940
 207  DIF = 1.0                                                         INS00950
 208  FACL = DIF*(BASEPT-PCNTRN)-BPVAL                                  INS00960
      GO TO 23                                                          INS00970
C  CAPACITY IS IDENTICAL FOR ALL THREE TURNING MOVEMENTS OF APPROACH    INS00980
 21   J2 = J                                                            INS00990
      CAP(J,1) = GCAP(I,J2)                                             INS01000
      IF(CAP(J,1) .EQ. 0.0) CAP(J,1) = NLANE(L)*1200.                   INS01010
      IF(VOL(J,1) .LE. 0.) GO TO 24                                     INS01020
      GO TO 225                                                         INS01030
 22   VOL (J,1) =(COUNT(2,L)+ COUNT(1,L) ) *3600./TP                    INS01040
      VOL( J, 3) = COUNT(3, L) * 3600./TP                               INS01050
C                                                                       INS01060
C  ASSUME A LEFT TURN CHANNEL THAT STORES ALL THE LEFT TURNING          INS01070
C  VEHICLES UNTIL THE LEFT TURN PHASE BEGINS.                           INS01080
      CAP(J,1) = CAP(J,1) *(1.0 +0.3/ NLANE (L))                        INS01090
      CAP(J,3) = GCAP(I,J1+2)                                           INS01100
      IF(CAP(J,3) .LE. 0.) CAP(J,3) = 1200.                             INS01110
      Y(J,3) = VOL(J,3)/ CAP(J,3)                                       INS01120
 225  FACL = 0.3/NLANE(L)                                               INS01130
C  MODIFY THE CAPACITIES DUE TO VARIATIONS IN RIGHT TURNING VEHICLES    INS01140
 23   FACR = 2.0                                                        INS01150
      IF(VOL(J,1) .LE. 0.0) GO TO 24                                    INS01160
      PCNTRN = AMIN1(COUNT(2,L)/VOL(J,1) * 3600./TP, 0.3)               INS01170
      IF(PCNTRN .GT. 0.1) FACR = 1.0                                    INS01180
      IF(NLANE(L) .EQ. 2) FACR = 0.5                                    INS01190
      IF(NLANE(L) .GT. 2) FACR = 0.25                                   INS01200
```

```
      FACR = (0.1-PCNTRN)*FACR                                          INS01210
      IF(NLANE(L) .EQ. 1 .AND. PCNTRN .GT. 0.2) FACR = FACR-(PCNTRN     INS01220
     1-0.2)/2.0                                                         INS01230
      CAP(J,1) = CAP(J,1) *(1.0+FACR+FACL)                              INS01240
C   COMPUTE WEBSTERS HIGHEST RATIO OF FLOW TO SATURATION FLOW FOR EACH  INS01250
C   PHASE                                                               INS01260
 24   Y(J,1) = 0.0                                                      INS01270
      IF ( CAP(J,1).GT.0.0 ) Y(J,1) = VOL(J,1)/ CAP(J,1)                INS01280
 25   J1 = 3-J1                                                         INS01290
C                                                                       INS01300
C   CALCULATE THE MAXIMUM Y FOR EACH PHASE                              INS01310
      YT(3) = Y(3,1)                                                    INS01320
      YT(1) = Y(1,1)                                                    INS01330
      IF(ITYPC(I) .EQ. 3 .OR. ITYPC(I) .EQ. 5) GO TO 26                 INS01340
      YT(1)= AMAX1 (Y(1,1),Y(3,1))                                      INS01350
      YT(3) = 0.0                                                       INS01360
      IF ( PH(I,3) .NE. 0.) YT(3)=AMAX1 ( Y(3,3), Y(1,3))               INS01370
 26   YT(4) = Y(4,1)                                                    INS01380
      YT(2) = Y(2,1)                                                    INS01390
      IF(ITYPC(I) .EQ. 3 .OR. ITYPC(I) .EQ. 4) GO TO 27                 INS01400
      YT(2)= AMAX1 (Y(2,1),Y(4,1))                                      INS01410
      YT(4) = 0.0                                                       INS01420
      IF ( PH(I,4) .NE. 0.0)  YT(4)= AMAX1( Y(2,3), Y(4,3))             INS01430
 27   DO 28 J=1,4                                                       INS01440
 28   PT(J,1) = PH(I,J)                                                 INS01450
      COT = ICYCL(I)                                                    INS01460
      IF (ITYPC(I) .EQ. 1) GO TO 40                                     INS01470
    CALCULATE THE CONTROLING  Y RATIO TO PHASE TIME RATIO               INS01480
      DO 29 J =1,4                                                      INS01490
      IF (PH(I,J).EQ.0) GO TO 29                                        INS01500
      IF(YT(J) .LE. 0.0) GO TO 29                                       INS01510
      SY = YT(J) + SY                                                   INS01520
      MINR =AMIN1( MINR, YT(J)/PH(I,J))                                 INS01530
      NP= NP + 1                                                        INS01540
      CONTINUE                                                          INS01550
    CALCULATE A CYCLE TIME CO                                           INS01560
      CO = 0.0                                                          INS01570
      DO 31 J = 1,4                                                     INS01580
      PT (J) =YT(J)/MINR                                                INS01590
      CO = CO + YT(J)/MINR                                              INS01600
```

```
C  CALCULATE WEBSTERS OPTIMUM CYCLE TIME                                INS01610
      COT = ICYCL(I)                                                    INS01620
      IF(SY .LT. 1.0) COT =(1.5*CI(I)*NP + 5.0)/ (1.0-SY)               INS01630
      IF (CO .GT. COT) GO TO34                                          INS01640
      IF(CO .GT. ICYCL(I)) GO TO 33                                     INS01650
      IF(COT .GT. ICYCL(I)) GO TO 33                                    INS01660
      GO TO 343                                                         INS01670
 33   CO =ICYCL(I)                                                      INS01680
      GO TO 35                                                          INS01690
 34   IF ( COT.GT.ICYCL(I)) GO TO 33                                    INS01700
 343  CO = COT                                                          INS01710
 35   COT = CO                                                          INS01720
      SPT = 0.0                                                         INS01730
      SPH = 0.0                                                         INS01740
C   FIND THE PHASE TIMES FOR THIS CYCLE LENGTH                          INS01750
      DO 37 J=1,4                                                       INS01760
      IF(PH(I,J) .LE. 0.0) GO TO 37                                     INS01770
      PT(J,1) = YT(J)/SY*COT                                            INS01780
      IF(PT(J,1) .GT. PH(I,J)) GO TO 36                                 INS01790
C  PHASE SKIPPABILITY IS ASSUMED FOR CONTROLLER.                        INS01800
      IF(PT(J,1) .LE. 0.0) GO TO 37                                     INS01810
      SPH = SPH + PH(I,J)                                               INS01820
      PT (J,1) = PH(I,J)                                                INS01830
      GO TO 37                                                          INS01840
 36   SPT = PT(J,1) + SPT                                               INS01850
 37   CONTINUE                                                          INS01860
      IF(SPT + SPH .LE. COT) GO TO 40                                   INS01870
      COT = SPT + SPH                                                   INS01880
      IF(COT .LE. ICYCL(I)) GO TO 40                                    INS01890
C  REDISTRIBUTE TIME BEYOND PHASE TIME.                                 INS01900
      RDIS = COT - ICYCL (I)                                            INS01910
      COT = ICYCL(I)                                                    INS01920
      DO 39 J=1,4                                                       INS01930
C  PHASE MIGHT BE 0.0 , EQUAL OR GREATER THAN PH(I,J)                   INS01940
      IF(PT(J,1) .LE. PH(I,J)) GO TO 39                                 INS01950
      PTJ = PT(J,1)                                                     INS01960
      PT(J,1) = PT(J,1) *(1.0-RDIS/SPT)                                 INS01970
      IF(PT(J,1) .LT. PH(I,J)) PT(J,1) = PH(I,J)                        INS01980
      SPT = SPT - PTJ                                                   INS01990
      RDIS=RDIS-PTJ+PT(J,1)                                             INS02000
```

```
 30   CONTINUE                                                          INS02010
C                                                                        INS02020
C  COMPUTE DELAY , QUEUE LENGTH, NSTOPS                                  INS02030
C    SET J1, A PHASE INSEX, J1 = 1,2,1,2 WHEN J= 1,2,3,4 AND ITYPC = 1 ,2INS02040
 40   J1 = 1                                                            INS02050
      DO 63 J=1,4                                                       INS02060
      NS = 0                                                            INS02070
      W1 = 0.0                                                          INS02080
      W2 = 0.0                                                          INS02090
      W3 = 0.0                                                          INS02100
      Q1 = 0.0                                                          INS02110
      L = LIN(I,J)                                                      INS02120
      IF(L.LE.0) GO TO 63                                               INS02130
      DO 61 K=1,3                                                       INS02140
      IF(K .EQ. 2) GO TO 60                                             INS02150
C  DETERMINE THE PHASE INDEX J2 OF THIS APPROACH AND TURNING MOVEMENT.  INS02160
      IF(ITYPC(I) .LT. 3) GO TO 43                                      INS02170
      IF(ITYPC(I) .EQ. 4 .AND. J1 .EQ. 1) GO TO 43                      INS02180
      IF(ITYPC(I) .EQ. 5 .AND. J1 .EQ. 2) GO TO 43                      INS02190
      IF(K .EQ. 3) GO TO 60                                             INS02200
      J2 = J                                                            INS02210
      GO TO 44                                                          INS02220
 43   J2 = J1 + K-1                                                     INS02230
 44   IF(PT(J2,1) .EQ. 0.0 .AND. K .EQ. 3) GO TO 60                     INS02240
C  MODIFY CAPACITY PER HOUR OF GREEN TO CAPACITY PER HOUR               INS02250
      CAP(J,K) = (AP(J,K)*(PT(J2,1)-CI(I) / 2.0)/COT                    INS02260
      CAP(J,K) = AMAX1(0.0, CAP(J,K))                                   INS02270
C  TEST THAT THE VOLUME DOES NOT EQUAL CAPACITY, CAPACITY IS DEVALUED   INS02280
      GREENQ = 0.0                                                      INS02290
      IF(PT(J2,1) .GT. 0.0) GREENQ = 3600./(COT + PT(J2,1))             INS02300
      IF(VOL(J,K) + GREENQ      .GE.      CAP(J,K)) GO TO 50            INS02310
C  COMPUTE QUE LENGTH AT END OF GREEN                                    INS02320
      W1 = (COT-PT(J2,1))**2/(2.0*COT*(1.-Y(J,K)))                      INS02330
      QO = VOL(J,K)/(2.0*(CAP(J,K) - VOL(J,K)))                         INS02340
      IF(K .GT. 1) GO TO 54                                             INS02350
C  W3 IS THE DELAY DUE TO RESIDUAL QUEUE FROM PREVIOUS TIME PERIOD      INS02360
      IF(VOL(J,1) .LE. 0.0) GO TO 52                                    INS02370
      W3 = QUE(I,J)*QUE(I,J)/((CAP(J,1)    -VOL(J,1))*2.0*VOL(J,1)) *360INS02380
      GO TO 52                                                          INS02390
 50   IF(GREENQ .GT. 1.0)                                               INS02400
```

```
     1 GO =  CAP(J,K)*(GREENQ*2.0 - 1.0 + AMIN1(VOL(J,K)-CAP(J,K),0.0)) INS02410
     2 /(2.0*GREENQ*(GREENQ-1.0))- 0.5                                 INS02420
      W1 = (COT - PT(J2,1))/2.0                                        INS02430
      IF(VOL(J,K).GT.CAP(J,K))                                         INS02440
     1Q1 = AMAX1 (0.0, (VOL(J,K)- CAP(J,K))*TP/3600.                   INS02450
     1 + AMAX1(0.0,QUE(I,J) - QO) )                                    INS02460
      IF(K .GT. 1) Q1 = AMAX1(0.0,(VOL(J,K)-CAP(J,K))*TP/3600.)        INS02470
      IF(Q1 .GT. 0.0) W3 = TP                                          INS02480
      IF(CAP(J,K) .GT. 0.0) W3 = Q1*1800./CAP(J,K)                     INS02490
 52   QO = QO + Q1                                                     INS02500
 54   W2 = QO*1800.                                                    INS02510
      IF(CAP(J,K) .NE. 0.) W2 = QO*3600./CAP(J,K)                      INS02520
      NS = (1.0-PT(J2,1)/COT)/(1.0-Y(J,K))*VOL(J,K)*TP/3600. + NS      INS02530
C  COMPUTE INVERTED VOLUME TO CAPACITY RATIOS AT EACH GATE             INS02540
      DO 56  IG = 1,NGATE                                              INS02550
      K1 = 1                                                           INS02560
      IF ( L .EQ. LGATE(1,IG))GO TO 57                                 INS02570
      K1 = 2                                                           INS02580
      IF ( LCON (L, K) .EQ. LGATE(2,IG)) GO TO 57                      INS02590
 56   CONTINUE                                                         INS02600
      GO TO 60                                                         INS02610
 57   VCR (IG,K1, K) = 0.0                                             INS02620
      IF(VOL(J,1) .NE. 0.0) VCR(IG,K1,K) = CAP(J,1)/VOL(J,1)           INS02630
      IF (VOL(J, K) .NE. 0.0) VCR(IG,K1,K) = CAP(J,K)/ VOL(J,K)        INS02640
 60   DELA(L,K) = W1 + W2 + W3                                         INS02650
      QQ(J,K) = QO                                                     INS02660
      IF (K.GT. 1) GO TO 61                                            INS02670
C   COMPUTE THE AVERAGE QUEUE LENGTH FOR A SIGNAL CYCLE.               INS02680
C  QUEUE AT END OF GREEN + AVERAGE QUEUE DURING RED PHASE              INS02690
      QUEUE(L) = QO + W1*VOL(J,1)/3600.                                INS02700
      QO = 0.0                                                         INS02710
 61   CONTINUE                                                         INS02720
      QUEUE(L) = QUEUE(L) + QO                                         INS02730
      NSTOPS(L) = NS                                                   INS02740
C   MOVE LEFT TURNING Q AND VOL FOR PRINTING PURPOSES                  INS02750
      QQ(J,2) = QQ(J,3)                                                INS02760
      VOL(J,2) = VOL(J,3)                                              INS02770
      CAP(J,2) = CAP(J,3)                                              INS02780
      Y(J,2) = Y(J,3)                                                  INS02790
      QUE(I,J) = QQ(J,1) + QQ(J,3)                                     INS02800
```

```
      DELAI(I,J) = DELA(L,1)                                            INS02810
      DELAI(I,J+4) = DELA(L,3)                                          INS02820
   63 J1 = 3-J1                                                         INS02830
      QG(9,1)=0.0                                                       INS02840
      VOL(9,1)=0.0                                                      INS02850
      CAP(9,1)=0.0                                                      INS02860
      Y(9,1)=0.0                                                        INS02870
      PRINT 9631, I,(J,J=1,4)                                           INS02880
 9631 FORMAT (13H0INTERSECTION I5/ 4(23X,5HPHASE,I2))                   INS02890
      PRINT 9632, (PT(J,1),J=1,4)                                       INS02900
 9632 FORMAT (5H TIME, 4(18X,F4.0,8X))                                  INS02910
      PRINT 9633, (YT(J),J=1,4)                                         INS02920
 9633  FORMAT (5H V/GC, 4(18X,F7.5,5X))                                 INS02930
      IF(ITYPC(I) .GT. 2) GO TO 6331                                    INS02940
      I2=3                                                              INS02950
      I4=4                                                              INS02960
      I5=5                                                              INS02970
      I6=6                                                              INS02980
      I7=7                                                              INS02990
      I8=8                                                              INS03000
      I9=7                                                              INS03010
      I10=6                                                             INS03020
      GO TO 6335                                                        INS03030
 6331 IF(ITYPC(I) .GT. 3) GO TO 6332                                    INS03040
      I2=9                                                              INS03050
      I4=9                                                              INS03060
      I6=9                                                              INS03070
      I8=9                                                              INS03080
      I5=3                                                              INS03090
      I7=4                                                              INS03100
      I9=9                                                              INS03110
      I10=4                                                             INS03120
      GO TO 6335                                                        INS03130
 6332 IF(ITYPC(I) .GT. 4) GO TO 6333                                    INS03140
      I2=3                                                              INS03150
      I4=9                                                              INS03160
      I8=9                                                              INS03170
      I5=5                                                              INS03180
      I6=6                                                              INS03190
      I7=4                                                              INS03200
```

```
      I9=7                                                              INS03210
      I10=4                                                             INS03220
      GO TO 6335                                                        INS03230
 6333 CONTINUE                                                          INS03240
      I2=9                                                              INS03250
      I6=9                                                              INS03260
      I4=4                                                              INS03270
      I5=3                                                              INS03280
      I7=7                                                              INS03290
      I8=8                                                              INS03300
      I9=9                                                              INS03310
      I10=6                                                             INS03320
 6335 PRINT 96335, NPHASE(1),NPHASE(I2),NPHASE(2),NPHASE(I4), NPHASE(I5)INS03330
     1, NPHASE(I6), NPHASE(I7), NPHASE(I8)                              INS03340
96335 FORMAT(5X,4(7X,A8,3X,A8,4X))                                      INS03350
      PRINT 9634, (DELAI(I,J),DELAI(I,J+2),J=1,2),DELAI(I,I5),DELAI(I,I9INS03360
     1), DELAI(I,I10),DELAI(I,I6)                                       INS03370
 9634 FORMAT (5H DELA, 4(5X,2F10.0,5X))                                 INS03380
      PRINT 9635, (QQ(J,1),QQ(J+2,1),J=1,2),QQ(I5,1),QQ(I9,1),          INS03390
     1QQ(I10,1),QQ(I8,1)                                                INS03400
 9635 FORMAT (6H QUEUE 4(4X,2F10.0,6X))                                 INS03410
      PRINT 9636, (VOL(J,1),VOL(J+2,1),J=1,2),VOL(I5,1),VOL(I9,1),      INS03420
     1VOL(I10,1),VOL(I8,1)                                              INS03430
 9636 FORMAT (7H VOLUME 4(3X,2F10.0,7X))                                INS03440
      PRINT 9637, (CAP(J,1),CAP(J+2,1),J=1,2),CAP(I5,1),CAP(I9,1),      INS03450
     1CAP(I10,1),CAP(I8,1)                                              INS03460
 9637 FORMAT (9H CAPACITY 4(1X,2F10.0,9X))                              INS03470
      PRINT 9638, (Y(J,1),Y(J+2,1),J=1,2),Y(I5,1),Y(I9,1),Y(I10,1)      INS03480
     1,Y(I8,1)                                                          INS03490
 9638 FORMAT (7H V/GCAP 4(7X,2F10.5,7X))                                INS03500
      GO TO 90                                                          INS03510
 90   CONTINUE                                                          INS03520
      RETURN                                                            INS03530
      END                                                               INS03540
```

```
      SUBROUTINE INSECU                                                 INU00010
      COMMON / COMM/ I,Z,L,J, K, TOD, DOW, TP, TOTATT, TTPZ,TOTGEN      INU00020
      COMMON /LINK/  NLINK, NLANE(200), X1(200), Y1(200),X2(200),Y2(200)INU00030
     1, LCAP(200), DIST(200), VEL(200), LCON (200,3)                    INU00040
     2,NSTOPS(200)                                                      INU00050
      COMMON /INTRST/NI, LIN(70,4),ITYPC(70),ICYCL(70), PH(70,4), CI(70)INU00060
     1 , GCAP(70,4),QUE(70,4)                                           INU00070
      COMMON /ZONES/    NZONES,NZLINK(50), ZLINKS(12,50), ZATTR(50),    INU00080
     1ZGENR (50), NEXT, ZAT(2), ZGT(2), ZVT(3,50), ZNAME(50)            INU00090
      COMMON / GATE/ NGATE, LGATE(2,10), VCR(10, 2,4)                   INU00100
      COMMON /VOLUME/ COUNT(4,200)                                      INU00110
      COMMON /PARKZ/ PZ,       PV(50), PLA(50), VZ(50), ZV(50)          INU00120
      COMMON TT(200,3), DELA(200,3), PRT(50), QUEUE(200)                INU00130
      COMMON                 PT(4,3),VOL(4,3),Y(4,3),CAP(4,3),YT(4)     INU00140
     1,DELAI(70,4)                                                      INU00150
      DIMENSION W(2),V(2)                                               INU00160
      INTEGER W                                                         INU00170
C                                                                       INU00180
C  UNSIGNALIZED INTERSECTIONS ARE HANDLED IN THIS SECTION               INU00190
C  NOTE THAT THE HCM RECOMMENDS ,IN THE ABSENSE OF A BETTER ALTERNATIVE,INU00200
C  TREATING AN UNCONTROLLED INTERSECTION AS A TYPE 2 SIGNALIZED INTERSECINU00210
C  TION.  ACTUALLY INSECU IS MOST ACCURATE IN TREATING A 4 WAY STOP     INU00220
C  BUT CAN USUALLY BE USED FOR ANY UNSIGNALIZED INTERSECTION.           INU00230
      SLANE = 0.0                                                       INU00240
      ABSV=0.0                                                          INU00250
      SVOL=0.0                                                          INU00260
      SCAP = 0.0                                                        INU00270
      I4 = 0                                                            INU00280
      J1 = 1                                                            INU00290
      W(1)=0.0                                                          INU00300
      W(2)=0.0                                                          INU00310
      V(1)=0.0                                                          INU00320
      V(2)=0.0                                                          INU00330
      J2 = 1                                                            INU00340
      DO 73 J = 1,4                                                     INU00350
C  SUM THE VOLUME AND VOLUME DIFFERENCE FOR VEHICLES ON N-S AND E-W APPRINU00360
      L = LIN (I,J)                                                     INU00370
      VOL(J,1) = 0.0                                                    INU00380
      IF (L .LE. 0) GO TO 73                                            INU00390
      DELA(L,1) = 0.0                                                   INU00400
```

```
      I4 = I4 + 1                                                       INU00410
      VOL(J,1) =(COUNT(1,L) + COUNT(2,L) + COUNT(3,L) ) *3600./TP        INU00420
      IF(VOL(J,1) .GT. VOL(J1,1)) J1 = J                                INU00430
      IF(ITYPC(I) .GE. -1) GO TO 715                                    INU00440
      SVOL = SVOL + VOL(J,1)                                            INU00450
      ABSV = VOL(J,1) - ABSV                                            INU00460
      IF ( GCAP(I,J) .GT.0) GO TO 727                                   INU00470
      IF ( NLANE(L) - 1) 71, 723, 725                                   INU00480
 71   STOP 1                                                            INU00490
 715  V(J2)= AMAX1(VOL(J,1),V(J2))                                      INU00500
      W(J2) = MAX0(NLANE(L),W(J2))                                      INU00510
      IF(GCAP(I,J) .LE. 0.0) GCAP(I,J) = 1800.*NLANE(L)                 INU00520
      GO TO 729                                                         INU00530
 723  GCAP( I,J) = 475.                                                 INU00540
      GO TO 727                                                         INU00550
 725  GCAP(I,J) = 450.*NLANE(L)                                         INU00560
C  SUM THE CAPACITIES FOR ALL APPROACH LEGS                             INU00570
 727  SCAP = SCAP + GCAP(I, J)                                          INU00580
 729  SLANE = SLANE + NLANE(L)                                          INU00590
 73   J2 = 3-J2                                                         INU00600
      J1 = MOD(J1 , 2)                                                  INU00610
      IF(ITYPC(I) .GE. -1) GO TO 735                                    INU00620
      ABSV = ABS(ABSV)                                                  INU00630
C  COMPUTE A MODIFIED INTERSECTION CAPACITY BASED ON VOLUME SPLIT       INU00640
C  , FROM HCM TABLE 6.8 )                                               INU00650
      IF(SVOL .GT. 0.0)                                                 INU00660
     1SNCAP = SCAP * AMAX1(1.0 - ABSV/SVOL/1.9, .816)                   INU00670
      IF (I4 .LT. 4) SNCAP = SCAP                                       INU00680
      GO TO 738                                                         INU00690
 735  IF(V(1) .LT. 1.0 .OR. V(2) .LT. 1.0) GO TO 738                    INU00700
      R = V(1)/V(2)*W(2)/W(1)                                           INU00710
      SNCAP = 1800.*(W(1)*R+W(2))/(R+1.0)                               INU00720
      SVOL = V(1) + V(2)                                                INU00730
C  COULD INCLUDE MODIFICATION TO GCAP AT A 2 WAY STOP BASED ON GAP ACCEPINU00740
 738  CONTINUE                                                          INU00750
C  COMPUTE NEW CAPACITIES FOR EACH APPROACH                             INU00760
      IG = 0                                                            INU00770
      IG1 = 0                                                           INU00780
      DO 77 J = 1,4                                                     INU00790
      L = LIN(I,J)                                                      INU00800
```

```
      DELAI(I,J) = 0.0                                                  INU00810
      QO = 0.0                                                          INU00820
      IF(L .LE. 0) GO TO 767                                            INU00830
      NSTOPS(L) = 0                                                     INU00840
      IF(SVOL .LE. 0.) GO TO 76                                         INU00850
      CAP(J,1) = VOL(J,1)*SNCAP/SVOL                                    INU00860
C  TEST IF OVER CAPACITY DEMAND                                         INU00870
      QO = AMIN1(VOL(J,1), (VOL(J,1) - CAP(J,1))*TP/3600.+QUE(I,J)+5.0) INU00880
      IF(VOL(J,1) .LT. CAP(J,1)) QO = AMIN1(5.0,1.0/(SNCAP/SVOL-1.0))   INU00890
C  COMPUTE THE NEW CAPACITY TO VOLUME RATIO AT EACH GATE                INU00900
      IF(IG .GT. 0) GO TO 742                                           INU00910
 741  IG = IG + 1                                                       INU00920
 742   IF (IG.GT. NGATE) GO TO 749                                      INU00930
      IF (L .NE. LGATE(1,IG)) GO TO 746                                 INU00940
      IF(VOL(J,1) .EQ. 0.)GO TO 749                                     INU00950
      DO 744 K=1,3                                                      INU00960
      VCR(IG , 1, K) = CAP(J,1) / VOL(J,1)                              INU00970
C     IF(VOL(J,K) .NE. 0) VCR(IG,1,K) = CAP(J,K)/VOL(J,K)               INU00980
 744  CONTINUE                                                          INU00990
      GO TO 749                                                         INU01000
 746  DO 747 K=1,3                                                      INU01010
      IF(LCON(L,K) .NE. LGATE(2,IG)) GO TO 747                          INU01020
      IF(VOL(J,1) .NE. 0.0)VCR(IG,2,K) = CAP(J,1)/VOL(J,1)              INU01030
      GO TO 749                                                         INU01040
 747  CONTINUE                                                          INU01050
 748  IF(IG1 .EQ. 0) GO TO 741                                          INU01060
 749  IG1 = IG                                                          INU01070
C  DELAY ON EACH APPROACH                                               INU01080
 75   CONTINUE                                                          INU01090
      PNS = SVOL/SNCAP                                                  INU01100
      IF(VOL(J,1) .LE. 0.0) GO TO 76                                    INU01110
C   COMPUTE THE PERCENTAGE STOPS                                        INU01120
      IF(ITYPC(I) .EQ. -2) PNS = 1.0                                    INU01130
      IF(ITYPC(I) .EQ. -1 .AND. MOD(J,2) .NE. J1) PNS = 1.0             INU01140
C  COMPUTE DELAY BASED ON INVERSE OF CAPACITY.  MAKE THEORETICAL MAXIMUMINU01150
C  OF CAPACITY 2*GCAP(I,J)                                              INU01160
      DELA (L,1) = NLANE(L)/(AMIN1(2.0, SNCAP/SVOL) *GCAP(I,J))*3600.*  INU01170
     1 (PNS + QO)                                                       INU01180
      NSTOPS(L) = PNS*VOL(J,1)*TP/3600.                                 INU01190
C   IF CAPACITY EXCEEDED ASSIGN EQUAL VEHICLE PROCESSING TIME TO EACH APINU01200
```

```
      IF(SNCAP .LT. SVOL) DELA(L,1) = SLANE/SNCAP*3600.*                INU01210
     1 (1.0 + (QO + QUE(I,J) + 5.0)/2.0)                                INU01220
   76 DELA(L,2)=DELA(L,1)                                               INU01230
      DELA(L,3)=DELA(L,1)                                               INU01240
      DELAI(I,J) = DELA(L,1)                                            INU01250
C  QUEUE LENGH AT THE END OF TP                                         INU01260
 767  QUEUE(L) = QO                                                     INU01270
      QUE(I,J) = QO                                                     INU01280
 77   CONTINUE                                                          INU01290
      IF(ISAVE .NE. I) PRINT 97                                         INU01300
 97   FORMAT(1H-18X 85H N-APPR DELAY QUEUE    E-APPR DELAY QUEUE    S-APINU01310
     1PR DELAY QUEUE    W-APPR DELAY QUEUE )                           INU01320
      ISAVE = I+1                                                       INU01330
      PRINT 98, I, (DELAI(I,J), QUE(I,J), J=1,4)                        INU01340
 90   CONTINUE                                                          INU01350
 98   FORMAT (13H INTERSECTION I5, 4(F10.0,F7.0,5X))                    INU01360
      RETURN                                                            INU01370
      END                                                               INU01380
```

```
      SUBROUTINE ZAREA                                                  ZAR00010
C  EACH ZONE IS DEFINED BY ADJACENT LINKS FROM WHICH VEHICLES ACCESS    ZAR00020
C  THAT ZONE.  IN ADDITION PARKING ZONE LINKS MUST ENCLOSE AN AREA      ZAR00030
C  AND THAT AREA IS ASSIGNED A CERTAIN NUMBER OF VEHICLE PARKING SPACES.ZAR00040
C  OF COURSE ALL LINKS SURROUNDING THE AREA MUST BE CONNECTED, OTHERWISEZAR00050
C  THE AREA WILL NECESSARILY BE INFINITE.  ONE EXCPTION---THE LAST LINK ZAR00060
C  NEED NOT CONNECT TO THE FIRST LINK WHEN LESS THAN 6 LINKS DEFINE THE ZAR00070
C  ZONE.                                                               ZAR00080
      COMMON /LINK/  NLINK, NLANE(200), X1(200), Y1(200),X2(200),Y2(200)ZAR00090
     1, LCAP(200), DIST(200), VEL(200), LCON (200,3)                    ZAR00100
      COMMON  /ZONES/  NZONES, NZLINK(50), ZLINK(12,50),ZATTR(50),      ZAR00110
     1ZGENR (50), NEXT. ZAT(2), ZGT(2), ZVT(3,50), ZNAME(50)            ZAR00120
      COMMON/EXCLUD/NOPARK                                              ZAR00130
      INTEGER ZNAME                                                     ZAR00140
      INTEGER ZLINK                                                     ZAR00150
      DIMENSION  PX(7), PY(7), TRIARE(7)                                ZAR00160
      COMMON /PARKZ/  PZ,    PV(50), PLA(50), VZ(50), ZV(50),PD,PLS,PLROZAR00170
     1, PNOS(50),PLL(50)                                                ZAR00180
      INTEGER PNOS                                                      ZAR00190
      INTEGER Z                                                         ZAR00200
      DO 52 Z=1,NZONES                                                  ZAR00210
      SAVPLA = PLA(Z)                                                   ZAR00220
 12   CONTINUE                                                          ZAR00230
      IF(Z .LE. NEXT) GO TO 52                                          ZAR00240
      NLT = PNOS(Z)                                                     ZAR00250
      XYMAX = 0.0                                                       ZAR00260
      DO 29 NL = 1, NLT                                                 ZAR00270
      L = IABS (ZLINK(NL.Z))                                            ZAR00280
      XYMAX=AMAX1((X1(L)-X2(L))**2+(Y1(L)-Y2(L))**2                     ZAR00290
     &.XYMAX)                                                           ZAR00300
C  SPECIAL HANDLING IS REQUIRED FOR THE FIRST AND SECOND LINKS          ZAR00310
      IF ( NL - 2) 13, 17, 20                                           ZAR00320
 13   PX(1) = X1(L)                                                     ZAR00330
      PX(2) = X2(L)                                                     ZAR00340
      PY(1) = Y1(L)                                                     ZAR00350
      PY(2) = Y2(L)                                                     ZAR00360
      LT = L                                                            ZAR00370
      GO TO 29                                                          ZAR00380
 17   IF (PX(2) .NE. X2(L) ) GO TO 18                                   ZAR00390
      IF (PY(2) .EQ. Y2(L) ) GO TO 24                                   ZAR00400
```

```
C  THE END POINTS FAIL TO MATCH, TWO POSSIBLE MATCHES REMAIN          ZAR00410
 18    IF (PX(2) .NE. X1(L)) GO TO 19                                 ZAR00420
       IF (PY(2) .EQ. Y1(L)) GO TO 26                                 ZAR00430
C  THE END POINTS FAIL TO MATCH, ONE POSSIBLE MATCH REMAINS           ZAR00440
 19    PX(2) = PX(1)                                                  ZAR00450
       PY(2) = PY(1)                                                  ZAR00460
       PX(1) = X2(LT)                                                 ZAR00470
       PY(1) = Y2(LT)                                                 ZAR00480
C  DETERMIN WHICH END OF LINK L CONECTS TO THE PREVIOUS LINK          ZAR00490
 20    IF ( PX(NL) .NE. X1(L)) GO TO 22                               ZAR00500
       IF ( PY(NL) .EQ. Y1(L)) GO TO 26                               ZAR00510
 22    IF ( PX(NL) .NE. X2(L)) GO TO 23                               ZAR00520
       IF( PY(NL) .EQ. Y2(L)) GO TO 24                                ZAR00530
C  THE NUMBER OF CONNECTED POINTS IS SAVED IN NPT                     ZAR00540
 23    NPT = NL                                                       ZAR00550
C  THE FIGURE IS NOT CONNECTED AT THE LINK END POINTS                 ZAR00560
       PRINT 923, ZNAME(Z), ZLINK(NL-1,Z),L                           ZAR00570
 923   FORMAT (7H ZONE  ,I2,32H  IS DISCONNECTED BETWEEN LINKS I4,    ZAR00580
      1 4H AND, I4)                                                   ZAR00590
       GO TO 30                                                       ZAR00600
 24    PX(NL+1) = X1(L)                                               ZAR00610
       PY(NL+1) = Y1(L)                                               ZAR00620
       GO TO 27                                                       ZAR00630
 26    PX(NL+1) = X2(L)                                               ZAR00640
       PY(NL+1) = Y2(L)                                               ZAR00650
 27    IF (NL .EQ. 3 .OR. NL .EQ. 5) GO TO 29                         ZAR00660
       TRIARE(NL) = (PX(NL-1)*PY(NL) - PX(NL-1)*PY(NL+1) + PX(NL)*PY(NL+1ZAR00670
      1 ) - PX(NL)*PY(NL-1) + PX(NL+1)*PY(NL-1) - PX(NL+1)*PY(NL))/ 2.0 ZAR00680
       TRIARE(NL) = ABS(TRIARE(NL))                                   ZAR00690
 29    NPT = NL + 1                                                   ZAR00700
       IF(NPT .LE. 6) GO TO 30                                        ZAR00710
       NPT = 6                                                        ZAR00720
       GO TO 31                                                       ZAR00730
C                                                                     ZAR00740
C       EACH AREA COMPUTED IS A POSSIBLE SUBTRACTION FROM THE TOTAL AREA.ZAR00750
C       WHEN  IT COVERS THE TRIANGLE CREATED BY CONSTRUCTED SIDES, SUB- ZAR00760
C       TRACT THE AREA.                                               ZAR00770
C  IF A CLOSING LINK WAS LEFT OUT ASSUME IT EXISTS                    ZAR00780
 30    IF ( PX(NPT) .NE. PX(1)) GO TO 32                              ZAR00790
       IF ( PY(NPT) .NE. PY(1)) GO TO 32                              ZAR00800
```

```
      NPT = NPT -1                                                      ZAR00810
 31   PLL(Z) = -SQRT(XYMAX)                                             ZAR00820
      GO TO 40                                                          ZAR00830
C    ONE LINK ZONES ARE LEGAL. ASSUME 12+12+14+14 FOOT WIDTH            ZAR00840
 32   IF(NPT .GT. 2) GO TO 35                                           ZAR00850
      PLL(Z) = SQRT(XYMAX)                                              ZAR00860
      PLA(Z) = PLL(Z) * 52.0                                            ZAR00870
      GO TO 50                                                          ZAR00880
   35 PLL(Z)=AMAX1((PX(NPT)-PX(1))**2+(PY(NPT)-PY(1))**2,XYMAX)         ZAR00890
      PLL(Z) = -SQRT(PLL(Z))                                            ZAR00900
      IF (NPT .EQ. 3 .OR. NPT.EQ. 5) GO TO 40                           ZAR00910
      TRIARE (NPT) =(PX(NPT-1)*PY(NPT) - PX(NPT-1)*PY(1) + PX(NPT)*PY(1)ZAR00920
     1 - PX(NPT)*PY(NPT-1) + PX(1)*PY(NPT-1) - PX(1)*PY(NPT))/ 2.0      ZAR00930
C  INITIALIZE THE AREA OF THE ZONE                                      ZAR00940
      TRIARE(NPT) = ABS(TRIARE(NPT))                                    ZAR00950
 40   IF(NPT .LE. 4) GO TO 413                                          ZAR00960
      PLA( Z) = ( PX(5)*PY(1) - PX(5)*PY(3) + PX(1)*PY(3) - PX(1)*PY(5) ZAR00970
     1 + PX(3)*PY(5) - PX(3)*PY(1))/ 2.0                                ZAR00980
      PLA(Z) = ABS(PLA(Z))                                              ZAR00990
      GO TO 415                                                         ZAR01000
 413  PLA(Z) = TRIARE( 2)                                               ZAR01010
 415  IF (NPT .LE. 3) GO TO 50                                          ZAR01020
      DO 495 NL= 4, NPT                                                 ZAR01030
C  COMPUTE AREA OF TRIANGLE WITH 1,2, OR 3 COMMON SIDES                 ZAR01040
C   WHERE NL-3 IS THE NUMBER OF SIDES HELD IN COMMON                   ZAR01050
      IF( NL - 5 ) 417,43, 45                                           ZAR01060
 417  IC = 1                                                            ZAR01070
C  IA IS THE VERTEX OF THE TRIANGLE 1-2-3 AND COMMON SIDE IS 1-3        ZAR01080
      IA = 2                                                            ZAR01090
      IF ( NPT.NE.4) GO TO 42                                           ZAR01100
 419  IB = 4                                                            ZAR01110
      GO TO 425                                                         ZAR01120
C  IB IS THE VERTEX OF THE TRIANGLE  1-5-3 AND COMMON SIDE IS 1-3       ZAR01130
 42   IB = 5                                                            ZAR01140
C  COMPUTE THE SLOPE OF THE LINE(OR SIDE) HELD COMMONLY BY TWO TRIANGLESZAR01150
 425  IF (PX(1) .EQ. PX(3) ) GO TO 46                                   ZAR01160
      CM = (PY(1)- PY(3))/(PX(1)- PX(3))                                ZAR01170
C  COMPUTE THE INTERCEPT OF THIS LINE WITH THE Y AXIS                   ZAR01180
      YN =  PY(1) - CM* PX(1)                                           ZAR01190
      GO TO 47                                                          ZAR01200
```

```
C   IA IS THE VERTEX OF THE TRIANGLE 3-4-5, IB IS VERTEX OF 5-1-3        ZAR01210
 43    IA = 4                                                            ZAR01220
       IB =1                                                             ZAR01230
       IC = 3                                                            ZAR01240
C   COMPUTE THE SLOPE OF THE COMMON SIDE 3-5                             ZAR01250
       IF(PX(3) .EQ. PX(5)) GO TO 46                                     ZAR01260
       CM = (PY(3) - PY(5))/(PX(3)-PX(5))                                ZAR01270
       YK = PY(3) - CM* PX(3)                                            ZAR01280
       GO TO 47                                                          ZAR01290
C   IA IS THE VERTEX OF THE TRIANGLE  5-6-1, IB IS VERTEX OF  1-3-5      ZAR01300
 45    IA = 6                                                            ZAR01310
       IB = 3                                                            ZAR01320
       IC = 1                                                            ZAR01330
C   COMPUTE SLOPE OF COMMON SIDE  5-1                                    ZAR01340
       IF(PX(5) .EQ. PX(1)) GO TO 46                                     ZAR01350
       CM = (PY(5) - PY(1))/(PX(5) - PX(1))                              ZAR01360
       YK =  PY(5) - CM*PX(5)                                            ZAR01370
       GO TO 47                                                          ZAR01380
 46    IF (PX(IA) .LT. PX(IC)) GO TO 463                                 ZAR01390
       IF (PX(IB) .GT. PX(IC))GO TO 48                                   ZAR01400
       GO TO 493                                                         ZAR01410
 463   IF (PX(IB) .GT. PX(IC))GO TO 493                                  ZAR01420
       GO TO 48                                                          ZAR01430
C   USING YK AD THE NEW Y ORIGIN COMPUTE THE SLOPE TO EACH NON-COMMON    ZAR01440
C   TRIANGLE VERTEX                                                      ZAR01450
 47    AM =(PY( IA) - YK) / PX(IA)                                       ZAR01460
       BM =(PY(IB ) - YK) / PX(IB)                                       ZAR01470
C   DETERMINE IF BOTH TRIANGLES FALL ON THE SAME SIDE OF THEIR COMMON    ZAR01480
C   SIDE, AND SUBTRACT THE SMALLER IF THEY DO.                           ZAR01490
       IF ( AM .GT. CM) GO TO 49                                         ZAR01500
       IF ( BM .GT. CM) GO TO 493                                        ZAR01510
C    TRIANGLES OVERLAP                                                   ZAR01520
 48    PLA(Z) = PLA(Z) - TRIARE(IA)                                      ZAR01530
       GO TO 495                                                         ZAR01540
 49    IF (BM .GT. CM) GO TO 48                                          ZAR01550
C    TRIANGLES DO NOT OVERLAP                                            ZAR01560
 493   PLA(Z) = PLA(Z) + TRIARE(IA)                                      ZAR01570
 495   CONTINUE                                                          ZAR01580
 50    PLA (Z) = ABS (PLA(Z))                                            ZAR01590
       IF(PLL(Z) .LT. 0.0) PLL(7) = -PLA(7)/PLL(Z) - PLL(Z)              ZAR01600
```

```
      IF(SAVPLA .NE. 0.0) PLA(Z) = SAVPLA                                ZAR01610
   52 PLL(Z) = AMAX1(PLL(Z),1.0)                                         ZAR01620
      IF(NOPARK.EQ.1) PRINT 960                                          ZAR01630
  960 FORMAT(/,1X,'PARKING ZONE CAPACITIES AND LENGTHS NOT USED IN THE  ZAR01640
     &CALCULATION',/)                                                    ZAR01650
C     PRINT 953                                                          ZAR01660
  953 FORMAT (42H-PARKING ZONE CAPACITIES AND TRIP LENGTHS /24H ZONE  CAZAR01670
     1PACITY  LENGTH )                                                   ZAR01680
      IZ = NEXT + 1                                                      ZAR01690
  957 FORMAT(2X,I2,2E15.4)                                               ZAR01700
      DO 57 Z = IZ,NZONES                                                ZAR01710
      PCAP = PLA(Z)/PD                                                   ZAR01720
C     PRINT 957, ZNAME(Z),PCAP,PLL(Z)                                    ZAR01730
   57 CONTINUE                                                           ZAR01740
      RETURN                                                             ZAR01750
      END                                                                ZAR01760
```

```
      SUBROUTINE ROUTEV                                                 ROU00010
      COMMON / COMM/ I,Z,L,J, K, TOD, DOW, TP, TOTATT, TTPZ,TOTGEN,ITM  ROU00020
      COMMON /LINK/  NLINK, NLANE(200), X1(200), Y1(200),X2(200),Y2(200)ROU00030
     1, LCAP(200), DIST(200), VEL(200), LCON (200,3)                    ROU00040
     2,NSTOPS(200)                                                      ROU00050
      COMMON / INTRST/ NINS, LIN(70,4),ITYPC (70), CYCL(70), PH(70,4),  ROU00060
     1 CI(70), GCAP(70,4), QUE(70,4)                                    ROU00070
      COMMON /ZONES/ NZONES, NZLINK(50), ZLINKS(12,50), ZATTR(50),      ROU00080
     1 ZGENR(50), NEXT , ZAT(2), ZGT(2), ZVT(3,50),ZNAME(50)            ROU00090
      INTEGER ZNAME                                                     ROU00100
      INTEGER ZLINKS                                                    ROU00110
      COMMON / GATE/ NGATE, LGATE(2,10), VCR(10, 2,4)                   ROU00120
      COMMON /VOLUME/ COUNT(4,200)                                      ROU00130
      COMMON /PARKZ/  PZ,    PV(50), PLA(50), VZ(50), ZV(50), PD,PLS,PLBROU00140
     1,  PNOS(50) , PLL(50)                                             ROU00150
      COMMON /VEHTYP/ VT, VTP(3), VTE(3), VTMULT(2),SZVTP(3,2)          ROU00160
      COMMON TT(200,3),NC,NR,RO, R(200), CO, C(200)                     ROU00170
      DIMENSION DELA(200,3)                                             ROU00180
      EQUIVALENCE (NC,DELA(1))                                          ROU00190
      INTEGER R                                                         ROU00200
      COMMON   RG(200,10), CG(200,10) ,            LCRZ(10,50,2),GCR( 4) ROU00210
     1, LISTL(4), CLIST(4), LISTS(4)                                    ROU00220
     2,VEHS(4), LCRZF(10,50)                                            ROU00230
      INTEGER RG, Z                                                     ROU00240
      DIMENSION QTEMP(4)                                                ROU00250
      DO 12 II7=1,200                                                   ROU00260
   12 C(II7)=0.0                                                        ROU00270
      IF(ITM .LE. 0) ITM = 1                                            ROU00280
      STP = TP                                                          ROU00290
      DO 80 IT = 1,ITM                                                  ROU00300
      TP = (STP*IT)/ITM                                                 ROU00310
      DO 25 IG = 1, NGATE                                               ROU00320
C  GET C AND R ARRAYS FROM EACH GATE LINK PAIR                          ROU00330
 18   CALL MINPTH (LGATE(1,IG) ,2)                                      ROU00340
      DO 19 L =1,NLINK                                                  ROU00350
      RG (L, IG) = R(L)                                                 ROU00360
 19   CG (L, IG) = C(L)                                                 ROU00370
C                                                                       ROU00380
C  GET ARRAY OF LINK NUMBERS WHICH IDENTIFY THE COST AND ROUTE FROM EACHROU00390
C  IG TO EACH ZONE                                                      ROU00400
```

```
      DO 24   Z =1,NZONES                                              ROU00410
      IORE  = 1                                                        ROU00420
      IF(Z .LE. NEXT) IORE = 2                                         ROU00430
      C1 = 1.0E10                                                      ROU00440
      L1 = 0                                                           ROU00450
      L2 = 0                                                           ROU00460
      NZL= NZLINK  (Z)                                                 ROU00470
      DO 21   J = 1, NZL                                               ROU00480
      L = ZLINKS(J,Z)                                                  ROU00490
      IF(L .LE. 0) GO TO 21                                            ROU00500
C  WHERE A GATE LINK AND ZONE LINK ARE EQUAL(I.E. EXTERNAL ZONE TO     ROU00510
C  OUTGOING GATE LINK, AND INTERNAL ZONE TO INCOMING GATE LINK)        ROU00520
C  PATHS ARE VOID.                                                     ROU00530
      IF (LGATE(IORE,IG) .EQ. L) GO TO 21                              ROU00540
C  GET 2 MIN COSTS TO EACH ZONE                                        ROU00550
      IF ((C(L) .GE. 1.0E10) GO TO 21                                  ROU00560
      IF (L2 .LE. 0) GO TO 20                                          ROU00570
      IF ((C(L2) .LT. C(L)) GO TO 21                                   ROU00580
      L2 = L                                                           ROU00590
 20   IF ( C1 .LT. C(L) ) GO TO 21                                     ROU00600
      L2 = L1                                                          ROU00610
      C1 = C(L)                                                        ROU00620
      L1 = L                                                           ROU00630
 21   CONTINUE                                                         ROU00640
C  IDENTIFY LINK WHICH SHOWS LEAST COST                                ROU00650
      LCRZ (IG, Z, 2) = L2                                             ROU00660
 23   LCRZ (IG,Z,1) = L1                                               ROU00670
C   FIND THE FIRST LINK IN THE ROUTE FROM THE GATE TO ZONE             ROU00680
 234   IF(L1 .LE. 0) GO TO 24                                          ROU00690
      L2 = L1                                                          ROU00700
      L1 = R(L2)                                                       ROU00710
      GO TO 234                                                        ROU00720
 24   LCRZF(IG,Z ) = L2                                                ROU00730
 25   CONTINUE                                                         ROU00740
C                                                                      ROU00750
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - ROU00760
C  FOR EACH ZONE GET A COST AND ROUTE ARRAY TO ALL LINKS IN THE NETWORK.ROU00770
C  DETERMINE A COST AND ROUTE THRU EACH GATE TO EACH INT/EXTERNAL ZONE, ROU00780
C  DETERMINE THE NUMBER OF VEH GOING THRU EACH GATE, APPLY VEH TO THE  ROU00790
C  APPROPRIATE ROUTE                                                   ROU00800
```

```
C - - -
```

```
      DO 334 KK = K,3                                                   ROU01210
      LISTG(JK+1) = LISTG(JK)                                           ROU01220
      LISTL(JK+1)=LISTL(JK)                                             ROU01230
      CLIST(JK+1) =CLIST(JK)                                            ROU01240
      GCR(JK+1) = GCR(JK)                                               ROU01250
 334  JK = JK-1                                                         ROU01260
 335  LISTG(K) = IG                                                     ROU01270
      LISTL(K) = LZ                                                     ROU01280
      CLIST(K) = C(LG) + CG(LZ,IG)                                      ROU01290
      NK = MIND(NK+1,4)                                                 ROU01300
      I1 = 1                                                            ROU01310
C  LOCATE THE CORRECT CAPACITY FOR THIS TURNING MOVEMENT               ROU01320
      IF(IORE .EQ. 1) GO TO 343                                         ROU01330
      LG1 = R( LG)                                                      ROU01340
      IF(LG1 .LE. 0) GO TO 35                                           ROU01350
      DO 34 J=1,3                                                       ROU01360
      IF(LCON(LG1,J) .EQ. LG) I1 = J                                    ROU01370
 34   CONTINUE                                                          ROU01380
      GO TO 35                                                          ROU01390
 343   DO 345 J= 2,3                                                    ROU01400
      LG1 = LCON(LG,J)                                                  ROU01410
      IF(LG .EQ. R(LG1)) I1 = J                                         ROU01420
 345  CONTINUE                                                          ROU01430
 35   GCR(K) = VCR(IG,IORE,I1)/(C(LG)+CG(LZ,IG))                       ROU01440
      GO TO 37                                                          ROU01450
 36   CONTINUE                                                          ROU01460
 37   CONTINUE                                                          ROU01470
 38   CONTINUE                                                          ROU01480
      DO 385 K=1,NK                                                     ROU01490
 385  GCRT = GCRT + GCR(K)                                              ROU01500
      DO 39 K = 1,4                                                     ROU01510
      VEHS(K) = GCR(K)/GCRT * ZZVEH                                     ROU01520
      VEHS(K) = VEHS(K) / ITM                                           ROU01530
 39   CONTINUE                                                          ROU01540
C     PRINT 939, ZNAME(Z),ZNAME(IZ),(LISTG(K),LISTL(K),CLIST(K),VEHS(K) ROU01550
C    1, K=1,NK)                                                         ROU01560
C939  FORMAT (6H ZONE I2,9H TO ZONE I2,19H GATE LINK,COST,VEH   ,       ROU01570
C    1 4(2I3, F5.0, E10.3))                                             ROU01580
C                                                                       ROU01590
C                                                                       ROU01600
```

```
C  APPLY VEHICLES TO THE LINKS OF THE NEWWORK OF 4 POSSIBLE ROUTES      ROU01610
      DO 49 K=1,4                                                       ROU01620
C  RECOVER LAST LINK IN THE ROUTE                                       ROU01630
      LZ = LISTL(K)                                                     ROU01640
      IF(LZ .LE. 0) GO TO 49                                            ROU01650
      IG = LISTG(K)                                                     ROU01660
C  STORE THE NUMBER THAT TERMINATE ON THIS LINK BEFORE TURNING ST,PT,LT ROU01670
      COUNT(4,LZ) = COUNT(4,LZ) + VEHS(K)                               ROU01680
C  BACK UP ONE LINK                                                     ROU01690
 41   LG =  RG(LZ,IG)                                                   ROU01700
      IF (LG.LE. 0) GO TO 45                                            ROU01710
C  FIND OUT IF CONNECTION IS ST,RT,LT                                   ROU01720
      DO 42  J = 1,3                                                    ROU01730
      IF (LCON (LG,J) .NE. LZ) GO TO 42                                 ROU01740
      COUNT( J,LG) = COUNT(J,LG) + VEHS (K )                            ROU01750
      GO TO 43                                                          ROU01760
 42   CONTINUE                                                          ROU01770
      CALL EXIT                                                         ROU01780
 43   LZ = LG                                                           ROU01790
      GO TO 41                                                          ROU01800
C  LZ"S THE LAST LINK IN ROUTE TO THE GATE                              ROU01810
C  BACK UP ONE LINK                                                     ROU01820
 45   LG = R( LZ)                                                       ROU01830
      IF( LG .LE. 0) GO TO 49                                           ROU01840
C  FIND OUT IF CONNECTION IS ST,RT,LT                                   ROU01850
      DO 46  J=1,3                                                      ROU01860
      IF (LCON (LG,J).NE. LZ) GO TO 46                                  ROU01870
      COUNT (J, LG) = COUNT(J,LG) + VEHS(K )                            ROU01880
      GO TO 47                                                          ROU01890
 46   CONTINUE                                                          ROU01900
 47   LZ = LG                                                           ROU01910
      GO TO 45                                                          ROU01920
 49   CONTINUE                                                          ROU01930
 50   CONTINUE                                                          ROU01940
      IF(IT .EQ. ITM) GO TO 80                                          ROU01950
      DO 70 I=1,NINS                                                    ROU01960
      DO 60 J=1,4                                                       ROU01970
 60   QTEMP(J) = QUE(I,J)                                               ROU01980
      IF (ITYPC(I) .GT. 0) GO TO 61                                     ROU01990
      CALL INSECU                                                       ROU02000
```

```
      GO TO 62                                                          R0U02010
61    CALL INSEC                                                        R0U02020
62    DO 65 J=1,4                                                       R0U02030
      QUE(I,J) = QTEMP(J)                                               R0U02040
      L = LIN(I,J)                                                      R0U02050
      IF(L.LE.0) GO TO 65                                               R0U02060
      IF(VEL(L) .LE. 0.) GO TO 65                                       R0U02070
      DO 64 K=1,3                                                       R0U02080
64    TT(L,K) = DIST(L)/VEL(L) + DELA(L,K)                              R0U02090
65    CONTINUE                                                          R0U02100
70    CONTINUE                                                          R0U02110
80    CONTINUE                                                          R0U02120
      TF = STP                                                          R0U02130
      RETURN                                                            R0U02140
      END                                                               R0U02150
```

```
      SUBROUTINE MINPTH (LINKS, NL)                                     MIN00010
C  THIS ROUTINE COMPUTES THE MINIMUM TIME PATH    FROM NL ORIGIN LINKS  MIN00020
C  L TO ALL ACCESSABLE LINKS IN A TRAFFIC NETWORK. ARRAY C STORES THE   MIN00030
C  COST OR TRAVEL TIME TO EACH LINK AND ARRAY R STORES THE ROUTES FROM  MIN00040
C  EACH LINK TO L ON COMPLETION.                                        MIN00050
      COMMON / COMM/ I,Z,L,J, K, TOD, DOW, TP, TOTATT, TTPZ,TOTGEN,ITM  MIN00060
     1,NYEAR,LHEAD(18)                                                  MIN00070
      COMMON / LINK / NLINK, NLANE(200), X1(200), Y1(200),X2(200),Y2(200MIN00080
     1), CAP(200),DIST(200), VEL(200), LCON (200,3),NSTOPS(200)         MIN00090
      INTEGER R                                                         MIN00100
      COMMON TT(200,3),NC,NR,RO, R(200), CO, C(200)                     MIN00110
      DIMENSION LINKS(12)                                               MIN00120
      NC = NLINK                                                        MIN00130
      DO 15 L = 1, NC                                                   MIN00140
 15   C (L) = 1.0E10                                                    MIN00150
      DO 17 L = 1, NC                                                   MIN00160
 17   R (L) = 0                                                         MIN00170
      L=1                                                               MIN00180
      DO 19 I =1,NL                                                     MIN00190
      IF (LINKS(I) .LE.0) GO TO 19                                      MIN00200
      L = LINKS(I)                                                      MIN00210
      C (L) = -1.0                                                      MIN00220
 19   CONTINUE                                                          MIN00230
      GO TO 3                                                           MIN00240
 20   IF ( L.GE. NLINK) L = 0                                           MIN00250
      L = L +1                                                          MIN00260
      IF (L .EQ. LS ) GO TO 9                                           MIN00270
 25   IF (C(L) .GE. 0) GO TO 20                                         MIN00280
 3    LF = 0                                                            MIN00290
      DO  4 J =1,3                                                      MIN00300
      LC = LCON(L,J)                                                    MIN00310
      IF ( LC .EQ. 0) GO TO 4                                           MIN00320
      IF  (TT(L,J) - C(L) .GE. ABS(C(LC))) GO TO 4                      MIN00330
      C (LC) = C(L) - TT(L,J)                                           MIN00340
      R (LC) = L                                                        MIN00350
      LF =  J                                                           MIN00360
 4    CONTINUE                                                          MIN00370
      C (L) = -C(L)                                                     MIN00380
      LS = L                                                            MIN00390
      IF ( LF .EQ. 0) GO TO 20                                          MIN00400
```

```
      L = LCON(L, LF)                                                   MIN00410
      GO TO 3                                                           MIN00420
    9 CONTINUE                                                          MIN00430
C     PRINT 98,(R(L),C(L),L=1,NC)                                       MIN00440
      RETURN                                                            MIN00450
C98   FORMAT (* MFATH*/5(I5, F10.0))                                    MIN00460
      END                                                               MIN00470
```

```
      SUBROUTINE PARKNG                                                 PAR00010
C  THIS SUB-MODEL PREDICTS AVERAGE TRAVEL TIME TO VEHICLES ENTERING A   PAR00020
C  PARKING AREA OR ZONE.  AT AN AVERAGE SPEED OF 7.0 MPS VEHICLES MOVE  PAR00030
C  THROUGH THE PARKING ZONE. TRAVEL TIME (TT) IN THE ZONE IS PROPORTIONLPAR00040
C  TO THE PARKED VEHICLE TO ZONE CAPACITY RATIO.  QUEUEING OCCURS WHEN  PAR00050
C  HEADWAYS ARE LESS THAN THE TIME REQUIRED TO BACK OUT OF AND TO CLEAR PAR00060
C  A PARKING STALL, OR WHEN VEHICLES MOVING IN THE ZONE PLUS PARKED     PAR00070
C  VEHICLES EXCEEDS CAPACITY.                                           PAR00080
      COMMON / COMM/ I,Z,L,J, K, TOD, DOW, TP, TOTATT, TTPZ,TOTGEN      PAR00090
      COMMON /PARKZ/ PZ,       PV(50), PLA(50), VZ(50), ZV(50),PD,PLS,PLBPAR00100
     10, PNOS(50) , PLL(50)                                             PAR00110
      COMMON /ZONES/ NZONES, NZLINS(50), ZLINKS(12,50), ZATTR(50),      PAR00120
     1ZGENR (50), NEXT, ZAT(2), ZGT(2), ZVT(3,50), ZNAME(50)            PAR00130
      COMMON/EXCLUD/NOPARK                                              PAR00140
C  VZ ARE ZONE ATTRACTIONS AND ZV ARE ZONE GENERATIONS                  PAR00150
      INTEGER  Z, PZ                                                    PAR00160
      REAL NVL                                                          PAR00170
C  COMPUTE THE CAPACITY OF THE PARKING ZONE                             PAR00180
      Z = PZ                                                            PAR00190
      PC = PLA(PZ) / PD                                                 PAR00200
C  COMPUTE MAXIMUM PARKING LOT DISCHARGE RATE                           PAR00210
      PLDR = AMAX1(0.5,PC/2400.)                                        PAR00220
C  COMPUTE THE TOTAL DRIVEABLE LENGTH OF THE PARKING ZONE               PAR00230
C   STALLS ASSUMED 3 METERS IN WIDTH AND AT 45DEGREE ANGLE              PAR00240
C     PLL= PC *3.0/ 1.414                                               PAR00250
      PL = PLL(PZ)                                                      PAR00260
C  COMPUTE THE PARKED VEHICLES AT END OF TIME PERIOD                    PAR00270
      PVZ =AMAX1 (    VZ( Z) - ZV( Z) +PV(PZ) ,0.0)                     PAR00280
C  COMPUTE PARKING UTILIZATION                                          PAR00290
      PU = PVZ / PC                                                     PAR00300
C   COMPUTE SPEED BASED ON GCA VALIDATION STUDY OF CO MODEL AUG 74 P27  PAR00310
       PLSS = PLS                                                       PAR00320
      PLS = 21.0-11.3*PU                                                PAR00330
      PLS = AMAX1 (10.75,PLS)                                           PAR00340
C  ASSIGN A PARGING LOT SPEED BASED ON TACOMA STUDY                     PAR00350
      PLS = 9.3                                                         PAR00360
C  THE TRAVEL TIME FOR ARRIVING VEHICLES                                PAR00370
C     IF(PU.LT. .99) GO TO 21                                           PAR00380
C     TTA = PU * PL/PLS                                                 PAR00390
C     GO TO 25                                                          PAR00400
```

```
C21   TTA =  (PVZ + PV(PZ))/2.0 /PC *PL /PLS                          PAR00410
C  DEPARTURES EXPERIENCE DIFFERENT TRAVEL TIME                         PAR00420
C25   TTA = AMAX1 (TTA,PL/2.0/PLS)                                    PAR00430
C     TTA = AMAX1(TTA, PLA(PZ) / PL / PLS)                            PAR00440
      TTA = AMAX1(TTA, PLA(PZ) / PL / PLS)                            PAR00450
      TTA = PL/PLS                                                    PAR00460
      TTD = TTA + PLBO                                                PAR00470
C                                                                     PAR00480
C  QUEUEING                                                           PAR00490
C                                                                     PAR00500
C   NVL I S THE NUMBER OF VEHICLES DRIVING THRU THE LOT AT ANY TIME   PAR00510
      DQ = 0.0                                                        PAR00520
      NVL =(TTA* VZ(Z) + TTD* ZV(Z))/TP + .000278                     PAR00530
      IF(NVL + PVZ - PC .GE. 0.0) GO TO 33                            PAR00540
      IF(PU.GE. 0.85) GO TO 33                                        PAR00550
C  CHECK HEADWAYS AGAINST TIME TO BACK OUT OF STALL                   PAR00560
C  THERE IS NO QUEUEING.                                              PAR00570
      AWQ= 0.0                                                        PAR00580
      GO TO 39                                                        PAR00590
C  COMPUTE THE DELAY CAUSED BY WAITING FOR VEHICLES TO BACK OUT OF STALSPAR00600
 33   IF(VZ(Z) .GT. 0.0)                                              PAR00610
     1DQ =AMAX1 ( PLBO* PLBO/(2.0*(TP/ VZ(Z) -1.0/PLDR)),PLBO/2.0)*   PAR00620
     1 (PU-.85)/0.15                                                  PAR00630
C  COMPUTE THE DELAY CAUSED BY EXCEEDENCE OF PARKING ZONE CAPACITY    PAR00640
      AWQ = 0.0                                                       PAR00650
      IF ( PVZ + PV(PZ) + NVL .LE. PC*2.0) GO TO 35                   PAR00660
      AWQ =((PVZ + PV(PZ))/2.0 + NVL - PC)* TP/(ZV(Z) + 1.0)          PAR00670
      GO TO 39                                                        PAR00680
 35   IF ( PVZ .GE. PC)  GO TO 37                                     PAR00690
      IF ( PV(PZ) .LT. PC ) GO TO 39                                  PAR00700
C  EXISTING QUEUE IS DISSIPATING                                      PAR00710
      AWQ = (PV(PZ) - PC +NVL)**2/(ZV(Z)-VZ(Z))*.5*TP/ZV(Z)          PAR00720
      GO TO 39                                                        PAR00730
C  ARRIVALS ARE BUILDING A QUEUE                                      PAR00740
 37   AWQ = (PVZ-PC+NVL)**2 / (VZ(Z)-ZV(Z))*.5*TP/(ZV(Z) + 1.)       PAR00750
 39   CONTINUE                                                        PAR00760
C     ADD A VEHICLE IDLEING TIME IF NO QUEUE EXISTS                   PAR00770
C     THIS INCLUDES COSTS TO ENTER THE ZONE BASED ON TACOMA STUDY     PAR00780
      AWQ = AMAX1(AWQ,0.67*TTA)                                       PAR00790
      PLS = PLSS                                                      PAR00800
```

```
C  TOTAL THE RUNNING TIME IN PARKING ZONE                                PAR00810
      TTPZ=(TTA + DQ + AWQ) * VZ(Z)  + (TTD + AWQ) * ZV(Z)               PAR00820
C     IF(PZ .EQ. NEXT+1) PRINT 97                                        PAR00830
 97   FORMAT (1H-15X, 4HZONE,8X,10HTOTAL TIME 3X 7HTT ARRV,3X 7HTT DEPT,PAR00840
     12X8HBACKNG Q,3X 7HQ DELAY,3X 7HDEPARTS,3X 7HARRVALS,4X6HLENGTH)   PAR00850
C     IF(NOPARK.EQ.1) PRINT 971                                          PAR00860
  971 FORMAT(//,1X,'THE FOLLOWING INFORMATION CONCERNING PARKING ZONES  PAR00870
     &DOES NOT APPLY SINCE PARKING LOT EMISSIONS ARE NOT CONSIDERED',/) PAR00880
C     PRINT 98, ZNAME(Z), TTPZ, TTA, TTD, DQ, AWQ, ZV(Z),VZ(Z), PL      PAR00890
   98 FORMAT(1H ,' PARKING ',7X,I3,16X,8E10.3)                          PAR00900
      RETURN                                                            PAR00910
      END                                                               PAR00920
```

```
      SUBROUTINE COORXY                                                 C0000010
C     PURPOSE IS TO ORGANIZE THE X,Y COORDINATES OF EACH LINK SO THAT   C0000020
C     X1, Y1 ARE UPSTREAM AND X2, Y2 ARE THE DOWNSTREAM COORDINATES.    C0000030
C     ALSO CONVERT COORDINATES FROM FEET TO METERS IN THIS WRITING      C0000040
C     OF THE SUBROUTINE.                                                C0000050
      COMMON /LINK/ NLINK, NLANE(200), X1(200), Y1(200),X2(200),Y2(200) C0000060
     1  , LCAP(200), DIST(200), VEL(200), LCON(200,3)                   C0000070
      DATA FOTMET / 0.3048 /                                            C0000080
C                                                                       C0000090
      DO 900 L = 1,NLINK                                                C0000100
         DO 100 J = 1,3                                                 C0000110
         IF(LCON(L,J) .EQ. 0) GO TO 100                                 C0000120
         LC = LCON(L,J)                                                 C0000130
         GO TO 200                                                      C0000140
 100     CONTINUE                                                       C0000150
      GO TO 900                                                         C0000160
C                                                                       C0000170
C     COMPUTE DISTANCES                                                 C0000180
 200  CONTINUE                                                          C0000190
      DIST1 = (X1(L) - X1(LC))**2 + (Y1(L) - Y1(LC))**2                 C0000200
      DIST2 = (X1(L) - X2(LC))**2 + (Y1(L) - Y2(LC))**2                 C0000210
      DIST3 = (X2(L) - X1(LC))**2 + (Y2(L) - Y1(LC))**2                 C0000220
      DIST4 = (X2(L) - X2(LC))**2 + (Y2(L) - Y2(LC))**2                 C0000230
C                                                                       C0000240
      LC1 = 0                                                           C0000250
      IF(DIST1 - DIST2)400,250,250                                      C0000260
 250  CONTINUE                                                          C0000270
      IF(DIST2 - DIST3)290,300,300                                      C0000280
 280  CONTINUE                                                          C0000290
      IF(DIST2 - DIST4)290,300,300                                      C0000300
 290  CONTINUE                                                          C0000310
      LC1 = 1                                                           C0000320
      GO TO 700                                                         C0000330
 300  CONTINUE                                                          C0000340
      IF(DIST3 - DIST4)850,850,750                                      C0000350
 400  CONTINUE                                                          C0000360
      IF(DIST1 - DIST3)450,450,500                                      C0000370
 450  CONTINUE                                                          C0000380
      IF(DIST1 - DIST4)700,700,500                                      C0000390
 500  CONTINUE                                                          C0000400
```

```
      IF(DIST3 - DIST4)850,850,750                                      C0000410
  700 CONTINUE                                                          C0000420
      XTEMP = X1(L)                                                     C0000430
      X1(L) = X2(L)                                                     C0000440
      X2(L) = XTEMP                                                     C0000450
      YTEMP = Y1(L)                                                     C0000460
      Y1(L) = Y2(L)                                                     C0000470
      Y2(L) = YTEMP                                                     C0000480
      IF(LC1)750,850,750                                                C0000490
  750 CONTINUE                                                          C0000500
      XTEMP = X1(LC)                                                    C0000510
      X1(LC) = X2(LC)                                                   C0000520
      X2(LC) = XTEMP                                                    C0000530
      YTEMP = Y1(LC)                                                    C0000540
      Y1(LC) = Y2(LC)                                                   C0000550
      Y2(LC) = YTEMP                                                    C0000560
  850 CONTINUE                                                          C0000570
  900 CONTINUE                                                          C0000580
      DO 910 L=1,NLINK                                                  C0000590
C     CONVERT X1 X2 Y1 Y2 FROM FEET TO METERS                            C0000600
      X1(L) = X1(L) * FOTMET                                            C0000610
      X2(L) = X2(L) * FOTMET                                            C0000620
      Y1(L) = Y1(L) * FOTMET                                            C0000630
      Y2(L) = Y2(L) * FOTMET                                            C0000640
  910 CONTINUE                                                          C0000650
      RETURN                                                            C0000660
      END                                                               C0000670
```

```
      SUBROUTINE EMIT                                                   EMI00010
C                                                                       EMI00020
C DETERMINE EMISSIONS OF HC,CO, CO2, NOX                                EMI00030
C                                                                       EMI00040
      REAL MISS                                                         EMI00050
      DIMENSION DQ(200)                                                 EMI00060
      DIMENSION TIME(4), EMISS(200,4)                                   EMI00070
      DIMENSION PRT(50)                                                 EMI00080
       COMMON / COMM / I,Z,L,J,K,TOD,DOW,TP,TOTATT,TTPZ,TOTGEN,ITM      EMI00090
     1,NYEAR,LHEAD(18),TUNFAC                                           EMI00100
      COMMON / LINK/ NLINK, NLANE(200), X1(200), Y1(200), X2(200),Y2(200EMI00110
     1),LCAP(200),DIST(200), VEL(200), LCON(200,3) ,NSTOPS(200)         EMI00120
      COMMON /ZONES/ NZONES, NZLINS(50), ZLINKS(12,50), ZATTR(50),      EMI00130
     1ZGENR (50), NEXT, ZAT(2), ZGT(2), ZVT(2,50), ZNAME(50)            EMI00140
      COMMON / VOLUME/ COUNT(4,200)                                     EMI00150
      COMMON /PARKZ/ PZ,      PV(50), PLA(50), VZ(50), ZV(50),PD,PLS,PLBEMI00160
     10, PNOS(50) , PLL(50)                                             EMI00170
      COMMON TT(200,3), DELA(200,3),  QZ(50), QUEUE(200)                EMI00180
     1      ,Q(200,4),DLINK(200,4),CCOUNT(200),MISS(200,4),CMISS(200,4) EMI00190
       INTEGER Z                                                        EMI00200
      COMMON/ELINK/IREG,EPARAM(10),IFLAG(3)                             EMI00210
      DIMENSION ECRU(20),ECRUZ(20)                                      EMI00220
      DATA I0,I1/0,1/                                                   EMI00230
      DATA R75/75.0/                                                    EMI00240
      DATA R5/5.0/                                                      EMI00250
      DATA R0,R1/0.0,1.0/                                               EMI00260
      DATA I77/77/                                                      EMI00270
      EQUIVALENCE (QUEUE,DQ),(PRT,QZ)                                   EMI00280
      EQUIVALENCE (Q,EMISS)                                             EMI00290
C                                                                       EMI00300
C     SET UP MATRIX OF EMISSION CORRECTION FACTORS BY SPEED             EMI00310
C     FOR THE SCENARIO SPECIFIED IN ELINK                               EMI00320
C                                                                       EMI00330
      IYEAR=IFIX(NYEAR/10000.)                                          EMI00340
      WRITE (6,955)IREG,IYEAR,(EPARAM(IJ1),IJ1=1,10)                    EMI00350
  955 FORMAT('1EMISSIONS SUBMODEL'//' ------------------'//             EMI00360
     &/'REGION =',I3,                                                   EMI00370
     &/'YEAR=',I3,                                                      EMI00380
     &/'TEMP,COLD,HOT,CCOLD=',4F10.3,                                   EMI00390
     &/'VMTMIX=',6F10.3/)                                               EMI00400
```

```
C                                                                       EMI00410
C     CALL SUP8 TO CALCULATE BASE YEAR LDV EMISSIONS                    EMI00420
C         (REPLACED BY DATA STATEMENTS)                                 EMI00430
C                                                                       EMI00440
C     CALL SUP8(I1,I1,I77,R5,R75,R0,R0,R0,R1,R0,R0,R0,R0,R0,I0,I0,I0,   EMI00450
C    &ECRU(1),EIDL77)                                                   EMI00460
C     DO 5 IS=2,20,1                                                    EMI00470
C     S=FLOAT(IS)*3.0+2.0                                               EMI00480
C   5 CALL SUP8(I0,I1,I77,S,R75,R0,R0,R0,R1,R0,R0,R0,R0,R0,I0,I0,I0,    EMI00490
C    &ECRU(IS),EIDL77)                                                  EMI00500
      DATA EIDL77/25.17/                                                EMI00510
      DATA ECRU/108.83,116.55,83.22,65.51,54.81,47.49,41.88,            EMI00520
     &37.25,33.29,29.33,27.16,24.98,23.28,22.28,21.58,21.06,            EMI00530
     &20.39,19.11,16.80,15.78/                                          EMI00540
C                                                                       EMI00550
C     CALL SUP8 TO CALCULATE EMISSIONS FOR SCENARIO AND                 EMI00560
C     TO COMPUTE EMISSION CORRECTION FACTORS AT SPEED IS                EMI00570
C                                                                       EMI00580
      CALL SUP8(I1,IREG,IYEAR,R5,EPARAM(1),EPARAM(2),EPARAM(3),         EMI00590
     &EPARAM(4),EPARAM(5),EPARAM(6),EPARAM(7),EPARAM(8),EPARAM(9),      EMI00600
     &EPARAM(10),IFLAG(1),IFLAG(2),IFLAG(3),ECRUZ(1),EIDL)              EMI00610
      DO 6 IS=2,20,1                                                    EMI00620
      S=FLOAT(IS)*3.0+2.0                                               EMI00630
      CALL SUP8(I0,IREG,IYEAR,S,EPARAM(1),EPARAM(2),EPARAM(3),          EMI00640
     &EPARAM(4),EPARAM(5),EPARAM(6),EPARAM(7),EPARAM(8),EPARAM(9),      EMI00650
     &EPARAM(10),IFLAG(1),IFLAG(2),IFLAG(3),ECRUZ(IS),EIDL)             EMI00660
    6 ECRUZ(IS)=ECRUZ(IS)/ECRU(IS)                                      EMI00670
      ECRUZ(1)=ECRUZ(1)/ECRU(1)                                         EMI00680
      CALL INITMM                                                       EMI00690
      ITEST=0                                                           EMI00700
C                                                                       EMI00710
C     CHECK IF SUPPLEMENTAL EMISSION PARAMETERS ARE CALLED FOR          EMI00720
C     IF SO, CORRECT IDLE EMISSIONS AND PRINT MESSAGE                   EMI00730
C                                                                       EMI00740
      DO 7 I=1,3                                                        EMI00750
    7 ITEST=IFLAG(I)+ITEST                                              EMI00760
      IF (ITEST.NE.0) EIDL=EIDL77*ECRUZ(1)                              EMI00770
      IF (ITEST.NE.0) WRITE (6,8)                                       EMI00780
    8 FORMAT(' CAUTION: IM,TRUCK,OR ALH CORRECTION CALLED FOR'/         EMI00790
     & ' IDLE EMISSIONS CALCULATED BY ALTERNATE METHOD')                EMI00800
```

```
      EIDL=EIDL/60.                                                     EMI00810
C                                                                       EMI00820
      WRITE(6,987) ECRUZ                                                EMI00830
  987 FORMAT(1H0,'EMISSION CORRECTION'/4(1X,5F10.6/))                   EMI00840
C                                                                       EMI00850
C  ACC RATE =3.41 MI/HP/SEC  OF 5 FT/SEC/SEC                            EMI00860
      ACC=3.41                                                          EMI00870
      ACCR=5.0                                                          EMI00880
      PRINT 110                                                         EMI00890
  110 FORMAT(10H-EMISSIONS/131H LINK NSTOPS  QLENGTH      COUNT      SPEEMI00900
     1ED  E AT SPEED  DISTANCE    E DECEL   DISTANCE     E IDLE   DISTANEMI00910
     2CL    E ACCEL   DISTANCE  )                                      EMI00920
      DO 100 KP=1,NLINK                                                 EMI00930
      CCOUNT(KP)=COUNT(1,KP) + COUNT(2,KP)+ COUNT(3,KP)                 EMI00940
      IF(CCOUNT(KP).LE.0.0)CCOUNT(KP)=COUNT(4,KP)                       EMI00950
      CAPL = 1200.0*NLANE(KP)                                           EMI00960
      IF(VEL(KP) .GT.70) CAPL = CAPL * 1.5                              EMI00970
      V = VEL(KP)*(1.0-0.5*CCOUNT(KP)/CAPL)                             EMI00980
      S=V*3600.0/5280.0                                                 EMI00990
      TT1=DIST(KP)/V                                                    EMI01000
      IF(NSTOPS(KP).LE.00) GO TO 19                                     EMI01010
      TT2=(TT(KP,1)*CCOUNT(KP)-(CCOUNT(KP)-NSTOPS(KP))*TT1)/NSTOPS(KP)  EMI01020
      TIME(2)=V/ACCR                                                    EMI01030
      TIME(4)=TIME(2)                                                   EMI01040
      DIS1=TIME(2)*V/2.                                                 EMI01050
      DISDC=DIST(KP)-DQ(KP)-2.*DIS1                                     EMI01060
      IF(DISDC.GE.0.0) GO TO 3                                          EMI01070
      DIS1=AMAX1(DIS1+DISDC/2.,0.0)                                     EMI01080
      DISDC=0.0                                                         EMI01090
    3 DIS2=DIS1                                                         EMI01100
      TIME(1)=DISDC/V                                                   EMI01110
      TIME(3)=TT2-TIME(1)-2.*TIME(2)                                    EMI01120
      IF (TIME(3).GE.0.0) GO TO 5                                       EMI01130
      TIME(2)=AMAX1(TIME(2)+TIME(3)/2.,0.0)                             EMI01140
      TIME(3)=0.0                                                       EMI01150
      TIME(4)=TIME(2)                                                   EMI01160
    5 CONTINUE                                                          EMI01170
      GO TO 22                                                          EMI01180
   19 CONTINUE                                                          EMI01190
      IF(DQ(KP).LE.0.0) GO TO 24                                        EMI01200
```

```
      TT3= TT1-TT(KP,1)                                                 EMI01210
      IF(TT3.LE.0.0) GO TO 25                                           EMI01220
      DIS1=TT3/2.*V/2.                                                  EMI01230
      DISDC=DIST(KP)-2.*DIS1                                            EMI01240
      IF(DISDC .GE. 0.0) GO TO 4                                        EMI01250
      DIS1 = AMAX1 (DIS1+DISDC/2. 0.0)                                  EMI01260
      DISDC = 0.0                                                       EMI01270
    4 DIS2 = DIS1                                                       EMI01280
      DQ(KP)=0.0                                                        EMI01290
      TIME(1)=DISDC/V                                                   EMI01300
      TIME(3)=0.0                                                       EMI01310
      TIME(2)=(TT(KP,1)+TT3-TIME(1))/2.                                 EMI01320
      TIME(4)=TIME(2)                                                   EMI01330
      GO TO 22                                                          EMI01340
   25 CONTINUE                                                          EMI01350
      DISDC=DIST(KP)-DQ(KP)                                             EMI01360
      DISDC=AMAX1(0.0,DISDC)                                            EMI01370
      DIS1=DQ(KP)/2.                                                    EMI01380
      DIS2=DIS1                                                         EMI01390
      DQ(KP)=0.0                                                        EMI01400
      TIME(1)=DISDC/V                                                   EMI01410
      TIME(2)=(TT(KP,1)-TIME(1))/2.                                     EMI01420
      TIME(3)=0.0                                                       EMI01430
      TIME(4)=TIME(2)                                                   EMI01440
      GO TO 22                                                          EMI01450
   24 CONTINUE                                                          EMI01460
      DIS1=0.0                                                          EMI01470
      DIS2=DIS1                                                         EMI01480
      DISDC=DIST(KP)                                                    EMI01490
      TIME(1)=DISDC/V                                                   EMI01500
      TIME(2)=0.0                                                       EMI01510
      TIME(3)=TIME(2)                                                   EMI01520
      TIME(4)=TIME(2)                                                   EMI01530
   22 CONTINUE                                                          EMI01540
C                                                                       EMI01550
C     COMPUTE EMISSIONS FOR THOSE NOT DELAYED                           EMI01560
C     PLUS CONSTANT CRUISE FOR THOSE DELAYED.                           EMI01570
C                                                                       EMI01580
      IS=IFIX((S-0.5)/3.0)                                              EMI01590
      IF(IS.LT.1) IS=1                                                  EMI01600
```

```
      IF(IS.GT.20) IS=20                                                EMI01610
      ISS=IFIX((S/2.0-0.5)/3.0)                                         EMI01620
      IF(ISS.LT.1) ISS=1                                                EMI01630
      TV=((COUNT(KP)-NSTOPS(KP))*TT1/DIST(KP)                           EMI01640
      TV1=NSTOPS(KP)*TIME(1)                                            EMI01650
      CALL CRUZ(S,ATOT)                                                 EMI01660
      ATOT=ATOT*ECRUZ(IS)                                               EMI01670
      MISS(KP,1)=ATOT*TV*DISDC                                          EMI01680
      MISS(KP,2)=ATOT*TV*DIS1                                           EMI01690
      MISS(KP,3)=ATOT*TV*DQ(KP)                                         EMI01700
      MISS(KP,4)=ATOT*TV*DIS2                                           EMI01710
      CMISS(KP,1)=ATOT*TV1                                              EMI01720
C                                                                       EMI01730
C   COMPUTE EMISSIONS DURING STOPPED TIME FOR DELAYED CARS              EMI01740
C                                                                       EMI01750
      TV=NSTOPS(KP)*TIME(3)                                             EMI01760
      CMISS(KP,3)=TV*EIDL                                               EMI01770
C                                                                       EMI01780
C  COMPUTE EMISSIONS DURING DECELERATION AND ACCELERATION               EMI01790
C                                                                       EMI01800
      TV=NSTOPS(KP)*TIME(2)                                             EMI01810
      CALL ACDC(S,ACC,ACTEMP)                                           EMI01820
      CALL ACDC(S,ACC*(-1.0),DCTEMP)                                    EMI01830
      CMISS(KP,4)=ACTEMP*TV*ECRUZ(ISS)                                  EMI01840
      CMISS(KP,2)=DCTEMP*TV*ECRUZ(ISS)                                  EMI01850
C                                                                       EMI01860
C   DETERMINE AND STORE DISTANCE FOR EACH SUB-LINK,FEET)                EMI01870
C                                                                       EMI01880
      DLINK(KP,4) = DIS2 * 0.3048                                       EMI01890
      DLINK(KP,3)=DQ(KP)*0.3048                                         EMI01900
      DLINK(KP,2) = DIS1*0.3048                                         EMI01910
      DLINK(KP,1) = DISDC*0.3048                                        EMI01920
C                                                                       EMI01930
C   COMPUTE EMISSION RATE FOR EACH PORTION OF THE LINK IN GRAMS/M/SEC   EMI01940
C                                                                       EMI01950
      DO 80 J=1,4                                                       EMI01960
      EMISS(KP,J)=0.0                                                   EMI01970
      IF(DLINK(KP,J).NE.0.0)                                            EMI01980
     1EMISS(KP,J)=(MISS(KP,J)+CMISS(KP,J))/(DLINK(KP,J)*TP)             EMI01990
   80 CONTINUE                                                          EMI02000
```

```
      PRINT 111, KP,NSTOPS(KP),DQ(KP), CCOUNT(KP),V,(EMISS(KP,JM),      EMI02010
     1DLINK(KP,JM), JM =1,4)                                            EMI02020
 111  FORMAT (2I5,3F11.4, 4(5I2.4,E10.4))                              EMI02030
  100 CONTINUE                                                         EMI02040
C  EMMISSIONS RATES ARE COMPUTED FOR EACH ZONE IN GRAMS/SEC            EMI02050
      DO 75 IZ = 1,NZONES                                              EMI02060
       IF(IZ .LE. NEXT) GO TO 73                                       EMI02070
C  COMPUTE A RATE OF EMISSIONS FROM THE ZONE                           EMI02080
      PRT(IZ) = PRT(IZ) * EIDL/TP                                      EMI02090
      GO TO 75                                                         EMI02100
C  EMISSION RATES ARE READ INTO PLA FOR EXTERIOR ZONES.                EMI02110
 73   PRT(IZ) = PLA(IZ)                                                EMI02120
 75   CONTINUE                                                         EMI02130
      RETURN                                                           EMI02140
      END                                                              EMI02150
```

```
      SUBROUTINE DISPER                                                 DIS00010
C     THIS ROUTINE CONTROLS THE DISPERSION COMPUTATIONS                 DIS00020
      COMMON/DISP/XO(200),YO(200),NUMREC,THETA(24),WS(24),YCOEFF(24)    DIS00030
     1,YEXP(24),YCONSL,YCONSZ,ZCOEFF(24),ZEXP(24),ZCONST,GRDSIZ,BCKG    DIS00040
     2BL,MIX,ISTAB(24),SLAT,ICLD(24),NMET,NY,ITIME,HEIGHT(200)          DIS00050
      COMMON/LINK/NLINK,NLANE(200),X1(200),Y1(200),X2(200),Y2(200),LCAP(DIS00060
     1200),DIST(200),VEL(200),LCON(200,3),NSTOPS(200)                   DIS00070
      COMMON / COMM/ I,Z,L,J, K, TOD, DOW, TP, TOTATT, TTPZ,TOTGEN,ITM  DIS00080
     1,NYEAR,LHEAD(18), TUNFAC                                          DIS00090
      COMMON /ZONES/ NZONES, NZLINS(50), ZLINKS(12,50), ZATTR(50),      DIS00100
     1ZGENR (50), NEXT, ZAT(2), ZGT(2), ZVT(3,50), ZNAME(50)            DIS00110
      COMMON /MAINN/M                                                   DIS00120
      COMMON/CONCTR/CONC(200)                                           DIS00130
      CALL WIND                                                         DIS00140
      CALL LSLOPE                                                       DIS00150
      IF(NMET.EQ.0) GO TO 4                                             DIS00160
      CALL STABLE                                                       DIS00170
 4    DO 3 I=1,NLINK                                                    DIS00180
      N=X1(I)*10.+0.5                                                   DIS00190
      X1(I)=N/10.                                                       DIS00200
      N=X2(I)*10.+0.5                                                   DIS00210
      X2(I)=N/10.                                                       DIS00220
      N=Y1(I)*10.+0.5                                                   DIS00230
      Y1(I)=N/10.                                                       DIS00240
      N=Y2(I)*10.+0.5                                                   DIS00250
 3    Y2(I)=N/10.                                                       DIS00260
      DO 2 M=1,NUMREC                                                   DIS00270
      CONC(M) = 0.0                                                     DIS00280
 2    CALL BASIC                                                        DIS00290
      IF(NZONES.EQ.0.) GO TO 14                                         DIS00300
      CALL GENGSC                                                       DIS00310
      CALL ZPORDR                                                       DIS00320
C     MULTIPLY CONCENTRATION BY PPM CONVERSION                          DIS00330
C     FACTOR,DIVIDE BY PI AND WIND SPEED, AND ADD BACKGROUND CONCTRN.   DIS00340
 14   NN=NY-70                                                          DIS00350
      DO 24 M=1,NUMREC                                                  DIS00360
   24 CONC(M)=((CONC(M)*273.)/(3.1415926*WS(ITIME)))                    DIS00370
     8+BCKGRD                                                           DIS00380
      RETURN                                                            DIS00390
      END                                                               DIS00400
```

```
      SUBROUTINE WIND                                                   WIN00010
C     THIS ROUTINE CONVERTS METEOROLOGICAL WIND DIRECTION TO TRIGONO-   WIN00020
C     METRIC DIRECTION, WITH ZERO DEGREES ON POSITIVE X-AXIS AND        WIN00030
C     PROCEEDING COUNTERCLOCKWISE.                                      WIN00040
      COMMON/DISP/XO(200),YO(200),NUMREC,THETA(24),WS(24),YCOEFF(24)    WIN00050
     1,YEXP(24),YCONSL,YCONSZ,ZCOEFF(24),ZEXP(24),ZCONST,GRDSIZ,BCKG    WIN00060
     2RD,MIX,ISTAB(24),SLAT,ICLD(24),NMET,NY,ITIME,HEIGHT(200)          WIN00070
      COMMON/WND/THETAP,AA,AB                                           WIN00080
      THETAP=450.-THETA(ITIME)                                          WIN00090
      IF(THETAP.GT.360.) THETAP=THETAP-360.                             WIN00100
      EPSLON=(THETAP-90.)*6.2831853/360.                                WIN00110
      AA=SIN(EPSLON)                                                    WIN00120
      AB=COS(EPSLON)                                                    WIN00130
      RETURN                                                            WIN00140
      END                                                               WIN00150
```

```
      SUBROUTINE LSLOPE                                                 LSL00010
C     THIS ROUTINE COMPUTES THE SLOPE AND Y-INTERCEPT OF EACH LINK      LSL00020
      COMMON/DISP/XO(200),YO(200),NUMREC,THETA(24),WS(24),YCOEFF(24)    LSL00030
     1,YEXP(24),YCONSL,YCONSZ,ZCOEFF(24),ZEXP(24),ZCONST,GRDSIZ,BCKG    LSL00040
     2PL,MIX,ISTAB(24),SLAT,ICLD(24),NMET,NY,ITIME,HEIGHT(200)          LSL00050
      COMMON/LINK/NLINK,NLANE(200),X1(200),Y1(200),X2(200),Y2(200),LCAP(LSL00060
     12(0),DIST(200),VEL(200),LCON(200,3),NSTOPS(200)                   LSL00070
      COMMON/LSLOP/AM(200),B(200)                                       LSL00080
      DO 1 I=1,NLINK                                                    LSL00090
C     CHECK TO SEE IF SLOPE IS INFINITE.  IF SO, SET SLOPE AND Y-INTER- LSL00100
C     CEPT EQUAL TO ZERO.  THIS WILL BE A CUE FOR VERTICAL LINKS.       LSL00110
      IF(X1(I).EQ.X2(I)) GO TO 2                                        LSL00120
      AM(I)=(Y1(I)-Y2(I))/(X1(I)-X2(I))                                 LSL00130
      B(I)=Y1(I)-AM(I)*X1(I)                                            LSL00140
      GO TO 1                                                           LSL00150
    2 AM(I)=0.                                                          LSL00160
      B(I)=0.                                                           LSL00170
    1 CONTINUE                                                          LSL00180
      RETURN                                                            LSL00190
      END                                                               LSL00200
```

```
      SUBROUTINE STABLE                                                 STB00010
C     THIS ROUTINE DETERMINES THE STABILITY AND HENCE THE DISPERSION    STB00020
C     PARAMETERS FOR EACH HOUR OF THE DAY                               STB00030
      DIMENSION SINELV(24),ISC(6,5),YC(5),YE(5),ZC(5),ZE(5)             STB00040
      COMMON/DISP/XO(200),YO(200),NUMREC,THETA(24),WS(24),YCOEFF(24)    STB00050
     1,YEXP(24),YCONSL,YCONSZ,ZCOEFF(24),ZEXP(24),ZCONST,GRDSIZ,BCKG    STB00060
     2RD,MIX,ISTAB(24),SLAT,ICLD(24),NMET,NY,ITIME,HEIGHT(200)          STB00070
      COMMON / COMM/ I,Z,L,J, K, TOD, DOW, TP, TOTATT, TTPZ,TOTGEN,ITM  STB00080
     1,NYEAR,LHEAD(18), TUNFAC                                          STB00090
      [illegible]                                                       [illegible]
     1,4/                                                               STB00110
      DATA YC/0.40,0.40,0.36,0.32,0.31/,YE/0.91,0.91,0.86,0.78,0.71/,ZC/STB00120
     1[illegible]/1.[illegible],1.26,1.13,1.00,0.85/                    STB00130
C     CALCULATE SOLAR ELEVATION ANGLE,INSOLATION,AND HOURS OF SUNRISE   STB00140
C     AND SUNSET                                                        STB00150
      [illegible]
      [illegible]
      [illegible]
```

```
      AINSOL=(1.-0.5*ICLD(IHR)/10.)*SINELV(IHR)                         STB00410
      IF(AINSOL.LE.0.33) GO TO 21                                       STB00420
      IF(AINSOL.LE.0.67) GO TO 22                                       STB00430
      IS=1                                                              STB00440
      GO TO 30                                                          STB00450
   21 IS=3                                                              STB00460
      GO TO 30                                                          STB00470
   22 IS=2                                                              STB00480
      GO TO 30                                                          STB00490
 23   IF(ICLD(IHR).GE.5) GO TO 24                                       STB00500
      IS=6                                                              STB00510
      GO TO 30                                                          STB00520
   24 IS=5                                                              STB00530
      GO TO 30                                                          STB00540
 25   IS=4                                                              STB00550
C     DETERMINE WIND SPEED CLASS                                        STB00560
 30   IF(WS(IHR).LE.3.) GO TO 31                                        STB00570
      IF(WS(IHR).LE.6.) GO TO 32                                        STB00580
      IF(WS(IHR).LE.10.) GO TO 33                                       STB00590
      IF(WS(IHR).LE.13.) GO TO 34                                       STB00600
      IW=5                                                              STB00610
      GO TO 40                                                          STB00620
   31 IW=1                                                              STB00630
      GO TO 40                                                          STB00640
   32 IW=2                                                              STB00650
      GO TO 40                                                          STB00660
   33 IW=3                                                              STB00670
      GO TO 40                                                          STB00680
 34   IW=4                                                              STB00690
C     DETERMINE STABILITY                                               STB00700
 40   ISTAB(IHR)=ISC(IS,IW)                                             STB00710
C     CONVERT WIND SPEED IN KNOTS TO METERS PER SECOND                  STB00720
 41   WS(IHR)=WS(IHR)*0.514791                                          STB00730
      IF(WS(IHR).LT.1.) WS(IHR)=1.                                      STB00740
C     DETERMINE DIFFUSION PARAMETERS                                    STB00750
      KK=ISTAB(IHR)                                                     STB00760
      YCOEFF(IHR)=YC(KK)                                                STB00770
      YEXP(IHR)=YE(KK)                                                  STB00780
      ZCOEFF(IHR)=ZC(KK)                                                STB00790
      ZEXP(IHR)=ZE(KK)                                                  STB00800
```

```
20    CONTINUE                                                STB00810
      RETURN                                                  STB00820
      END                                                     STB00830
```

```
      SUBROUTINE BASIC                                                  BAS00010
C     THIS ROUTINE WILL DETERMINE, ACCORDING TO WIND DIRECTION, WHICH   BAS00020
C     LINKS WILL AFFECT THE CONCENTRATION AT THE RECEPTOR AND, FOR THOSEBAS00030
C     LINKS, CALL THE APPROPRIATE ROUTINES TO COMPUTE THE CONCENTRATION BAS00040
C     AT A RECEPTOR                                                     BAS00050
      COMMON/DISP/XO(200),YO(200),NUMREC,THETA(24),WS(24),YCOEFF(24)    BAS00060
     1,YEXP(24),YCONSL,YCONSZ,ZCOEFF(24),ZEXP(24),ZCONST,GRDSIZ,BCKG    BAS00070
     2RD,MIX,ISTAB(24),SLAT,ICLD(24),IMET,NY,ITIME,HEIGHT(200)          BAS00080
      COMMON/LINK/NLINK,NLANE(200),X1(200),Y1(200),X2(200),Y2(200),LCAP(BAS00090
     1200),DIST(200),VEL(200),LCON(200,3),NSTOPS(200)                   BAS00100
      COMMON/MAINN/M                                                    BAS00110
      COMMON/WND/THETAP,AA,AB                                           BAS00120
      COMMON/BASC/I                                                     BAS00130
      COMMON/ORIGN/ALPHA,A,XX,YY                                        BAS00140
      COMMON/STCAN/ISTR(200),AST(200),NLKST(200),WST(200),              BAS00150
     &BUILDH(200),RECHGT(200),ISTLIN(200,2),IRSIDE(200)                 BAS00160
      DIMENSION RK(5)                                                   BAS00170
      DATA RK/25.5,5.5,1.75,1.0,0.5/                                    BAS00180
      DO 200 I=1,NLINK                                                  BAS00190
C     WRITE(6,45)                                                       BAS00200
C  45 FORMAT(/,1X,'INTERMEDIATE RESULTS FROM SUBROUTINE BASIC')         BAS00210
C     WRITE(6,46) I,NLINK,M,ISTR(M),NLKST(M)                            BAS00220
C  46 FORMAT(1X,'I=',I5,1X,'NLINK=',I5,1X,'M=',I5,1X,'ISTR(M)=',I5,     BAS00230
C    &1X,'NLKST(M)=',I5)                                                BAS00240
      IF(ISTR(M).EQ.0)GOTO 80                                           BAS00250
      NKLDUM=NLKST(M)                                                   BAS00260
      DO 50 II=1,NKLDUM                                                 BAS00270
C     WRITE(6,47) I,II,M,ISTLIN(M,II)                                   BAS00280
   47 FORMAT(1X,'I=',I5,1X,'II=',I5,1X,'M=',I5,1X,'ISTLIN(M,II)=',I5)   BAS00290
      IF(I.EQ.ISTLIN(M,II))GOTO 55                                      BAS00300
50    CONTINUE                                                          BAS00310
      GOTO 80                                                           BAS00320
   55 ISTM=ISTAB(ITIME)                                                 BAS00330
      DELTA=7.0*SQRT(RK(ISTM)*WST(M)/                                   BAS00340
     &WS(ITIME))                                                        BAS00350
C     WRITE(6,48) M,DELTA,BUILDH(M)                                     BAS00360
   48 FORMAT(1X,'M=',I5,1X,'DELTA=',E12.3,1X,'BUILDH(M)=',E12.3)        BAS00370
      IF(DELTA.GT.BUILDH(M))GOTO 80                                     BAS00380
      WRITE(6,56)M,I                                                    BAS00390
56    FORMAT(/,1X,'SUBROUTINE STREET CALLED FOR RECEPTOR',I5,1X,'AND ', BAS00400
```

```
      &'LINK',I5)                                                       BAS00410
       CALL STREET                                                      BAS00420
       GOTO 200                                                         BAS00430
80     CONTINUE                                                         BAS00440
       BA=X1(I)-XO(M)                                                   BAS00450
       BB=X2(I)-XO(M)                                                   BAS00460
       BC=Y1(I)-YO(M)                                                   BAS00470
       BD=Y2(I)-YO(M)                                                   BAS00480
       X1TR=BA*AB+BC*AA                                                 BAS00490
       X2TR=BB*AB+BD*AA                                                 BAS00500
       Y1TR=BC*AB-BA*AA                                                 BAS00510
       Y2TR=BD*AB-BB*AA                                                 BAS00520
       IF(Y1TR.LE.0.0.AND.Y2TR.LE.0.) GO TO 200                         BAS00530
       IF(Y1TR.LE.0.) GO TO 1                                           BAS00540
       IF(ABS(ATAN(ABS(Y1TR/X1TR))).GE.45.*6.2831853/360.) GO TO 6      BAS00550
       IF(Y2TR.LE.0.) GO TO 100                                         BAS00560
1      IF(ABS(ATAN(ABS(Y2TR/X2TR))).GE.45.*6.2831853/360.) GO TO 6      BAS00570
100    IF(X1TR.GT.0.0.AND.X2TR.LT.0.) GO TO 7                           BAS00580
       IF(X1TR.LT.0.0.AND.X2TR.GT.0.) GO TO 7                           BAS00590
       GO TO 200                                                        BAS00600
7      YONXEQ=Y1TR-((Y1TR-Y2TR)/(X1TR-X2TR))*X1TR                       BAS00610
       IF(YONXEQ.LT.0.) GO TO 200                                       BAS00620
6      CALL ORIGIN                                                      BAS00630
       CALL CORTRN                                                      BAS00640
       IF(ALPHA.EQ.0.) GO TO 400                                        BAS00650
       CALL CONCA1                                                      BAS00660
       GO TO 200                                                        BAS00670
400    CALL CONCA2                                                      BAS00680
200    CONTINUE                                                         BAS00690
       RETURN                                                           BAS00700
       END                                                              BAS00710
```

```
      SUBROUTINE ORIGIN                                                 ORI00010
C     THIS ROUTINE COMPUTES THE COORDINATES (XX,YY) OF POINT TO BE THE  ORI00020
C     ORIGIN IN NEW COORDINATE SYSTEM AND THE ANGLE FROM THE WIND VECTORORI00030
C     TO THE LINK                                                       ORI00040
      COMMON/DISP/XO(200),YO(200),NUMREC,THETA(24),WS(24),YCOEFF(24)    ORI00050
     1,YEXP(24),YCONSL,YCONSZ,ZCOEFF(24),ZEXP(24),ZCONST,GRDSIZ,BCKG    ORI00060
     2RD,MIX,ISTAB(24),SLAT,ICLD(24),NMET,NY,ITIME,HEIGHT(200)          ORI00070
      COMMON/LINK/NLINK,NLANE(200),X1(200),Y1(200),X2(200),Y2(200),LCAP(ORI00080
     12(0),DIST(200),VEL(200),LCON(200,3),NSTOPS(200)                   ORI00090
      COMMON/MAINN/M                                                    ORI00100
      COMMON/WND/THETAP,AA,AB                                           ORI00110
      COMMON/LSLOP/AM(200),B(200)                                       ORI00120
      COMMON/BASC/I                                                     ORI00130
      COMMON/ORIGN/ALPHA,A,XX,YY                                        ORI00140
C     COMPUTE THE POINT OF INTERSECTION (XX,YY) OF LINK (OR ITS EXTEN-  ORI00150
C     SION) WITH WIND VECTOR                                            ORI00160
      IF(X1(I).EQ.X2(I)) GO TO 1                                        ORI00170
      IF(THETAP.EQ.90.0.OR.THETAP.EQ.270.) GO TO 2                      ORI00180
      A=TAN(THETAP*3.14159265/180.)                                     ORI00190
      IF(A.EQ.AM(I)) GO TO 3                                            ORI00200
      XX=(XO(M)*A-YO(M)+B(I))/(A-AM(I))                                 ORI00210
      YY=AM(I)*XX+B(I)                                                  ORI00220
C     COMPUTE ANGLE FROM WIND VECTOR TO LINK, MEASURED COUNTERCLOCKWISE ORI00230
    6 ALPHA=ATAN2((AM(I)-A),(AM(I)*A+1.))                               ORI00240
      GO TO 5                                                           ORI00250
C     TAKE CARE OF CASES WHERE DENOMINATOR=0                            ORI00260
    1 IF(THETAP.EQ.90.0.OR.THETAP.EQ.270.) GO TO 4                      ORI00270
C     IN THIS CASE THE LINK IS PARALLEL TO THE OLD Y-AXIS BUT THE WIND  ORI00280
C     IS NOT                                                            ORI00290
      XX=X1(I)                                                          ORI00300
      A=TAN(THETAP*3.14159265/180.)                                     ORI00310
      YY=(XX-XO(M))*A+YO(M)                                             ORI00320
      ALPHA=ATAN2(1.,A)                                                 ORI00330
      GO TO 5                                                           ORI00340
C     IN THIS CASE THE WIND IS FROM EITHER NORTH OR SOUTH BUT THE LINK  ORI00350
C     IS NOT PARALLEL TO IT                                             ORI00360
    2 XX=XO(M)                                                          ORI00370
      YY=AM(I)*XO(M)+B(I)                                               ORI00380
      ALPHA=ATAN2(1.,-AM(I))                                            ORI00390
      GO TO 5                                                           ORI00400
```

```
C     IN THIS CASE THE WIND IS PARALLEL TO THE LINK BUT NOT TO THE OLD  ORI00410
C     Y-AXIS                                                            ORI00420
 3    IF(AM(I).EQ.0.) GO TO 11                                          ORI00430
      AMPRIM=-1./AM(I)                                                  ORI00440
      GO TO 12                                                          ORI00450
 11   XX=XO(M)                                                          ORI00460
      YY=Y1(I)                                                          ORI00470
      GO TO 7                                                           ORI00480
 12   BPRIME=YO(M)-AMPRIM*XO(M)                                         ORI00490
      XX=(BPRIME-B(I))/(AM(I)-AMPRIM)                                   ORI00500
      YY=AM(I)*XX+B(I)                                                  ORI00510
      GO TO 7                                                           ORI00520
C     IN THIS CASE THE WIND, THE LINK AND THE OLD Y-AXIS ARE ALL        ORI00530
C     PARALLEL                                                          ORI00540
 4    XX=X1(I)                                                          ORI00550
      YY=YO(M)                                                          ORI00560
 7    ALPHA=0.                                                          ORI00570
      GO TO 23                                                          ORI00580
C     THE FOLLOWING STATEMENTS CHECK TO SEE IF THE ANGLE BETWEEN THE    ORI00590
C     LINK AND THE WIND VECTOR IS LESS THAN 1 DEGREE--IF SO, THE ANGLE  ORI00600
C     IS SET EQUAL TO 0.                                                ORI00610
 5    ZZ=Y1(I)-Y2(I)                                                    ORI00620
      ZZZ=X1(I)-X2(I)                                                   ORI00630
      ZETA=(ATAN2(ZZ,ZZZ))*180./3.14159265                              ORI00640
      IF(ZETA.LT.0.) ZETA=360.+ZETA                                     ORI00650
      IF(ABS(ZETA-THETAP).LT.1.) GO TO 50                               ORI00660
      GO TO 23                                                          ORI00670
 50   IF(THETAP.EQ.90.0.OR.THETAP.EQ.270.) GO TO 4                      ORI00680
      GO TO 3                                                           ORI00690
 23   RETURN                                                            ORI00700
      END                                                               ORI00710
```

```
      SUBROUTINE CORTRN                                                 COR00010
C     THIS ROUTINE TRANSFORMS THE X,Y COORDINATES OF THE RECEPTOR AND   COR00020
C     THE LINK END POINTS INTO A NEW SYSTEM WITH THE Y-AXIS ALONG THE   COR00030
C     LINK AND WITH THE ORIGIN AT THE POINT OF INTERSECTION OF THE LINK COR00040
C     (OR ITS EXTENSION) AND THE WIND VECTOR                            COR00050
      COMMON/DISP/XO(200),YO(200),NUMREC,THETA(24),WS(24),YCOEFF(24)    COR00060
     1,YEXP(24),YCONSL,YCONSZ,ZCOEFF(24),ZEXP(24),ZCONST,GRDSIZ,BCKG    COR00070
     2RD,MIX,ISTAB(24),SLAT,ICLD(24),NMET,NY,ITIME,HEIGHT(200)          COR00080
      COMMON/LINK/NLINK,NLANE(200),X1(200),Y1(200),X2(200),Y2(200),LCAP(COR00090
     1200),DIST(200),VEL(200),LCON(200,3),NSTOPS(200)                   COR00100
      COMMON/MAINN/M                                                    COR00110
      COMMON/LSLOP/AM(200),B(200)                                       COR00120
      COMMON/BASC/I                                                     COR00130
      COMMON/ORIGN/ALPHA,A,XX,YY                                        COR00140
      COMMON/CRTRN/OMEGA,X1NEW,X2NEW,Y1NEW,Y2NEW,XONEW,YONEW            COR00150
C     COMPUTE THE ANGLE FROM OLD Y-AXIS TO NEW Y-AXIS (LINK), MEASURED  COR00160
C     COUNTERCLOCKWISE                                                  COR00170
      IF(X1(I).EQ.X2(I)) GO TO 1                                        COR00180
      OMEGA=ATAN2(1.,-AM(I))                                            COR00190
      GO TO 2                                                           COR00200
    1 OMEGA=0.                                                          COR00210
C     COMPUTE COORDINATES IN NEW SYSTEM                                 COR00220
    2 COSINE=COS(OMEGA)                                                 COR00230
      SINE=SIN(OMEGA)                                                   COR00240
      X1NEW=0.                                                          COR00250
      X2NEW=0.                                                          COR00260
      Y1NEW=(Y1(I)-YY)*COSINE-(X1(I)-XX)*SINE                           COR00270
      Y2NEW=(Y2(I)-YY)*COSINE-(X2(I)-XX)*SINE                           COR00280
      XONEW=(YO(M)-YY)*SINE+(XO(M)-XX)*COSINE                           COR00290
      YONEW=(YO(M)-YY)*COSINE-(XO(M)-XX)*SINE                           COR00300
      RETURN                                                            COR00310
      END                                                               COR00320
```

```
      SUBROUTINE CONCA1                                                 CN100010
C     THIS ROUTINE CALCULATES THE NORMALIZED CONCENTRATION AT A RECEPTORCN100020
C     RESULTING FROM EACH POINT (SEGMENT) ALONG THE LINK AND ADDS THESE CN100030
C     TO GIVE THE CONCENTRATION FROM THE LINK                           CN100040
C     THEN THE CONCENTRATIONS AT A RECEPTOR FROM ALL LINKS ARE SUMMED   CN100050
C     THIS ROUTINE IS USED WHEN THE WIND IS NOT PARALLEL TO THE LINK    CN100060
      COMMON /LINK/ NLINK, NLANE(200), X1(200), Y1(200),X2(200),Y2(200) CN100070
     1  , LCAP(200), DIST(200), VEL(200), LCON(200,3)                   CN100080
     2,NSTOPS(200)                                                      CN100090
      COMMON/DISP/XO(200),YO(200),NUMREC,THETA(24),WS(24),YCOEFF(24)    CN100100
     1,YEXP(24),YCONSL,YCONSZ,ZCOEFF(24),ZEXP(24),ZCONST,GRDSIZ,BCKG    CN100110
     2RD,MIX,ISTAB(24),SLAT,ICLD(24),NMET,NY,ITIME,HEIGHT(200)          CN100120
      COMMON TT(200,3),DELA(200,3),QZ(50),QUEUE(200),Q(200,4),DLINK(200,CN100130
     14),CCOUNT(200)                                                    CN100140
      COMMON/MAINN/M                                                    CN100150
      COMMON/BASC/I                                                     CN100160
      COMMON/ORIGN/ALPHA,A,XX,YY                                        CN100170
      COMMON/CRTRN/OMEGA,X1NEW,X2NEW,Y1NEW,Y2NEW,XONEW,YONEW            CN100180
      COMMON/CONA1/A1,A2,A3,CONNOR                                      CN100190
      COMMON/CONCTR/CONC(200)                                           CN100200
      COMMON/XSGN/A4                                                    CN100210
      COMMON/SPCIN/DELTAY                                               CN100220
      COMMON/MSC/NP,Y,GAMMA(5)                                          CN100230
      IF(XX.EQ.XO(M).AND.YY.EQ.YO(M).AND.ALPHA.EQ.1.570796) GO TO 15    CN100240
      CALL MISC                                                         CN100250
      A1=SIN(ALPHA)                                                     CN100260
      A2=COS(ALPHA)                                                     CN100270
      A3=XONEW/A1                                                       CN100280
      CALL XSIGN                                                        CN100290
      IF(I.EQ.131) PRINT 99,I,YCOEFF(ITIME),YEXP(ITIME),ZCOEFF(ITIME),ZECN100300
     1XP(ITIME),HEIGHT(I)                                               CN100310
 99   FORMAT(2X,I5,5(5X,F12.3))                                         CN100320
      DO 100 K=1,NP                                                     CN100330
      Y=Y+DELTAY                                                        CN100340
      A5=A3-Y*A2                                                        CN100350
      IF(A5.GT.0.0.AND.A4.LT.0.) GO TO 100                              CN100360
      IF(A5.LT.0.0.AND.A4.GT.0.) GO TO 100                              CN100370
      A6=ABS(A5)                                                        CN100380
      SIGY=YCOEFF(ITIME)*A6**YEXP(ITIME)+YCONSL                         CN100390
      IF(ZCONST.LT.3.) ZCONST=3.                                        CN100400
```

```
      SIGZ=ZCOEFF(ITIME)*A6**ZEXP(ITIME)+ZCONST                         CN100410
      A7=Y*A1/SIGY                                                      CN100420
      A8=HEIGHT(I)/SIGZ                                                 CN100430
      A9=0.5*(A7*A7+A8*A8)                                              CN100440
      IF(A9.GE.88.028) GO TO 100                                        CN100450
      DO 10 II=1,4                                                      CN100460
      IF(Y.GT.GAMMA(II).AND.Y.LE.GAMMA(II+1)) GO TO 3                   CN100470
      IF(Y.GE.GAMMA(II+1).AND.Y.LT.GAMMA(II)) GO TO 3                   CN100480
10    CONTINUE                                                          CN100490
3     CONNOR=CONNOR+Q(I,II)/(SIGY*SIGZ*EXP(A9))                         CN100500
100   CONTINUE                                                          CN100510
      CONC(M)=CONC(M)+CONNOR*DELTAY                                     CN100520
15    RETURN                                                            CN100530
      END                                                               CN100540
```

```
      SUBROUTINE ZBORDR                                                 ZB000010
C     THIS ROUTINE KEEPS TRACK OF THE END POINTS OF THE LINKS THAT      ZB000020
C     BORDER A ZONE--WHICH POINTS ARE OPEN AND WHICH CONNECT.  THIS MUSTZB000030
C     BE DONE BECAUSE THE LAST SIDE OF A ZONE MAY NOT BE A LINK.  IF IT ZB000040
C     IS NOT, THE OPEN END POINTS OF THE FIRST AND LAST LINKS ARE CON-  ZB000050
C     NECTED TO FORM THE LAST SIDE.                                     ZB000060
C     THIS ROUTINE CALLS TWO OTHER SUBROUTINES WHICH FIND THE GRID      ZB000070
C     CENTERS THAT LIE IN EACH ZONE.                                    ZB000080
      DIMENSION XCOOR(2),YCOOR(2),LOE(12),XC(12),YC(12)                 ZB000090
      COMMON/DISP/XO(200),YO(200),NUMREC,THETA(24),WS(24),YCOEFF(24)    ZB000100
     1,YEXP(24),YCONSL,YCONSZ,ZCOEFF(24),ZEXP(24),ZCONST,GRDSIZ,BCKG    ZB000110
     2RD,MIX,ISTAB(24),SLAT,ICLD(24),NMET,NY,ITIME,HEIGHT(200)          ZB000120
      COMMON/LINK/NLINK,NLANE(200),X1(200),Y1(200),X2(200),Y2(200),LCAP(ZB000130
     120O),DIST(200),VEL(200),LCON(200,3),NSTOPS(200)                   ZB000140
      COMMON/ZONES/NZONES,NZLINS(50),ZLINKS(12,50),ZATTR(50),ZGENR(50), ZB000150
     1NEXT,ZAT(2),ZGT(2),ZVT(3,50),ZNAME(50)                            ZB000160
      COMMON/PARKZ/PZ,PV(50),PLA(50),VZ(50),ZV(50),PD,PLS,PLBO,PNOS(50),ZB000170
     1PLL(50)                                                           ZB000180
      COMMON/LSLOP/AM(200),B(200)                                       ZB000190
      COMMON/GNGSC/XGRID(300),YGRID(300),NGRIDX,NGRIDY                  ZB000200
      COMMON/ZBORR/SLOPE(12),YINCPT(12),X1P,X2P,Y1P,Y2P,K,J             ZB000210
      COMMON/INERS/NPGL(300),YINT(300,4)                                ZB000220
      COMMON/CONCTR/CONC(200)                                           ZB000230
      COMMON/ZONGSC/NGSCZ(50),XZGSC,YZGSC                               ZB000240
      COMMON/CONAZ/CON(200,50)                                          ZB000250
      COMMON TT(200,3),DELA(200,3),QZ(50),QUEUE(200),Q(200,4),DLINK(200,ZB000260
     14),CCOUNT(200)                                                    ZB000270
      INTEGER ZLINKS,PNOS                                               ZB000280
      DO 10 J=1,NZONES                                                  ZB000290
      DO 11 M=1,NUMREC                                                  ZB000300
   11 CON(M,J)=0.                                                       ZB000310
      IF(QZ(J).EQ.0.) GO TO 10                                          ZB000320
      IF(PNOS(J) .GT. 1) GO TO 15                                       ZB000330
      NGSCZ(J) =0                                                       ZB000340
      GO TO 10                                                          ZB000350
   15 DO 20 MM=1,NGRIDY                                                 ZB000360
   20 NPGL(MM)=0                                                        ZB000370
      NOL=PNOS(J)                                                       ZB000380
      DO 16 K=1,NOL                                                     ZB000390
      I=IABS(ZLINKS(K,J))                                               ZB000400
```

```
      IF(K-2)1,2,3                                                      ZB000410
C     STORE END POINTS OF FIRST LINK ON BORDER OF ZONE. THEN CALL       ZB000420
C     ROUTINE THAT FINDS INTERSECTIONS OF THE VERTICAL LINES OF GRID    ZB000430
C     CENTERS WITH FIRST LINK.                                          ZB000440
 1    XCOOR(1)=X1(I)                                                    ZB000450
      XCOOR(2)=X2(I)                                                    ZB000460
      YCOOR(1)=Y1(I)                                                    ZB000470
      YCOOR(2)=Y2(I)                                                    ZB000480
      GO TO 50                                                          ZB000490
C     FIND END OF SECOND BORDER LINK THAT CONNECTS TO FIRST LINK.  THE  ZB000500
C     OTHER END IS OPEN END.  THEN CALL INTERSECTION ROUTINE TO MAKE    ZB000510
C     CALCULATIONS FOR SECOND LINK.                                     ZB000520
 2    DO 30 LL=1,2                                                      ZB000530
      IF(XCOOR(LL).NE.X1(I)) GO TO 31                                   ZB000540
      IF(YCOOR(LL).EQ.Y1(I)) GO TO 32                                   ZB000550
 31   IF(XCOOR(LL).NE.X2(I)) GO TO 33                                   ZB000560
      IF(YCOOR(LL).EQ.Y2(I)) GO TO 34                                   ZB000570
 33   IF(LL.EQ.2) GO TO 101                                             ZB000580
      GO TO 30                                                          ZB000590
 32   IF(LL.EQ.2) GO TO 35                                              ZB000600
      LOE(1)=2                                                          ZB000610
      LOE(2)=2                                                          ZB000620
      GO TO 40                                                          ZB000630
   35 LOE(1)=1                                                          ZB000640
      LOE(2)=2                                                          ZB000650
      GO TO 40                                                          ZB000660
 34   IF(LL.EQ.2) GO TO 36                                              ZB000670
      LOE(1)=2                                                          ZB000680
      LOE(2)=1                                                          ZB000690
      GO TO 40                                                          ZB000700
   36 LOE(1)=1                                                          ZB000710
      LOE(2)=1                                                          ZB000720
      GO TO 40                                                          ZB000730
 30   CONTINUE                                                          ZB000740
C     FIND OPEN END OF ALL OTHER BORDER LINKS AND CALL INTERSECTION     ZB000750
C     ROUTINE FOR EACH OF THESE LINKS                                   ZB000760
 3    IF(XC(K-1).NE.X1(I)) GO TO 43                                     ZB000770
      IF(YC(K-1).EQ.Y1(I)) GO TO 44                                     ZB000780
 43   IF(XC(K-1).NE.X2(I)) GO TO 101                                    ZB000790
      IF(YC(K-1).EQ.Y2(I)) GO TO 46                                     ZB000800
```

```
      GO TO 101                                                         ZB000810
 44   LOE(K)=2                                                          ZB000820
      GO TO 40                                                          ZB000830
 46   LOE(K)=1                                                          ZB000840
C     STORE COORDINATES OF OPEN (OPEN WITH RESPECT TO PREVIOUS LINK) ENDZB000850
C     OF EACH LINK SO THAT THE END OF THE LAST LINK,THAT SHOULD CONNECT ZB000860
C     WITH THE FIRST LINK, CAN BE FOUND.                                ZB000870
 40   IF(LOE(K).EQ.2) GO TO 41                                          ZB000880
      XC(K)=X1(I)                                                       ZB000890
      YC(K)=Y1(I)                                                       ZB000900
      GO TO 50                                                          ZB000910
 41   XC(K)=X2(I)                                                       ZB000920
      YC(K)=Y2(I)                                                       ZB000930
C     STORE SLOPES, Y-INTERCEPTS, AND COORDINATES OF THE ENDS OF EACH   ZB000940
C     LINK IN NEW LOCATIONS SO THAT THE SLOPE, INTERCEPT AND COORDINATESZB000950
C     OF A BORDER SEGMENT THAT IS NOT A LINK (IF IT EXISTS) MAY BE PUT  ZB000960
C     IN THE SAME ARRAY.                                                ZB000970
 50   SLOPE(K)=AM(I)                                                    ZB000980
      YINCPT(K)=B(I)                                                    ZB000990
      X1P=X1(I)                                                         ZB001000
      X2P=X2(I)                                                         ZB001010
      Y1P=Y1(I)                                                         ZB001020
      Y2P=Y2(I)                                                         ZB001030
C     FIND POINTS OF INTERSECTION OF VERTICAL LINES OF GRID CENTERS     ZB001040
C     WITH LINK                                                         ZB001050
      CALL INTERS                                                       ZB001060
 16   CONTINUE                                                          ZB001070
C     CHECK WHETHER FIRST AND LAST BORDER LINKS CONNECT.  IF NOT,       ZB001080
C     COMPUTE SLOPE AND Y-INTERCEPT OF SEGMENT THAT CONNECTS OPEN ENDS  ZB001090
C     OF FIRST AND LAST BORDER LINKS AND STORE IN ARRAYS                ZB001100
      LE1=LOE(1)                                                        ZB001110
      IF(XCOOR(LE1).NE.XC(NOL)) GO TO 49                                ZB001120
      IF(YCOOR(LE1).EQ.YC(NOL)) GO TO 100                               ZB001130
 49   AAA=XCOOR(LE1)-XC(NOL)                                            ZB001140
      IF(AAA.EQ.0.) GO TO 52                                            ZB001150
      SLOPE(NOL+1)=(YCOOR(LE1)-YC(NOL))/AAA                             ZB001160
      YINCPT(NOL+1)=YC(NOL)-SLOPE(NOL+1)*XC(NOL)                        ZB001170
      GO TO 51                                                          ZB001180
 52   SLOPE(NOL+1)=0.                                                   ZB001190
      YINCPT(NOL+1)=0.                                                  ZB001200
```

```
C       COUNT NUMBER OF SIDES OF ZONE INCLUDING NON-LINK SIDE (IF ONE     ZB001210
C       EXISTS).AND STORE COORDINATES OF END POINTS OF THIS LAST SIDE.    ZB001220
 51     N=NOL+1                                                           ZB001230
        X1F=XCOOR(LE1)                                                    ZB001240
        X2F=XC(NOL)                                                       ZB001250
        Y1F=YCOOR(LE1)                                                    ZB001260
        Y2F=YC(NOL)                                                       ZB001270
C       FIND INTERSECTION POINTS FOR SEGMENT                              ZB001280
        CALL INTERS                                                       ZB001290
C       FIND WHICH GRID CENTERS ARE LOCATED IN EACH ZONE                  ZB001300
 100    CALL ZNGSC                                                        ZB001310
        GO TO 10                                                          ZB001320
 101    PRINT 102,ZNAME(J)                                                ZB001330
  102   FORMAT(/////,10X,'ZONE ',I2,' HAS ILLEGAL BOUNDARY'/10X,'CALCULATIZB001340
       1ONS OF AREA SOURCE CONCENTRATIONS FROM THIS ZONE HAVE NOT BEEN MADZB001350
       2E')                                                               ZB001360
 10     CONTINUE                                                          ZB001370
        DO 12 M=1,NUMREC                                                  ZB001380
        DO 13 J=1,NZONES                                                  ZB001390
        IF(NGSCZ(J) .EQ. 0) GO TO 13                                      ZB001400
        CONC(M)=CONC(M)+CON(M,J)*QZ(J)/NGSCZ(J)                           ZB001410
 13     CONTINUE                                                          ZB001420
 12     CONTINUE                                                          ZB001430
        RETURN                                                            ZB001440
        END                                                               ZB001450
```

```
      SUBROUTINE CONCA2                                                 CN200010
C     THIS ROUTINE CALCULATES THE NORMALIZED CONCENTRATION AT A RECEPTORCN200020
C     RESULTING FROM EACH POINT (SEGMENT) ALONG THE LINK AND ADDS THESE CN200030
C     TO GIVE THE CONCENTRATION FROM THE LINK                           CN200040
C     THEN THE CONCENTRATIONS AT A RECEPTOR FROM ALL LINKS ARE SUMMED   CN200050
C     THIS ROUTINE IS USED WHEN THE WIND IS PARALLEL TO THE LINK        CN200060
      COMMON/DISP/XO(200),YO(200),NUMREC,THETA(24),WS(24),YCOEFF(24)    CN200070
     1,YEXP(24),YCONSL,YCONSZ,ZCOEFF(24),ZEXP(24),ZCONST,GRDSIZ,BCK6    CN200080
     2PD,MIX,ISTAB(24),SLAT,ICLD(24),NMET,NY,ITIME,HEIGHT(200)          CN200090
      COMMON TT(200,3),DELA(200,3),QZ(50),QUEUE(200),Q(200,4),DLINK(200,CN200100
     14),CCOUNT(200)                                                    CN200110
      COMMON/MAINN/M                                                    CN200120
      COMMON/BASC/I                                                     CN200130
      COMMON/CRTRN/OMEGA,X1NEW,X2NEW,Y1NEW,Y2NEW,XONEW,YONEW            CN200140
      COMMON/CONA1/A1,A2,A3,CONNOR                                      CN200150
      COMMON/CONCTR/CONC(200)                                           CN200160
      COMMON/XSGN/A4                                                    CN200170
      COMMON/SPCIN/DELTAY                                               CN200180
      COMMON/MSC/NP,Y,GAMMA(5)                                          CN200190
      CALL MISC                                                         CN200200
      CALL XSIGN                                                        CN200210
      DO 100 K=1,NP                                                     CN200220
      Y=Y+DELTAY                                                        CN200230
      IF(Y.GT.0.0.AND.A4.LT.0.) GO TO 100                               CN200240
      IF(Y.LT.0.0.AND.A4.GT.0.) GO TO 100                               CN200250
      D1=ABS(Y)                                                         CN200260
      SIGY=YCOEFF(ITIME)*D1**YEXP(ITIME)+YCONSL                         CN200270
      IF(ZCONST.LT.3.) ZCONST=3.                                        CN200280
      SIGZ=ZCOEFF(ITIME)*D1**ZEXP(ITIME)+ZCONST                         CN200290
      D2=XONEW/SIGY                                                     CN200300
      D3=HEIGHT(I)/SIGZ                                                 CN200310
      D4=0.5*(D2*D2+D3*D3)                                              CN200320
      IF(D4.GE.100.) GO TO 100                                          CN200330
      DO 10 II=1,4                                                      CN200340
      IF(Y.GT.GAMMA(II).AND.Y.LE.GAMMA(II+1)) GO TO 3                   CN200350
      IF(Y.GE.GAMMA(II+1).AND.Y.LT.GAMMA(II)) GO TO 3                   CN200360
   10 CONTINUE                                                          CN200370
    3 CONNOR=CONNOR+Q(I,II)/(SIGY*SIGZ*EXP(D4))                         CN200380
  100 CONTINUE                                                          CN200390
      CONC(M)=CONC(M)+CONNOR*DELTAY                                     CN200400
```

```
      RETURN                                                            CN200410
      END                                                               CN200420
```

```
      SUBROUTINE XSIGN                                                  XSI00010
C     THIS ROUTINE FINDS THE SIGN OF THE X-DISTANCE (USED IN THE POINT  XSI00020
C     SOURCE EQUATION) OF A POINT ON THE LINK OR ITS EXTENSION THAT     XSI00030
C     WOULD CONTRIBUTE TO THE CONCENTRATION AT THE RECEPTOR IF A SOURCE XSI00040
C     WERE LOCATED AT THAT POINT                                        XSI00050
      COMMON/DISP/XO(200),YO(200),NUMREC,THETA(24),WS(24),YCOEFF(24)    XSI00060
     1,YEXP(24),YCONSL,YCONSZ,ZCOEFF(24),ZEXP(24),ZCONST,GRDSIZ,BCKG    XSI00070
     2RD,MIX,ISTAB(24),SLAT,ICLD(24),NMET,NY,ITIME,HEIGHT(200)          XSI00080
      COMMON/LINK/NLINK,NLANE(200),X1(200),Y1(200),X2(200),Y2(200),LCAP(XSI00090
     1200),DIST(200),VEL(200),LCON(200,3),NSTOPS(200)                   XSI00100
      COMMON/MAINN/M                                                    XSI00110
      COMMON/WND/THETAP,AA,AB                                           XSI00120
      COMMON/BASC/I                                                     XSI00130
      COMMON/ORIGN/ALPHA,A,XX,YY                                        XSI00140
      COMMON/CRTRN/OMEGA,X1NEW,X2NEW,Y1NEW,Y2NEW,XONEW,YONEW            XSI00150
      COMMON/CONA1/A1,A2,A3,CONNOR                                      XSI00160
      COMMON/CONCTR/CONC(200)                                           XSI00170
      COMMON/XSGN/A4                                                    XSI00180
      C1=XX-XO(M)                                                       XSI00190
      C2=YY-YO(M)                                                       XSI00200
      IF(C1.EQ.0.0.AND.C2.EQ.0.) GO TO 9                                XSI00210
      IF(ALPHA.EQ.0.) GO TO 12                                          XSI00220
      A4=XONEW/A1                                                       XSI00230
      GO TO 10                                                          XSI00240
   12 IF(Y1NEW.EQ.YONEW) GO TO 11                                       XSI00250
      IF(THETAP.EQ.90.0.OR.THETAP.EQ.270.) GO TO 26                     XSI00260
      IF(A.EQ.0.) GO TO 13                                              XSI00270
   56 PWINT=Y1(I)+X1(I)/A                                               XSI00280
   29 WINDNT=YO(M)-A*XO(M)                                              XSI00290
      XPERP=A*(PWINT-WINDNT)/(A*A+1.)                                   XSI00300
      YPERP=A*XPERP+WINDNT                                              XSI00310
      C1=XPERP-XO(M)                                                    XSI00320
      C2=YPERP-YO(M)                                                    XSI00330
      IF(Y1NEW.NE.YONEW) GO TO 51                                       XSI00340
   30 A4=Y2NEW                                                          XSI00350
      GO TO 10                                                          XSI00360
   51 IF(ALPHA.NE.0.) GO TO 52                                          XSI00370
      A4=Y1NEW                                                          XSI00380
      GO TO 10                                                          XSI00390
   52 A4=XONEW/A1-Y1NEW*A2                                              XSI00400
```

```
      GO TO 10                                                          XSI00410
 26   C1=0.                                                             XSI00420
      C2=Y1(I)-YO(M)                                                    XSI00430
      GO TO 51                                                          XSI00440
 12   C1=X1(I)-XO(M)                                                    XSI00450
      C2=0.                                                             XSI00460
      GO TO 51                                                          XSI00470
 11   IF(THETAP.EQ.90.0.OR.THETAP.EQ.270.) GO TO 27                     XSI00480
      IF(A.EQ.0.) GO TO 28                                              XSI00490
      PWINT=Y2(I)+X2(I)/A                                               XSI00500
      GO TO 29                                                          XSI00510
 27   C1=0.                                                             XSI00520
      C2=Y2(I)-YO(M)                                                    XSI00530
      GO TO 30                                                          XSI00540
 28   C1=X2(I)-XO(M)                                                    XSI00550
      C2=0.                                                             XSI00560
      GO TO 30                                                          XSI00570
 9    IF(ALPHA.NE.0.) GO TO 56                                          XSI00580
      C1=X1(I)-XO(M)                                                    XSI00590
      C2=Y1(I)-YO(M)                                                    XSI00600
      A4=Y1NEW                                                          XSI00610
C     GO AHEAD WITH SIGN CHECK                                          XSI00620
 10   IF(THETAP.EQ.90.) GO TO 20                                        XSI00630
      IF(THETAP.EQ.270.) GO TO 24                                       XSI00640
      IF(A.GE.0.0.AND.A.LT.1.0E30) GO TO 101                            XSI00650
      IF(THETAP.GT.90.0.AND.THETAP.LE.185.) GO TO 22                    XSI00660
      KK=1                                                              XSI00670
      LL=-1                                                             XSI00680
      GO TO 42                                                          XSI00690
   22 KK=-1                                                             XSI00700
      LL=1                                                              XSI00710
      GO TO 42                                                          XSI00720
 101  IF(THETAP.GT.0.0.AND.THETAP.LE.95.) GO TO 20                      XSI00730
   24 KK=-1                                                             XSI00740
      LL=-1                                                             XSI00750
      GO TO 42                                                          XSI00760
   20 KK=1                                                              XSI00770
      LL=1                                                              XSI00780
C     IF POINT IS DOWNWIND OF RECEPTOR, CHANGE SIGN OF A4 SO THAT A4 S  XSI00790
C     OF THE SAME SIGN WILL CONTRIBUTE TO CONCENTRATION                 XSI00800
```

```
42    IF(C1*KK.LT.0.0.AND.C2*LL.LT.0.) A4=-A4                    XSI00810
      RETURN                                                     XSI00820
      END                                                        XSI00830
```

```
      SUBROUTINE SPCING                                                 SPC00010
C     THIS ROUTINE COMPUTES THE DISTANCE TO BE TAKEN BETWEEN POINT      SPC00020
C     SOURCES                                                           SPC00030
      COMMON/LINK/NLINK,NLANE(200),X1(200),Y1(200),X2(200),Y2(200),LCAP(SPC00040
     1200),DIST(200),VEL(200),LCON(200,3),NSTOPS(200)                   SPC00050
      COMMON/DISP/XO(200),YO(200),NUMREC,THETA(24),WS(24),YCOEFF(24)    SPC00060
     1,YEXP(24),YCONSL,YCONSZ,ZCOEFF(24),ZEXP(24),ZCONST,GRDSIZ,BCKG    SPC00070
     2RD,MIX,ISTAB(24),SLAT,ICLD(24),NMET,NY,ITIME,HEIGHT(200)          SPC00080
      COMMON/BASC/I                                                     SPC00090
      COMMON/CRTRN/OMEGA,X1NEW,X2NEW,Y1NEW,Y2NEW,XONEW,YONEW            SPC00100
      COMMON/SPCIN/DELTAY                                               SPC00110
      IF(Y1NEW.LE.YONEW.AND.YONEW.LE.Y2NEW) GO TO 1                     SPC00120
      IF(Y2NEW.LE.YONEW.AND.YONEW.LE.Y1NEW) GO TO 1                     SPC00130
      ABC=ABS(YONEW-Y1NEW)                                              SPC00140
      ABD=ABS(YONEW-Y2NEW)                                              SPC00150
      IF(ABC.GT.ABD) GO TO 2                                            SPC00160
      DSTNCE=SQRT(ABC*ABC+XONEW*XONEW)                                  SPC00170
      GO TO 3                                                           SPC00180
    1 DSTNCE=ABS(XONEW)                                                 SPC00190
      GO TO 3                                                           SPC00200
    2 DSTNCE=SQRT(ABD*ABD+XONEW*XONEW)                                  SPC00210
    3 SGYPRM=YCOEFF(ITIME)*DSTNCE**YEXP(ITIME)                          SPC00220
      DELTAY=0.1906817*SGYPRM                                           SPC00230
      IF(DELTAY.GE.1.5) GO TO 4                                         SPC00240
      DELTAY=0.160145*(SGYPRM+3.)                                       SPC00250
      IF(DELTAY.GE.1.5) GO TO 4                                         SPC00260
      YCONSL=3.                                                         SPC00270
      GO TO 5                                                           SPC00280
    4 YCONSL=DELTAY*2.                                                  SPC00290
      YCL=6.*NLANE(I)                                                   SPC00300
      IF(YCONSL.GT.YCL) YCONSL=YCL                                      SPC00310
    5 RETURN                                                            SPC00320
      END                                                               SPC00330
```

```
      SUBROUTINE MISC                                                   MIS00010
C     THIS ROUTINE CALCULATES THE VALUES OF VARIOUS PARAMETERS FOR USE  MIS00020
C     IN CONA1 OR CONA2                                                 MIS00030
      COMMON TT(200,3),DELA(200,3),QZ(50),QUEUE(200),Q(200,4),DLINK(200,MIS00040
     14),CCOUNT(200)                                                    MIS00050
      COMMON/BASC/I                                                     MIS00060
      COMMON/CRTRN/OMEGA,X1NEW,X2NEW,Y1NEW,Y2NEW,XONEW,YONEW            MIS00070
      COMMON/CONA1/A1,A2,A3,CONNOR                                      MIS00080
      COMMON/CONCTR/CONC(200)                                           MIS00090
      COMMON/SPCIN/DELTAY                                               MIS00100
      COMMON/MSC/NP,Y,GAMMA(5)                                          MIS00110
      CONNOR=0.                                                         MIS00120
      CALL SPCING                                                       MIS00130
      ALENTH=ABS(Y1NEW-Y2NEW)                                           MIS00140
      NP=ALENTH/DELTAY+0.9991                                           MIS00150
      DELTAY=ALENTH/NP                                                  MIS00160
      GAMMA(1)=Y1NEW                                                    MIS00170
      DO 11 II=2,5                                                      MIS00180
      IF(Y1NEW.GT.Y2NEW) GO TO 1                                        MIS00190
      Y=Y1NEW-DELTAY/2.                                                 MIS00200
      GAMMA(II)=GAMMA(II-1)+DLINK(I,II-1)                               MIS00210
      GO TO 11                                                          MIS00220
 1    Y=Y2NEW-DELTAY/2.                                                 MIS00230
      GAMMA(II)=GAMMA(II-1)-DLINK(I,II-1)                               MIS00240
 11   CONTINUE                                                          MIS00250
      RETURN                                                            MIS00260
      END                                                               MIS00270
```

```
      SUBROUTINE CENGSC                                                 GEN00010
C     THIS ROUTINE GENERATES THE COORDINATES OF THE CENTERS OF THE GRID GEN00020
C     SQUARES THAT OVERLAY THE INDIRECT SOURCE.  THE GRID OVERLAY IS    GEN00030
C     USED IN CALCULATING CONCENTRATIONS RESULTING FROM ZONE AREA       GEN00040
C     SOURCES.                                                          GEN00050
C     THE USER HAS THE OPTION OF SPECIFYING A GRID SIZE OR USING THE    GEN00060
C     DEFAULT VALUE OF 10 METERS.                                       GEN00070
      COMMON/DISP/XO(200),YO(200),NUMREC,THETA(24),WS(24),YCOEFF(24)    GEN00080
     1,YEXP(24),YCONSL,YCONSZ,ZCOEFF(24),ZEXP(24),ZCONST,GRDSIZ,BCKG    GEN00090
     2RD,MIX,ISTAB(24),SLAT,ICLD(24),NMET,NY,ITIME,HEIGHT(200)          GEN00100
      COMMON/LINK/NLINK,NLANE(200),X1(200),Y1(200),X2(200),Y2(200),LCAP(GEN00110
     1200),DIST(200),VEL(200),LCON(200,3),NSTOPS(200)                   GEN00120
      COMMON/GNGSC/XGRID(300),YGRID(300),NGRIDX,NGRIDY                  GEN00130
      XMAX=X1(1)                                                        GEN00140
      YMAX=Y1(1)                                                        GEN00150
      DO 1 I=1,NLINK                                                    GEN00160
      XMAX=AMAX1(XMAX,X1(I),X2(I))                                      GEN00170
      YMAX=AMAX1(YMAX,Y1(I),Y2(I))                                      GEN00180
    1 CONTINUE                                                          GEN00190
      IF(GRDSIZ.EQ.0.) GRDSIZ =10.                                      GEN00200
      NGRIDX=XMAX/GRDSIZ+1.999                                          GEN00210
      NGRIDY=YMAX/GRDSIZ+1.999                                          GEN00220
      DO 3 MM=1,NGRIDX                                                  GEN00230
    3 XGRID(MM)=0.05+(MM-1)*GRDSIZ                                      GEN00240
      DO 4 NN=1,NGRIDY                                                  GEN00250
    4 YGRID(NN)=0.05+(NN-1)*GRDSIZ                                      GEN00260
      YCONSZ=GRDSIZ                                                     GEN00270
      RETURN                                                            GEN00280
      END                                                               GEN00290
```

```
      SUBROUTINE INTERS                                                 INT00010
C     THIS ROUTINE FINDS THE POINTS OF INTERSECTION OF THE VERTICAL     INT00020
C     LINES OF GRID CENTERS WITH THE LINK AND STORES THEM IN AN ARRAY   INT00030
C     TO BE OPERATED ON BY SUBROUTINE ZNGSC.                            INT00040
      COMMON/GNGSC/XGRID(300),YGRID(300),NGRIDX,NGRIDY                  INT00050
      COMMON/ZEORR/SLOPE(12),YINCPT(12),X1P,X2P,Y1P,Y2P,K,J             INT00060
      COMMON/INERS/NPGL(300),YINT(300,4)                                INT00070
      IF(SLOPE(K).EQ.0.0.AND.YINCPT(K).EQ.0.) GO TO 1                   INT00080
      DO 2 MM=1,NGRIDX                                                  INT00090
      X=XGRID(MM)                                                       INT00100
      IF(X1P.GT.X2P) GO TO 3                                            INT00110
      IF(X.GT.X1P.AND.X.LT.X2P) GO TO 4                                 INT00120
      GO TO 2                                                           INT00130
    3 IF(X.GE.X1P.OR.X.LE.X2P) GO TO 2                                  INT00140
    4 Y=SLOPE(K)*X+YINCPT(K)                                            INT00150
    5 NPGL(MM)=NPGL(MM)+1                                               INT00160
      NSTAR=NPGL(MM)                                                    INT00170
      YINT(MM,NSTAR)=Y                                                  INT00180
    2 CONTINUE                                                          INT00190
    1 RETURN                                                            INT00200
      END                                                               INT00210
```

```
      SUBROUTINE ZNGSC                                                  ZNG00010
C     THIS ROUTINE DETERMINES WHICH GRID CENTERS FALL IN EACH ZONE      ZNG00020
C     THIS IS DONE BY FINDING WHICH GRID SQUARE CENTERS ON EACH VERTICALZNG00030
C     LINE OF THEM FALL BETWEEN THE POINTS WHERE THE VERTICAL LINE      ZNG00040
C     INTERSECTS THE ZONE BORDER LINKS                                  ZNG00050
      DIMENSION ARY(4)                                                  ZNG00060
      COMMON/ZONES/NZONES,NZLINS(50),ZLINKS(12,50),ZATTR(50),ZGENR(50), ZNG00070
     1NEXT,ZAT(2),ZGT(2),ZVT(3,50),ZNAME(50)                            ZNG00080
      COMMON/GNGSC/XGRID(300),YGRID(300),NGRIDX,NGRIDY                  ZNG00090
      COMMON/ZBORR/SLOPE(12),YINCPT(12),X1P,X2P,Y1P,Y2P,K,J             ZNG00100
      COMMON/INERS/NPGL(300),YINT(300,4)                                ZNG00110
      COMMON/ZONGSC/NGSCZ(50),XZGSC,YZGSC                               ZNG00120
      NGSCZ(J)=0                                                        ZNG00130
      DO 1 MM=1,NGRIDX                                                  ZNG00140
      NSTAR=NPGL(MM)                                                    ZNG00150
      IF(NSTAR.EQ.0) GO TO 1                                            ZNG00160
      GO TO(2,3,2,3),NSTAR                                              ZNG00170
C     THESE STATEMENTS PUT INTERSECTION POINTS ON EACH VERTICAL GRID    ZNG00180
C     CENTER LINE IN AN ARRAY IN DESCENDING NUMERICAL ORDER OF THEIR    ZNG00190
C     VALUES                                                            ZNG00200
    2 NP=1                                                              ZNG00210
      NI=NSTAR                                                          ZNG00220
      DO 20 KOO=1,NSTAR                                                 ZNG00230
      DO 4 L=1,NI                                                       ZNG00240
      DO 5 LP=1,NI                                                      ZNG00250
      IF(YINT(MM,L).LT.YINT(MM,LP)) GO TO 4                             ZNG00260
      IF(LP.LT.NI) GO TO 5                                              ZNG00270
      ARY(NP)=YINT(MM,L)                                                ZNG00280
      GO TO 6                                                           ZNG00290
    5 CONTINUE                                                          ZNG00300
    4 CONTINUE                                                          ZNG00310
    6 IP=0                                                              ZNG00320
      DO 7 L=1,NI                                                       ZNG00330
      IF(YINT(MM,L).EQ.ARY(NP)) GO TO 7                                 ZNG00340
      IP=IP+1                                                           ZNG00350
      YINT(MM,IP)=YINT(MM,L)                                            ZNG00360
    7 CONTINUE                                                          ZNG00370
      NI=NI-1                                                           ZNG00380
      NP=NP+1                                                           ZNG00390
   20 CONTINUE                                                          ZNG00400
```

```
C     THESE STATEMENTS CHECK WHICH GRID CENTERS FALL BETWEEN PAIRS OF   ZNG00410
C     INTERSECTION POINTS.                                              ZNG00420
      DO 21 NN=1,NGRIDY                                                 ZNG00430
      N=1                                                               ZNG00440
 23   IF(YGRID(NN).GE.ARY(N)) GO TO 21                                  ZNG00450
      IF(YGRID(NN).GT.ARY(N+1)) GO TO 22                                ZNG00460
      IF(NPGL(MM).EQ.2) GO TO 21                                        ZNG00470
      IF(N.EQ.3) GO TO 21                                               ZNG00480
      N=3                                                               ZNG00490
      GO TO 23                                                          ZNG00500
 22   NGSCZ(J)=NGSCZ(J)+1                                               ZNG00510
      XZGSC=XGRID(MM)                                                   ZNG00520
      YZGSC=YGRID(NN)                                                   ZNG00530
C     SUBROUTINE ZCHECK IS CALLED TO SEE IF GRID SQUARE CENTER LOCATION ZNG00540
C     IS SUCH THAT THE POINT WILL AFFECT THE CONCENTRATION AT THE       ZNG00550
C     RECEPTOR                                                          ZNG00560
      CALL ZCHECK                                                       ZNG00570
 21   CONTINUE                                                          ZNG00580
 1    CONTINUE                                                          ZNG00590
      GO TO 25                                                          ZNG00600
 2    PRINT 102,ZNAME(J)                                                ZNG00610
  102 FORMAT(/////,10X,'ZONE ',I2,' HAS ILLEGAL BOUNDARY'/10X,'CALCULATIZNG00620
     1ONS OF AREA SOURCE CONCENTRATIONS FROM THIS ZONE HAVE NOT BEEN MADZNG00630
     2E')                                                               ZNG00640
 25   RETURN                                                            ZNG00650
      END                                                               ZNG00660
```

```
      SUBROUTINE ZCHECK                                                 ZCH00010
C     THIS ROUTINE WILL DETETMINE, ACCORDING TO WIND DIRECTION, WHICH   ZCH00020
C     GRID SQUARE CENTERS THAT LIE WITHIN ZONES ARE IN A LOCATION FROM  ZCH00030
C     WHICH THEY CAN CONTRIBUTE TO THE CONCENTRATION AT THE RECEPTOR ANDZCH00040
C     FOR THOSE CENTERS, CALL THE ROUTINE THAT COMPUTES THE CONCENTRA-  ZCH00050
C     TION AT THE RECEPTOR.                                             ZCH00060
      COMMON/DISP/XO(200),YO(200),NUMREC,THETA(24),WS(24),YCOEFF(24)    ZCH00070
     1,YEXP(24),YCONSL,YCONSZ,ZCOEFF(24),ZEXP(24),ZCONST,GRDSIZ,BCKG    ZCH00080
     2BD,MIX,ISTAB(24),SLAT,ICLD(24),NMET,NY,ITIME,HEIGHT(200)          ZCH00090
      COMMON/WND/THETAB,AA,AB                                           ZCH00100
      COMMON/ZONGSC/NGSCZ(50),XZGSC,YZGSC                               ZCH00110
      COMMON/ZCHCK/XZTR,YZTR,M                                          ZCH00120
      DO 2 M=1,NUMREC                                                   ZCH00130
      BA=XZGSC-XO(M)                                                    ZCH00140
      BC=YZGSC-YO(M)                                                    ZCH00150
      XZTR=BA*AB+BC*AA                                                  ZCH00160
      YZTR=BC*AB-BA*AA                                                  ZCH00170
      IF(XZTR.EQ.0.0.AND.YZTR.GE.0.) GO TO 4                            ZCH00180
      IF(YZTR.LE.0.) GO TO 2                                            ZCH00190
      IF(ABS(ATAN(ABS(YZTR/XZTR))).LT.45.*6.2831853/360.) GO TO 2       ZCH00200
    4 CALL CONCA3                                                       ZCH00210
    2 CONTINUE                                                          ZCH00220
      RETURN                                                            ZCH00230
      END                                                               ZCH00240
```

```
      SUBROUTINE CONCAT                                                 CN300010
C     THIS ROUTINE CALCULATES THE CONTRIBUTION FROM ZONES (AREA SOURCES)CN300020
C     TO THE CONCENTRATION AT THE RECEPTOR AND ADDS IT TO THE LINK      CN300030
C     CONTRIBUTIONS                                                     CN300040
      COMMON/DISP/XO(200),YO(200),NUMREC,THETA(24),WS(24),YCOEFF(24)    CN300050
     1,YEXP(24),YCONSL,YCONSZ,ZCOEFF(24),ZEXP(24),ZCONST,GRDSIZ,BCKG    CN300060
     2RD,MIX,ISTAB(24),SLAT,ICLD(24),NMET,NY,ITIME,HEIGHT(200)          CN300070
      COMMON/ZBORR/SLOPE(12),YINCPT(12),X1P,X2P,Y1P,Y2P,K,J             CN300080
      COMMON/ZCHCK/XZTR,YZTR,M                                          CN300090
      COMMON/CONA3/CON(200,50)                                          CN300100
      COMMON/EXCLUD/NOPARK                                              CN300110
      SIGY=YCOEFF(ITIME)*YZTR**YEXP(ITIME)+YCONSZ                       CN300120
      IF(ZCONST.LT.3.) ZCONST=3.                                        CN300130
      SIGZ=ZCOEFF(ITIME)*YZTR**ZEXP(ITIME)+ZCONST                       CN300140
      D=(0.5*XZTR*XZTR)/(SIGY*SIGY)                                     CN300150
      IF(D.GE.100.) GO TO 1                                             CN300160
      IF(NOPARK.EQ.1)GOTO 1                                             CN300170
      CON(M,J)=CON(M,J)+1./(SIGY*SIGZ*EXP(D))                           CN300180
    1 RETURN                                                            CN300190
      END                                                               CN300200
```

```
      SUBROUTINE PRNTOU                                                 PRN00010
C     THIS ROUTINE PRINTS THE MODEL RESULTS                             PRN00020
      COMMON/DISP/XO(200),YO(200),NUMREC,THETA(24),WS(24),YCOEFF(24)    PRN00030
     1,YEXP(24),YCONSL,YCONSZ,ZCOEFF(24),ZEXP(24),ZCONST,GRDSIZ,BCKG    PRN00040
     2RD,MIX,ISTAB(24),SLAT,ICLD(24),NMET,NY,ITIME,HEIGHT(200)          PRN00050
      COMMON / COMM/ I,Z,L,J, K, TOD, DOW, TP, TOTATT, TTPZ,TOTGEN,ITM  PRN00060
     1,NYEAR,LHEAD(18), TUNFAC                                          PRN00070
      COMMON/LINK/NLINK,NLANE(200),X1(200),Y1(200),X2(200),Y2(200),LCAP(PRN00080
     1200),DIST(200),VEL(200),LCON(200,3),NSTOPS(200)                   PRN00090
      COMMON/ZONES/NZONES,NZLINS(50),ZLINKS(12,50),ZATTR(50),ZGENR(50), PRN00100
     1NEXT,ZAT(2),ZGT(2),ZVT(3,50),ZNAME(50)                            PRN00110
      COMMON / INTRST/ NINS, LIN(70,4),ITYPC (70), CYCL(70), PH(70,4),  PRN00120
     1 CI(70), GCAP(70,4), QUE(70,4)                                    PRN00130
      COMMON/CONCTR/CONC(200)                                           PRN00140
      PRINT 21,LHEAD,NYEAR                                              PRN00150
   21 FORMAT(1H1,47X,'INDIRECT SOURCE MODEL OF AIR POLLUTION'/////,5X,18PRN00160
     1A4,10X,'DATE = ',I6/)                                             PRN00170
      PRINT 22,NLINK,NZONES,NINS                                        PRN00180
   22 FORMAT(22X,'NO. OF LINKS = ',I3,10X,'NO. OF ZONES = ',I5,10X,'NO. PRN00190
     1OF INTERSECTIONS = ',I5/////)                                     PRN00200
      PRINT 43,TOD,THETA(ITIME),WS(ITIME),ISTAB(ITIME)                  PRN00210
   43 FORMAT(10X,'HOUR = ',F5.0,8X,'WIND DIRECTION = ',F8.0,8X,'WIND SPEPRN00220
     1ED = ',1PE8.1,8X,'STABILITY = ',I6///)                            PRN00230
      PRINT 45                                                          PRN00240
   45 FORMAT(1X,8HRECEPTOR,49X,'CO CONCENTRATION (PPM)'//)              PRN00250
      MM=NUMREC/10.+1.01                                                PRN00260
      DO 40 L=1,MM                                                      PRN00270
      L2=L*10                                                           PRN00280
      L1=L2-9                                                           PRN00290
      IF(L.EQ.MM) L2=NUMREC                                             PRN00300
      PRINT 46,L1,L2,(CONC(M),M=L1,L2)                                  PRN00310
   46 FORMAT(1H I3,2H -,I3,1X,10(2X,1PE10.2))                           PRN00320
   40 CONTINUE                                                          PRN00330
      RETURN                                                            PRN00340
      END                                                               PRN00350
```

```
      SUBROUTINE STREET                                                 STR00010
      COMMON/LINK/NLINK,NLANE(200),X1(200),Y1(200),X2(200),             STR00020
     &Y2(200),LCAP(200),DIST(200),VEL(200),LCON(200,3)                  STR00030
      COMMON/DISP/XO(200),YO(200),NUMREC,THETA(24),WS(24),YCOEFF(24),   STR00040
     &YEAR(24),YCONSL,YCONSZ,ZCDEFF(24),ZEXP(24),ZCONST,GRDSIZ,         STR00050
     &BCKGRD,MIX,ISTAB(24),SLAT,ICLD(24),IMET,NY,ITIME,                 STR00060
     &HEIGHT(200)                                                       STR00070
      COMMON/STCAN/ISTR(200),AST(200),NLKST(200),WST(200),              STR00080
     &BUILDH(200),RECHGT(200),ISTLIN(200,2),IRSIDE(200)                 STR00090
      COMMON/MAINN/M                                                    STR00100
      COMMON/BASC/I                                                     STR00110
      COMMON/CONCTR/CONC(200)                                           STR00120
      COMMON TT(200,3),DELA(200,3),QZ(50),QUEUE(200),Q(200,4),          STR00130
     &DLINK(200,4),COUNT(200)                                           STR00140
      DIMENSION DX(4),DY(4),XX(4),YY(4)                                 STR00150
C***  CONVERT STREET WIDTH, BUILDING HEIGHT AND RECEPTOR HEIGHT TO      STR00160
C***  METERS                                                            STR00170
      W=WST(M)                                                          STR00180
      BHT=BUILDH(M)                                                     STR00190
      RHT=RECHGT(M)                                                     STR00200
      CK=7.0                                                            STR00210
      XLO=2.                                                            STR00220
      WRITE(6,5) M,I,W,BHT,RHT,XLO                                      STR00230
5     FORMAT(1X,'M=',I5,1X,'I=',I5,1X,'W=',F10.2,1X,'BHT=',F10.2,       STR00240
     &1X,'RHT=',F10.2,1X,'XLO=',F10.2)                                  STR00250
C***  DETERMINE WHICH SEGMENT OF THE LINK CONTRIBUTES MOST              STR00260
C***  STRONGLY TO THE STREET CANYON RECEPTOR                            STR00270
      DXT=X2(I)-X1(I)                                                   STR00280
      DYT=Y2(I)-Y1(I)                                                   STR00290
      SL=SQRT(DXT**2+DYT**2)                                            STR00300
      WRITE(6,6)M,I,DXT,DYT,SL                                          STR00310
6     FORMAT(1X,'M=',I5,1X,'I=',I5,1X,'DXT=',F10.2,1X,'DYT=',F10.2,1X,  STR00320
     &'SL=',F10.2)                                                      STR00330
      DO 10 II=1,4                                                      STR00340
      DX(II)=(DLINK(I,II)/SL)*DXT                                       STR00350
      DY(II)=(DLINK(I,II)/SL)*DYT                                       STR00360
      WRITE(6,12)M,I,II,DX(II),DY(II)                                   STR00370
12    FORMAT(1X,'M=',I5,1X,'I=',I5,1X,'II=',I5,1X,'DX(II)=',F10.2,      STR00380
     &1X,'DY(II)=',F10.2)                                               STR00390
10    CONTINUE                                                          STR00400
```

```
      XX(1)=DX(1)/2.+X1(I)                                              STR00410
      YY(1)=DY(1)/2.+Y1(I)                                              STR00420
      DO 20 JJ=2,4                                                      STR00430
      XX(JJ)=XX(JJ-1)+DX(JJ-1)/2.+DX(JJ)/2.                             STR00440
      YY(JJ)=YY(JJ-1)+DY(JJ-1)/2.+DY(JJ)/2.                             STR00450
20    CONTINUE                                                          STR00460
      DTEST=1.0E+8                                                      STR00470
      DO 30 JJ=1,4                                                      STR00480
      DTS=SQRT((XX(JJ)-XO(M))**2+(YY(JJ)-YO(M))**2)                     STR00490
      WRITE(6,22) JJ,M,XX(JJ),YY(JJ),XO(M),YO(M),DTS                    STR00500
   22 FORMAT(1X,'JJ=',I5,1X,'M=',I5,1X,'XX(JJ)=',E12.3,/,1X,            STR00510
     &'YY(JJ)=',E12.3,1X,'XO(M)=',E12.3,1X,'YO(M)=',E12.3,/,            STR00520
     &1X,'DTS=',E12.3)                                                  STR00530
      IF(DTS.LT.DTEST)GOTO 25                                           STR00540
      GOTO 30                                                           STR00550
25    DTEST=DTS                                                         STR00560
      JKEEP=JJ                                                          STR00570
30    CONTINUE                                                          STR00580
      WRITE(6,32) I,JKEEP,Q(I,JKEEP)                                    STR00590
   32 FORMAT(1X,'I=',I5,1X,'JKEEP=',I5,1X,'Q(I,JKEEP)=',F12.3)          STR00600
      CW=CK*Q(I,JKEEP)*(RHT-RHT)/(W*(WS(ITIME)+0.5)*RHT)                STR00610
      CL=CK*Q(I,JKEEP)/((WS(ITIME)+0.5)*(SQRT((W/2.)**2+RHT**2)+XLO))   STR00620
      WRITE(6,35)M,I,JKEEP,CW,CL                                        STR00630
   35 FORMAT(1X,'M=',I5,1X,'I=',I5,1X,'JKEEP=',I5,1X,'CW=',E12.3,1X,    STR00640
     &'CL=',E12.3)                                                      STR00650
      EPSIL=THETA(ITIME)-AST(M)                                         STR00660
      WRITE(6,40)M,I,EPSIL                                              STR00670
40    FORMAT(1X,'M=',I5,1X,'I=',I5,1X,'EPSIL=',F10.2)                   STR00680
      IF(EPSIL.LT.0.)EPSIL=360.+EPSIL                                   STR00690
      IF(EPSIL.GT.30.. AND.EPSIL.LE.150..AND.IRSIDE(M).EQ.1) GO TO 50   STR00700
      IF(EPSIL.GT.30..AND.EPSIL.LE.150..AND.IRSIDE(M).EQ.2) GO TO 60    STR00710
      IF(EPSIL.GT.210..AND.EPSIL.LE.330..AND.IRSIDE(M).EQ.1) GO TO 60   STR00720
      IF(EPSIL.GT.210..AND.EPSIL.LE.330..AND.IRSIDE(M).EQ.2) GO TO 50   STR00730
      CONC(M)=CONC(M)+((CL+CW)/2.)*3.14159265*WS(ITIME)                 STR00740
      GOTO 100                                                          STR00750
50    CONC(M)=CONC(M)+CL*3.14159265*WS(ITIME)                           STR00760
      GOTO 100                                                          STR00770
60    CONC(M)=CONC(M)+CW*3.14159265*WS(ITIME)                           STR00780
  100 CONTINUE                                                          STR00790
      RETURN                                                            STR00800
```

```
END
```

STR00810

```
      SUBROUTINE INITMM                                                 INI00010
      COMMON/AGEMM/DEC                                                  INI00020
      COMMON/COCOEF/COEF                                                INI00030
      COMMON/COEFMM/BAD                                                 INI00040
      COMMON/DET/DETER                                                  INI00050
      COMMON/MYMCOM/MYM,MYR,TF                                          INI00060
      REAL MYM(120),MYR(120)                                            INI00070
      REAL DEC(20),TF(20,6),COEF(20,12),BAD(12,20),DETER(20)            INI00080
C                                                                       INI00090
C                                                                       INI00100
C     SET UP ARRAY OF VMT (OR TRAVEL FACTOR) BY MODEL YEAR GROUP        INI00110
C                                                                       INI00120
C                                                                       INI00130
      DEC(20)=TF(20,1)                                                  INI00140
      DEC(19)=TF(19,1)                                                  INI00150
      DEC(18)=TF(18,1)                                                  INI00160
      DEC(17)=TF(17,1)+TF(16,1)                                         INI00170
      DEC(16)=TF(15,1)                                                  INI00180
      DO 5 I=8,15                                                       INI00190
    5 DEC(I)=0.0                                                        INI00200
      DEC(7)=TF(14,1)                                                   INI00210
      DEC(6)=TF(13,1)                                                   INI00220
      DEC(5)=TF(12,1)                                                   INI00230
      DEC(4)=TF(11,1)                                                   INI00240
      DEC(3)=0.0                                                        INI00250
      DEC(2)=0.0                                                        INI00260
      DO 10 I=1,10                                                      INI00270
   10 DEC(2)=DEC(2)+TF(I,1)                                             INI00280
      DEC(1)=0.0                                                        INI00290
C                                                                       INI00300
C                                                                       INI00310
C     DETERIORATE MODAL MODEL COEFFICIENTS TO 1977                      INI00320
C                                                                       INI00330
C                                                                       INI00340
      DO 100 J=1,20                                                     INI00350
      DO 100 K=1,12                                                     INI00360
  100 COEF(J,K)=BAD(K,J)*(1.0+DETER(J))                                 INI00370
C     WRITE(6,200)((BAD(IE,I1),IE=1,12),I1=1,20),                       INI00380
C    &(DETER(I1),I1=1,20),                                              INI00390
C    &((COEF(I1,IE),IE=1,12),I1=1,20),                                  INI00400
```

```
C     &(TF(I1,1),I1=1,20),(DEC(I1),I1=1,20)                    INI00410
C 200 FORMAT (60(1X,4E15.8/),'1'./20(1X,F15.5/),               INI00420
C     &'1'/60(1X,4F15.8/),'1'/20(1X,F15.5/))                   INI00430
      RETURN                                                   INI00440
      END                                                      INI00450
```

```
      SUBROUTINE CRUZ(V,EMCRUZ)                                         CRU00010
      COMMON/AGEMM/DEC                                                  CRU00020
      COMMON/COCOEF/COEF                                                CRU00030
      DIMENSION COEF(20,12),DEC(20),X(12)                               CRU00040
      EMCRUZ=0.                                                         CRU00050
      X(10)=1.                                                          CRU00060
      X(11)=V                                                           CRU00070
      X(12)=V**2                                                        CRU00080
      DO 100 I1=1,20                                                    CRU00090
      IF(DEC(I1).EQ.0.) GO TO 100                                       CRU00100
      DO 50 I2=10,12                                                    CRU00110
      EMCRUZ=EMCRUZ+X(I2)*COEF(I1,I2)*DEC(I1)                           CRU00120
   50 CONTINUE                                                          CRU00130
  100 CONTINUE                                                          CRU00140
      RETURN                                                            CRU00150
      END                                                               CRU00160
```

```
      SUBROUTINE ACDC(V,A,EMAD)                                         ACD00010
      COMMON/AGEMM/DEC                                                  ACD00020
      COMMON/COCOEF/COEF                                                ACD00030
      DIMENSION COEF(20,12),DEC(20),X(12)                               ACD00040
      EMAD=0.                                                           ACD00050
      IPRSW=0                                                           ACD00060
      AMAX=1.0                                                          ACD00070
      AMIN=-1.2                                                         ACD00080
      A1=-1.0/AMIN                                                      ACD00090
      A2=-1.0/AMAX                                                      ACD00100
      IF(A.EQ.0.) GO TO 200                                             ACD00110
      T=ABS(V/A)                                                        ACD00120
      AA=A*A                                                            ACD00130
      TT=T*T                                                            ACD00140
      X(1)=T                                                            ACD00150
      X(2)=A*TT/2.0                                                     ACD00160
      X(3)=A*T                                                          ACD00170
      X(4)=AA*TT/2.0                                                    ACD00180
      X(5)=AA*TT*T/3.0                                                  ACD00190
      X(6)=AA*T                                                         ACD00200
      X(8)=AA*A*TT*T/3.0                                                ACD00210
      X(7)=AA*A*TT/2.0                                                  ACD00220
      X(9)=AA*AA*TT*T/3.0                                               ACD00230
      X(10)=X(1)                                                        ACD00240
      X(11)=X(2)                                                        ACD00250
      X(12)=X(5)                                                        ACD00260
      IF(A.GE.AMAX) HOA=0.                                              ACD00270
      IF(A.LE.AMIN) HOA=0.                                              ACD00280
      IF(A.GE.0.0.AND.A.LT.AMAX) HOA=(A2*A)+1.0                         ACD00290
      IF(A.LE.0.0.AND.A.GT.AMIN) HOA=(A1*A)+1.0                         ACD00300
      SOA=1.-HOA                                                        ACD00310
      DO 100 I1=1,20                                                    ACD00320
      IF(DEC(I1).EQ.0.) GO TO 100                                       ACD00330
      DO 70 I2=1,9                                                      ACD00340
      IF(A.LT.0.0.AND.I2.EQ.2) GO TO 50                                 ACD00350
      IF(A.LT.0.0.AND.I2.EQ.4) GO TO 50                                 ACD00360
      IF(A.LT.0.0.AND.I2.EQ.7) GO TO 50                                 ACD00370
      EMAD=EMAD+X(I2)*SOA*COEF(I1,I2)*DEC(I1)                           ACD00380
      GO TO 65                                                          ACD00390
   50 EMAD=EMAD-X(I2)*SOA*COEF(I1,I2)*DEC(I1)                           ACD00400
```

```
   65 IF(IPRSW.GE.1) WRITE(6,9020) EMAD,I1,I2                          ACD00410
 9020 FORMAT (10X,'EMAD =',F12.3,2I5)                                  ACD00420
   70 CONTINUE                                                         ACD00430
      DO 80 I2=10,12                                                   ACD00440
      IF(A.LT.0.0.AND.I2.EQ.11) GO TO 75                               ACD00450
      EMAD=EMAD+X(I2)*HOA*COEF(I1,I2)*DEC(I1)                          ACD00460
      GO TO 80                                                         ACD00470
   75 EMAD=EMAD-X(I2)*HOA*COEF(I1,I2)*DEC(I1)                          ACD00480
   80 CONTINUE                                                         ACD00490
  100 CONTINUE                                                         ACD00500
      GO TO 250                                                        ACD00510
  200 EMAD=0.                                                          ACD00520
  250 CONTINUE                                                         ACD00530
      EMAD=EMAD/T                                                      ACD00540
      RETURN                                                           ACD00550
      END                                                              ACD00560
```

```
      SUBROUTINE SUP2 (INIFLG,T1,I2,A1,X1,X2,X3,X4,A21,A22,A23,
     &A24,A25,A26,I5,I6,I7,EMCRUZ,EMIDLE)
C
      COMMON/FLGCOM/ALTFLG,ALHFLG,TRKFLG,IRDFLG,SP3FLG,NMHFLG,IDLFLG,
     *              UNFFLG,MYMFFG,IMFLG,ICEVFG,PRTFLG,IFORM
      COMMON/REGCOM/IREJN
      COMMON/LNKCOM/SPD,TEMP,PCCO,PCHS,PCCC,VMTMIX
C     COMMON/ALTCOM/NMYALT,ALTKOD
      COMMON/MYMCOM/MYM,MYR,TF
C     COMMON/JUNK1/AC,XLOAD,TRAILR,ABSHUM
C     COMMON/JUNK2/HGWGT,HDWGT,HGCID,HDCID
C     COMMON/PROJCM/PROJID
C     COMMON/IMCOM/ICYIM,ISTRIN,IMTFLG,MODYR1,MODYR2
C     COMMON/SP1COM/SP1,SPAV
C
      INTEGER ALHFLG,TRKFLG,ALTFLG,ALTKOD(4,20),CY,VMFLAG,SP3FLG,
     *        NMHFLG,IDLFLG,UNFFLG,PRTFLG
      REAL SPD(3),VMTMIX(6),MYM(20,6),MYR(20,6),TF(20,6)
      REAL EFRETM(3,6),WTEDEF(7),LNKDTA(7),PROJID(20)
      EQUIVALENCE (SPD(1),LNKDTA(1))
C     DATA VMTMIX/.803,.058,.058,.045,.031,.005/
      DATA IREAD,IWRI/5,6/
      REAL COMCCC(6),IDLPTM(3,6),IDLWTD(7)
      DATA IPJOLD/0/
C
C
C
C
C
      IREJN=I1
      CY=I2
      TEMP=X1
      PCCO=X2
      PCHS=X3
      PCCC=X4
      VMTMIX(1)=A21
      VMTMIX(2)=A22
      VMTMIX(3)=A23
      VMTMIX(4)=A24
      VMTMIX(5)=A25
```

```
      VMTMIX(6)=A26                                                     SUP00410
      ALHFLG=I5                                                         SUP00420
      TRKFLG=I6                                                         SUP00430
      IMFLG=I7                                                          SUP00440
      SPD(1)=A1                                                         SUP00450
      SPD(2)=A1                                                         SUP00460
      SPD(3)=A1                                                         SUP00470
C                                                                       SUP00480
C                                                                       SUP00490
      IF(INIFLG.EQ.0) GO TO 100                                         SUP00500
      MYMRFG=0                                                          SUP00510
      MSFLG=1                                                           SUP00520
      ALTFLG=0                                                          SUP00530
      SP3FLG=1                                                          SUP00540
      NMHFLG=0                                                          SUP00550
      IDLFLG=1                                                          SUP00560
      UNFFLG=0                                                          SUP00570
      ICEVFG=1                                                          SUP00580
      IFORM=1                                                           SUP00590
      PRTFLG=0                                                          SUP00600
      IPRTSW=0                                                          SUP00610
      INFLG=0                                                           SUP00620
      IF(MYMRFG.EQ.1) READ(2,50) MYM,MYR                                SUP00630
   50 FORMAT(20F4.1)                                                    SUP00640
      CALL TFCALX                                                       SUP00650
100   CONTINUE                                                          SUP00660
C                                                                       SUP00670
C**CHECK CALENDAR YEAR                                                  SUP00680
      IF(CY.GE.70) GO TO 505                                            SUP00690
      WRITE(IWRI,785)                                                   SUP00700
785   FORMAT(' ***ERROR: CALENDAR YEAR RANGE IS 70 THRU 99')            SUP00710
      STOP                                                              SUP00720
505   CONTINUE                                                          SUP00730
      ICYPR = CY                                                        SUP00740
      IF(CY.GT.95) CY = 95                                              SUP00750
C                                                                       SUP00760
C**CHECK AMBIENT TEMPERATURE                                            SUP00770
      IF(TEMP.GE.0.0.AND.TEMP.LE.110.) GO TO 697                        SUP00780
      WRITE(IWRI,787)                                                   SUP00790
787   FORMAT(' ***ERROR: VALID TEMPERATURE RANGE IS 0-110 DEG.(F)')     SUP00800
```

```
      STOP                                                              SUP00810
697   CONTINUE                                                          SUP00820
C                                                                       SUP00830
C**CHECK INPUT SPEED(S)                                                 SUP00840
      IF(SP3FLG.EQ.1) GO TO 650                                         SUP00850
      IF(SP1.GT.0.) GO TO 620                                           SUP00860
      WRITE(IWRI,786)                                                   SUP00870
786   FORMAT(' ***ERROR: SPEED MUST BE POSITIVE')                       SUP00880
      STOP                                                              SUP00890
620   IF(SP1.GT.60.) SP1 = 60.                                          SUP00900
      IF(SP1.LT.5.) WRITE(IWRI,788)                                     SUP00910
788   FORMAT(' **WARNING: AVG. ROUTE SPEED LESS THAN 5 M.P.H.')         SUP00920
      GO TO 695                                                         SUP00930
C                                                                       SUP00940
650   CONTINUE                                                          SUP00950
      IF(SPD(1).GT.0.0.AND.SPD(2).GT.0.0.AND.SPD(3).GT.0.) GO TO 670    SUP00960
      WRITE(IWRI,786)                                                   SUP00970
      STOP                                                              SUP00980
670   CONTINUE                                                          SUP00990
      DO 671 I=1,3                                                      SUP01000
      IF(SPD(I).LT.5.) WRITE(IWRI,788)                                  SUP01010
671   IF(SPD(I).GT.60.) SPD(I) = 60.                                    SUP01020
C                                                                       SUP01030
695   CONTINUE                                                          SUP01040
C                                                                       SUP01050
C**END SPEED CHECK                                                      SUP01060
C                                                                       SUP01070
      IF(SP3FLG.EQ.1) GO TO 520                                         SUP01080
C**HERE TO DECOMPOSE SINGLE ENTERED SPEED INTO BAG SPEEDS               SUP01090
C     SPD(1) = SP1*1.30                                                 SUP01100
C     SPD(2) = SP1*.825                                                 SUP01110
C     SPD(3) = SP1*1.30                                                 SUP01120
      SPD(1) = SP1                                                      SUP01130
      SPD(2) = SP1                                                      SUP01140
      SPD(3) = SP1                                                      SUP01150
C     SPD(1) = SP1*(1.3777386 - (PCCC*.30/79.42))                       SUP01160
C     SPD(2) = SP1*(.75935 + (PCHS*.175/72.72))                         SUP01170
C     SPD(3) = SPD(1)                                                   SUP01180
C                                                                       SUP01190
C     PCHSNC = PCCC - PCCO + PCHS                                       SUP01200
```

```
C       PCHSTB = 100. - PCCC - PCHS                                     SUP01210
C       SPD(1) = SP1*(1.41114406 - .0041114406*PCCO)                    SUP01220
C       SPD(2) = SP1*(.6162276026 + .003837724*PCHSTB)                  SUP01230
C       SPD(3) = SP1*(1.4492449 - .0044902449*PCHSNC)                   SUP01240
520     CONTINUE                                                        SUP01250
C                                                                       SUP01260
C                                                                       SUP01270
        IF(IREJN.NE.IRJOLD) CALL INITEX(NMYALT,ALTKOD)                  SUP01280
        IF(IREJN.NE.IRJOLD.AND.INFLG.EQ.1) CALL LDVIMX                  SUP01290
        IRJOLD=IREJN                                                    SUP01300
        CALL EFCALX(CY,LNKDTA,VMTMIX,EFRETM,WTEDEF,COMCCC,IDLRTM,IDLWTD, SUP01310
       &INIFLG)                                                         SUP01320
        IF(IPRTSW.EQ.1) CALL OUTPUT(ICYPR,EFRETM,WTEDEF,COMCCC,IDLRTM,  SUP01330
       &IDLWTD)                                                         SUP01340
        EMCRUZ=WTEDEF(2)                                                SUP01350
        EMIDLE=IDLWTD(2)                                                SUP01360
        RETURN                                                          SUP01370
        END                                                             SUP01380
```

```
      SUBROUTINE OUTPUT(CY,EFRETM,WTEDEF,COMCCC,IDLRTM,IDLWTD)
C
      COMMON/FLGCOM/ALTFLG,ALHFLG,TRKFLG,IRDFLG,SP3FLG,NMHFLG,IDLFLG,
     *              UNFFLG,MYMRFG,IMFLG,ICEVFG,PRTFLG,IFORM
      COMMON/REGCOM/IREJN
      COMMON/BEFCOM/BEF
      COMMON/LNKCOM/SPD,TEMP,PCCO,PCHS,PCCC,VMTMIX
      COMMON/ALTCOM/NMYALT,ALTKOD
      COMMON/MYMCOM/MYM,MYR,TF
      COMMON/JUNK1/AC,XLOAD,TRAILR,ABSHUM
      COMMON/JUNK2/HGWGT,HDWGT,HGCID,HDCID
      COMMON/RET1/ALHRET
      COMMON/RET2/TRKRET(20,3,2)
      COMMON/IMCOM/ICYIM,ISTRIN,IMTFLG,MODYR1,MODYR2
      COMMON/SP1COM/SP1,SPAV
C
      INTEGER ALHFLG,TRKFLG,ALTFLG,ALTKOD(4,20),CY,VMFLAG,SP3FLG,
     *        NMHFLG,IDLFLG,UNFFLG,MYMRFG,PRTFLG
      REAL VMTMIX(6),MYM(20,6),MYR(20,6),PNAM1(3),PNAM2(3),PNAM3(3)
      REAL SPD(3)
      REAL EVPNAM(3),EXHHDR(9),IDLNM1(3),IDLNM2(3),IDLNM3(3)
      REAL IDLRTM(3,6),IDLWTD(3),PNAMH(2)
      REAL IDLHDR(8),IDLHD2(4),REGN1(3),REGN2(3)
      REAL COMPEF(20,3,6)
      REAL ALHRET(20,4,3)
      REAL A(20,3,3),U(20,3,3),L(3,3),H(3),XLOAD(3)
      REAL EFRETM(3,6),WTEDEF(3),LNKDTA(7),PROJID(20)
      REAL COMCCC(6),TF(20,6)
      REAL BEF(20,26,3,6)
      REAL YESNO(2)
      EQUIVALENCE (SPD(1),LNKDTA(1))
C
      DATA PROJID/'EMIS','SION',' FAC','TOR ','CALC',
     &'ULAT','ION ','    ','SUBR','OUTI',
     &'NE S','UP8','(MOD','ILE1',' PRO',
     &'GRAM',' MOD','IFIE','D)  ','    '/
      DATA PNAMH/'NON-','METH'/
      DATA PNAM1/'   T','EXHA','EXHA'/
      DATA PNAM2/'OTAL','UST ','UST '/
      DATA PNAM3/' HC ',' CO ','NOX '/
```

```
      DATA IDLNM1/'   I','   I','   I'/                                OUT00410
      DATA IDLNM2/'DLE ','DLE ','DLE '/                               OUT00420
      DATA IDLNM3/' HC ',' CO ','NOX '/                               OUT00430
      DATA EVFNAM/'   *','EVAP',' HC '/                               OUT00440
      DATA IWRI,IWRUNF/6,9/                                           OUT00450
      DATA EXHHDR/'COMP','OSIT','E EM','ISSI','ON F','ACTO',          OUT00460
     *     'RS (','GM/M','ILE)'/                                      OUT00470
      DATA IDLHDR/'IDLE',' EMI','SSIO','N FA','CTOR','S (G',          OUT00480
     *   'M/MI','N)  '/                                               OUT00490
      DATA IDLHD2/'    ','  CO','RREC','TED '/                        OUT00500
      DATA REGN1/'49-S','CALI','HI-A'/                                OUT00510
      DATA REGN2/'TATE','F.  ','LT. '/                                OUT00520
      DATA DASH/'----'/                                               OUT00530
      DATA INITFL/1/                                                  OUT00540
      DATA YESNO/'NO  ','YES '/                                       OUT00550
C                                                                     OUT00560
C*********************************************************************OUT00570
      IF(NMHFLG.EQ.0) GO TO 77                                        OUT00580
      PNAM1(1) = PNAMH(1)                                             OUT00590
      PNAM2(1) = PNAMH(2)                                             OUT00600
77    CONTINUE                                                        OUT00610
C                                                                     OUT00620
      IF(INITFL.NE.1) GO TO 175                                       OUT00630
C**HERE FIRST TIME                                                    OUT00640
      INITFL=0                                                        OUT00650
      IF(IFORM.EQ.0) WRITE(IWRI,7011) PROJID                          OUT00660
      IF(IFORM.EQ.1) WRITE(IWRI,701) PROJID                           OUT00670
      IF(IFORM.EQ.1)                                                  OUT00680
     *WRITE(IWRI,7017) PNAM1(1),PNAM2(1),PNAM3(1)                     OUT00690
7017  FORMAT('0',2X,'*',2A4,A3,' EMISSION FACTORS INCLUDE ',          OUT00700
     *  'EVAP. HC EMISSION FACTORS')                                  OUT00710
C                                                                     OUT00720
      IF(IFORM.EQ.1) WRITE(IWRI,710)                                  OUT00730
175   CONTINUE                                                        OUT00740
C                                                                     OUT00750
      IF(IFORM.EQ.0) GO TO 698                                        OUT00760
C*********************************************************************OUT00770
C                                                                     OUT00780
      WRITE(IWRI,7029)                                                OUT00790
      IF(SF3FLG.EQ.1)                                                 OUT00800
```

```
     *WRITE(IWRI,703) CY,TEMP,VMTMIX,REGN1(IREJN),REGN2(IREJN),SPD,SPAV,OUT00810
     *  PCCO,PCHS,PCCC                                                  OUT00820
      IF(SP3FLG.EQ.0)                                                   OUT00830
     *WRITE(IWRI,7035) CY,TEMP,VMTMIX,REGN1(IREJN),REGN2(IREJN),SP1,SPD,OUT00840
     *  SPAV,PCCO,PCHS,PCCC                                             OUT00850
      IF(ALHFLG.EQ.1) WRITE(IWRI,704) AC,XLOAD,TRAILR,ABSHUM            OUT00860
      IF(TRKFLG.EQ.1) WRITE(IWRI,705) HGWGT,HDWGT,HGCID,HDCID           OUT00870
      IF(IMFLG.EQ.1) WRITE(IWRI,7051) ICYIM,ISTRIN,YESNO(IMTFLG+1)      OUT00880
      IF(IMFLG.EQ.1) WRITE(IWRI,7052) MODYR1,MODYR2                     OUT00890
C                                                                       OUT00900
      WRITE(IWRI,7059) EXHHDR                                           OUT00910
      WRITE(IWRI,706)                                                   OUT00920
C                                                                       OUT00930
      IF(PRTFLG.EQ.0.OR.PRTFLG.EQ.1)                                    OUT00940
     *WRITE(IWRI,707) PNAM1(1),PNAM2(1),PNAM3(1),(EFRETM(1,IM),IM=1,6), OUT00950
     *  WTEDEF(1)                                                       OUT00960
      IF(ICEVFG.EQ.1) WRITE(IWRI,708) EVPNAM,COMCCC,DASH                OUT00970
      IF(PRTFLG.EQ.0.OR.PRTFLG.EQ.2)                                    OUT00980
     *WRITE(IWRI,707) PNAM1(2),PNAM2(2),PNAM3(2),(EFRETM(2,IM),IM=1,6), OUT00990
     *  WTEDEF(2)                                                       OUT01000
      IF(PRTFLG.EQ.0.OR.PRTFLG.EQ.3)                                    OUT01010
     *WRITE(IWRI,707) PNAM1(3),PNAM2(3),PNAM3(3),(EFRETM(3,IM),IM=1,6), OUT01020
     *  WTEDEF(3)                                                       OUT01030
C                                                                       OUT01040
      IF(IDLFLG.EQ.0) GO TO 200                                         OUT01050
      WRITE(IWRI,709) IDLHD2,IDLHDR                                     OUT01060
      WRITE(IWRI,707) (IDLNM1(IQ),IDLNM2(IQ),IDLNM3(IQ),(IDLRTM(IQ,IM), OUT01070
     *  IM=1,6),IDLWTD(IQ),IQ=1,3)                                      OUT01080
200   CONTINUE                                                          OUT01090
C                                                                       OUT01100
      WRITE(IWRI,710)                                                   OUT01110
C                                                                       OUT01120
      GO TO 899                                                         OUT01130
C                                                                       OUT01140
C*********************************************************************OUT01150
C                                                                       OUT01160
888   CONTINUE                                                          OUT01170
         IPP1=PRTFLG                                                    OUT01180
         IPP2=PRTFLG                                                    OUT01190
         IF(PRTFLG.EQ.0) IPP1=1                                         OUT01200
```

```
      IF(PRTFLG.EQ.0) IPP2=3                                            OUT01210
      DO 1203 IPNX=IPP1,IPP2                                            OUT01220
      WRITE(IWRI,79080) IREJN,CY,SPD,TEMP,PCCO,PCHS,PCCC,IPNX,          OUT01230
     *  (EFRETM(IPNX,IM),IM=1,6),VMTMIX,WTEDEF(IPNX)                    OUT01240
1203  CONTINUE                                                          OUT01250
79080 FORMAT(1X,I1,1X,I2,1X,F4.1,'/',F4.1,'/',F4.1,1X,                  OUT01260
     *  F3.0,1X,F5.1,1X,F5.1,1X,F5.1,1X,I1,6F6.1,6F6.3,F7.2)            OUT01270
C                                                                       OUT01280
899   CONTINUE                                                          OUT01290
C                                                                       OUT01300
C                                                                       OUT01310
C***********************************************************************OUT01320
C                                                                       OUT01330
      IF(UNFFLG.EQ.1) WRITE(IWRUNF) EFRETM,WTEDEF                       OUT01340
C                                                                       OUT01350
C***********************************************************************OUT01360
701   FORMAT('1',20A4/)                                                 OUT01370
7011  FORMAT('1',20A4)                                                  OUT01380
C                                                                       OUT01390
7029  FORMAT(32X,'VEH. TYPE:  LDV   LDT1  LDT2  HDG   HDD   MC')        OUT01400
C                                                                       OUT01410
707   FORMAT('  CAL. YEAR: 19',I2,5X,'TEMP: ',F4.1,                    OUT01420
     *  '(F)',8X,5(F5.3,'/'),F5.3,                                      OUT01430
     *  /1X,' REGION: ',2A4,4X,F4.1,'/',F4.1,'/',F4.1,                  OUT01440
     *  ' MPH ','(',F4.1,')',7X,' ',F5.1,'/',F5.1,'/',F5.1 )            OUT01450
C                                                                       OUT01460
7035  FORMAT('  CAL. YEAR: 19',I2,5X,'TEMP: ',F4.1,                    OUT01470
     *  '(F)',8X,5(F5.3,'/'),F5.3,                                      OUT01480
     *  /1X,' REGION: ',2A4,4X,F4.1,':',F4.1,'/',F4.1,'/',F4.1,         OUT01490
     *  ' MPH ','(',F4.1,')',3X,' ',F5.1,'/',F5.1,'/',F5.1 )            OUT01500
704   FORMAT('   AC:',F6.2,2X,'XLOAD:',3F7.2,2X,'TRAILR:',              OUT01510
     *  F7.2,2X,'ABSHUM:',F7.2)                                         OUT01520
C                                                                       OUT01530
705   FORMAT('   HGWGT:',F7.0,2X,'HDWGT:',F7.0,2X,'HGCID:',             OUT01540
     *  F7.2,2X,'HDCID:',F7.2)                                          OUT01550
C                                                                       OUT01560
7051  FORMAT('   LDV I/M PROGRAM STARTING IN 19',I2,                    OUT01570
     *  ', STRINGENCY LEVEL ',I2,'%, MECH. TRAINING: ',A4)              OUT01580
C                                                                       OUT01590
7052  FORMAT('   I/M PROG. BENEFITS APPLY ONLY TO MODEL YEARS 19',      OUT01600
```

```
     *  I2,' THROUGH 19',I2)                                          OUT01610
705   FORMAT('0',22X,9A4)                                             OUT01620
C                                                                     OUT01630
706   FORMAT(' ',15X,'LDV       LDT1      LDT2      HDG       HDD    ',  OUT01640
     *'      MC      ALL MODES')                                      OUT01650
C                                                                     OUT01660
707   FORMAT((1X,2A4,A3,':',6(F7.2,2X),1X,F8.2))                      OUT01670
C                                                                     OUT01680
708   FORMAT(1X,2A4,A3,':',6(F7.2,2X),5X,A4)                          OUT01690
C                                                                     OUT01700
709   FORMAT('0',16X,4A4,8A4)                                         OUT01710
C                                                                     OUT01720
710   FORMAT(' ----------------------------------------',              OUT01730
     *'----------------------------------------')                     OUT01740
C                                                                     OUT01750
C                                                                     OUT01760
C     DEBUG SUBCHK                                                    OUT01770
      RETURN                                                          OUT01780
      END                                                             OUT01790
```

```
      SUBROUTINE EFCALX(CY,LNKDTA,MS,EFRETM,WTEDEF,CCRETM,IDLRTM,IDLWTD,EFC00010
     * INIFLG)                                                          EFC00020
C                                                                       EFC00030
      COMMON/FLGCOM/ALTFLG,ALHFLG,TRKFLG,IRDFLG,SP3FLG,NMHFLG,IDLFLG,   EFC00040
     *              UNFFLG,MYMRFG,IMFLG,ICEVFG,PRTFLG,IFORM             EFC00050
      COMMON/CEVCOM/CCEV                                                EFC00060
      COMMON/REGCOM/IREJN                                               EFC00070
      COMMON/BEFCOM/BEF                                                 EFC00080
      COMMON/IDLCOM/IDLBEF                                              EFC00090
      COMMON/MYMCOM/MYM,MYR,TF                                          EFC00100
C                                                                       EFC00110
      INTEGER MYP , P ,  CYP                                            EFC00120
      INTEGER ALHFLG,TRKFLG,ALTFLG,ALTKOD(4,20),CY,MSFLG,SP3FLG,        EFC00130
     *        NMHFLG,IDLFLG,UNFFLG,MYMRFG,PRTFLG                        EFC00140
C                                                                       EFC00150
      REAL IDLBEF(20,26,3,6),IDLFAC(20,3,6),IDLCMP(3,6),IDLWTD(3)       EFC00160
      REAL IDLRTM(3,6)                                                  EFC00170
      REAL COMPCC(20,6),CCRETM(6)                                       EFC00180
      REAL CFRET(20,3,6)                                                EFC00190
      REAL      BEF(20,26,3,6),      CCEV(45,6),                        EFC00200
     *          CEFLDV(3),           CEFLT1(3),                         EFC00210
     *          CEFLT2(3),           CEFHDG(3),                         EFC00220
     *          CEFHDD(3),           CEFMCC(3),                         EFC00230
     *          EFRETM(3,6),         COMPEF(20,3,6),                    EFC00240
     *          COMPMY(3,6),         LNKDTA(7),                         EFC00250
     *          MYM(20,6),           MYR(20,6),                         EFC00260
     *          MS(6),               PCCO,                              EFC00270
     *          PCCC,                PCHS,                              EFC00280
     *          SPD(3),              TEMP,                              EFC00290
     *          TF(20,6),            TFNORM(6),                         EFC00300
     *          WTEDEF(3)                                               EFC00310
      DIMENSION R1(20,3,6),R2(20,3,6),XISPD(3)                          EFC00320
C                                                                       EFC00330
C                                                                       EFC00340
C***********************************************************************EFC00350
C                                                                       EFC00360
      SPD(1)  = LNKDTA(1)                                               EFC00370
      SPD(2)  = LNKDTA(2)                                               EFC00380
      SPD(3)  = LNKDTA(3)                                               EFC00390
      TEMP    = LNKDTA(4)                                               EFC00400
```

```
      PCCO     = LNKDTA(5)                                          EFC00410
      PCHS     = LNKDTA(6)                                          EFC00420
      PCCC     = LNKDTA(7)                                          EFC00430
C                                                                   EFC00440
C                                                                   EFC00450
            CALL BIGCFX(CY,TEMP,PCCO,PCHS,PCCC,SPD,CFRET,INIFLG)    EFC00460
C                                                                   EFC00470
      XISPD(1)=5.0                                                  EFC00480
      XISPD(2)=5.0                                                  EFC00490
      XISPD(3)=5.0                                                  EFC00500
      XITEMP=75.0                                                   EFC00510
      XIPCCO=0.0                                                    EFC00520
      XIPCHS=0.0                                                    EFC00530
      XIPCCC=0.0                                                    EFC00540
      INIFLG=0                                                      EFC00550
      CALL BIGCFX(CY,TEMP,PCCO,PCHS,PCCC,XISPD,R1,INIFLG)           EFC00560
      CALL BIGCFX(CY,XITEMP,XIPCCO,XIPCHS,XIPCCC,XISPD,R2,INIFLG)   EFC00570
C                                                                   EFC00580
      CYP = CY - 69                                                 EFC00590
C                                                                   EFC00600
C                                                                   EFC00610
      DO 500 P = 1,3                                                EFC00620
               WTEDEF(P) = 0.0                                      EFC00630
               IDLWTD(P) = 0.0                                      EFC00640
         DO 400 M = 1,6                                             EFC00650
               COMPMY(P,M) = 0.0                                    EFC00660
               IDLCMP(P,M) = 0.0                                    EFC00670
      IF(P.EQ.1) CCRETM(M)=0.                                       EFC00680
            DO 300 I = 1,20                                         EFC00690
C                                                                   EFC00700
      MYP = CY -(20-I) - 50                                         EFC00710
C                                                                   EFC00720
               IF (P.EQ.1)  GO TO 333                               EFC00730
               GO TO 444                                            EFC00740
C                                                                   EFC00750
333   COMPEF(I,P,M)=(BEF(I,CYP,P,M)*CFRET(I,P,M)+ CCEV(MYP,M))*TF(I,M)  EFC00760
      COMPCC(I,M)   =CCEV(MYP,M)*TF(I,M)                            EFC00770
      GO TO 555                                                     EFC00780
444   COMPEF(I,P,M)=(BEF(I,CYP,P,M)*CFRET(I,P,M)              )*TF(I,M)  EFC00790
C                                                                   EFC00800
```

```
555   CONTINUE                                                          EFC00810
C                                                                       EFC00820
      IDLFAC(I,P,M)=IDLBEF(I,CYF,P,M)*R1(I,P,M)*TF(I,M)/R2(I,P,M)       EFC00830
C                                                                       EFC00840
      COMPMY(P,M) = COMPMY(P,M) + COMPEF(I,P,M)                         EFC00850
      IDLCMP(P,M) = IDLCMP(P,M) + IDLFAC(I,P,M)                         EFC00860
      IF(P.EQ.1) CCRETM(M) = CCRETM(M) + COMPCC(I,M)                    EFC00870
C                                                                       EFC00880
300         CONTINUE                                                    EFC00890
      WTEDEF(P) = WTEDEF(P) + COMPMY(P,M)*MS(M)                         EFC00900
      IDLWTD(P) = IDLWTD(P) + IDLCMP(P,M)*MS(M)                         EFC00910
400      CONTINUE                                                       EFC00920
C                                                                       EFC00930
500   CONTINUE                                                          EFC00940
C                                                                       EFC00950
      DO 600 M = 1,6                                                    EFC00960
      DO 600 P = 1,3                                                    EFC00970
        EFRETM(P,M) = COMPMY(P,M)                                       EFC00980
        IDLRTM(P,M) = IDLCMP(P,M)                                       EFC00990
600   CONTINUE                                                          EFC01000
C                                                                       EFC01010
C                                                                       EFC01020
      RETURN                                                            EFC01030
C                                                                       EFC01040
      END                                                               EFC01050
```

```
      SUBROUTINE BIGCFX(CY,T,PCCO,PCHS,PCCC,SPD,CFRET,INITFL)
C
      COMMON/FLGCOM/ALTFLG,ALHFLG,TRKFLG,IRDFLG,SP3FLG,NMHFLG,IDLFLG,
     *              UNFFLG,MYMRFG,IMFLG,ICEVFG,PRTFLG,IFORM
      COMMON/ALTCOM/NMYALT,ALTKOD
      COMMON/REGCOM/IREJN
      COMMON/JUNK1/AC,XLOAD,TRAILR,ABSHUM
      COMMON/JUNK2/HGWGT,HDWGT,HGCID,HDCID
      COMMON/RET1/ALHRET(20,4,3)
      COMMON/RET2/TRKRET(20,3,2)
      COMMON/SP1COM/SP1,SPAV
      COMMON/MYMCOM/MYM,MYR,TF
C
      INTEGER ALHFLG,TRKFLG,ALTFLG,ALTKOD(4,20),CY,MSFLG,SP3FLG,
     *        NMHFLG,IDLFLG,UNFFLG,MYMRFG,PRTFLG
      INTEGER INITFL,IEQNAR(45,3,3,4),HDINDX(45,3)
      INTEGER CY,G(45,3,4)
C IN G, 1ST INDEX IS MY, 2ND IS REGION, AND 3RD IS MODE
C IN IEQNAR, 2ND INDEX IS POLLUTANT, 3RD IS REGION, 4TH IS MODE
C IN HDINDX, 2ND INDEX IS REGION (NOT FUNCTION OF POL. OR MODE)
C
      REAL COMCCC(6),A(20,3,3),U(20,3,3),
     *  L(3,3),H(3)
      REAL XLOAD(3)
      REAL COMPEF(20,3,6)
      REAL W,T,SPD(3),SP(3),CFRET(20,3,6)
      REAL D1(4,3),D2(4,3)
      REAL MIDDLE,LEFT
      REAL SPBACK(18,3,3),SPPOT(18,3,3)
      REAL ATR(4,3,2),BTR(4,3,2),CTR(4,3,2)
      REAL C(4,8,3)
      REAL CUMMIL(20,6),MYM(20,6),MYR(20,6),TF(20,6)
      INTEGER MAXAGE(4)
C
      DATA MAXAGE/19,19,19,19/
C
C*****************************************************************
C
C**COEFFICIENTS IN SPEED/TEMP/COLD START FACTOR: EQN,INDEX,POL.
      DATA C/
```

```
     *   2.931000, 2.931000, 2.433900, 1.993400,                          BIG00410
     *  -0.014779,-0.014779,-0.023591,-0.022269,                          BIG00420
     *   0.673,-2.410, 0.623,-0.032, 0.569, 0.863, 0.301, 0.445,          BIG00430
     *   4.750, 2.430, 1.110, 0.497, 0.393, 0.555, 0.284, 0.357,          BIG00440
     *   5.690, 2.610, 1.050, 0.243, 0.471, 0.597, 0.270, 0.175,          BIG00450
     *   5.654800, 5.654800, 5.546000, 4.239100,                          BIG00460
     *  -0.015965,-0.015965,-0.028945,-0.017522,                          BIG00470
     *  -14.74,-33.89, 11.29, -0.20,  9.62,  9.77,                        BIG00480
     *    4.24,  6.99, 42.84, 25.26, 15.85,  4.12,                        BIG00490
     *    5.76,  4.71,  2.34,  2.20, 57.57, 35.90,                        BIG00500
     *   21.17,  3.96,  7.74,  6.70,  3.13,  2.12,                        BIG00510
     *  -0.10E+03,-0.10E+03,-0.10E+03,-0.10E+03,                          BIG00520
     *   0.0  , 0.0  , 0.0  , 0.0  , 1.140, 1.160,                        BIG00530
     *   3.260, 3.050, 0.0  , 0.0  , 0.335, 0.318,                        BIG00540
     *   1.250, 1.260, 2.990, 2.880, 0.0  , 0.0  ,                        BIG00550
     *   0.184, 0.180, 0.810, 0.800, 1.890, 2.010,                        BIG00560
     *   0.0  , 0.0  , 0.116, 0.126  /                                    BIG00570
C                                                                         BIG00580
C**COEFFICIENTS IN DENOMINATOR OF SPEED/TEMP/COLD START FACTOR            BIG00590
      DATA D1/5.67,2.8,1.38,.54,56.43,36.4,23.7,6.98,1.,1.,2.47,2.46/     BIG00600
      DATA D2/.47,.64,.28,.28,7.59,6.79,3.14,3.14,0.,0.,.18,.18/          BIG00610
C                                                                         BIG00620
C**SPEED CORRECTION FACTOR INDEX: CAL. YR., REGION, MODE                  BIG00630
      DATA G/15*2, 2, 2, 4, 5, 6, 7,14,17,17,18,18,18,18*18,              BIG00640
     *       15*2, 3, 3, 4, 5, 6, 7,13,16,16,18,18,18,18*18,             BIG00650
     *       15*1, 1, 1, 8, 9,10,11,12,15,15,18,18,18,18*18,             BIG00660
     *                                                                   BIG00670
     *       15*2, 2, 2, 4, 5, 6, 7,14,17,17,18,18,18,18*18,             BIG00680
     *       15*2, 2, 2, 4, 5, 6, 7,13,16,16,18,18,18,18*18,             BIG00690
     *       15*1, 1, 1, 8, 9,10,11,12,15,15,18,18,18,18*18,             BIG00700
     *                                                                   BIG00710
     *       28*14,17*18,                                                BIG00720
     *       27*13,18*18,                                                BIG00730
     *       28*12,17*18,                                                BIG00740
     *                                                                   BIG00750
     *       15*2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,18*17,             BIG00760
     *       15*2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,16*16,             BIG00770
     *       15*1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,18*15 /            BIG00780
C                                                                         BIG00790
      DATA IEQNAR/17*1,7*2,5*3,16*4,  17*1,7*2,5*3,16*4,                  BIG00800
```

```
     *            17*1,7*2,2*3,19*4,  15*1,9*2,     21*4,                BIG00810
     *            15*1,9*2,     21*4,  15*1,9*2,     21*4,               BIG00820
     *            17*1,7*2,5*3,16*4,  17*1,7*2,5*3,16*4,                 BIG00830
     *            17*1,7*2,2*3,19*4,                                     BIG00840
     *            17*1,7*2,21*3,      17*1,7*2,21*3,                     BIG00850
     *            17*1,7*2,21*3,      17*1,7*2,     21*4,                BIG00860
     *            17*1,7*2,     21*4, 17*1,7*2,     21*4,                BIG00870
     *            17*1,7*2,21*3,      17*1,7*2,21*3,                     BIG00880
     *            17*1,7*2,21*3,                                         BIG00890
     *            28*2,17*3,          28*2,17*3,                         BIG00900
     *            28*2,17*3,          27*2,18*4,                         BIG00910
     *            27*2,18*4,          27*2,18*4,                         BIG00920
     *            28*2,17*3,          28*2,17*3,                         BIG00930
     *            28*2,17*3,                                             BIG00940
     *            27*1,18*2,27*1,18*2,27*1,18*2,                         BIG00950
     *            27*1,18*2,27*1,18*2,27*1,18*2,                         BIG00960
     *            27*1,18*2,27*1,18*2,27*1,18*2  /                       BIG00970
C                                                                       BIG00980
      DATA HDINDX/19*1,4*2,5*3,17*4,                                    BIG00990
     *            19*1,4*2,1*3,21*4,                                    BIG01000
     *            19*1,4*2,5*3,17*4   /                                 BIG01010
C                                                                       BIG01020
C                                                                       BIG01030
C*********************************************************************BIG01040
C                                                                       BIG01050
      IF(INITFL.NE.1) GO TO 177                                         BIG01060
C  HERE 1ST TIME                                                        BIG01070
      INITFL = 0                                                        BIG01080
      SPB(1) = 26.0                                                     BIG01090
      SPB(2) = 16.0                                                     BIG01100
      SPB(3) = 26.0                                                     BIG01110
      CALL SPFCLX(SPB,SPBOT)                                            BIG01120
C                                                                       BIG01130
C                                                                       BIG01140
      CALL GETCUM(CUMMIL)                                               BIG01150
C                                                                       BIG01160
C                                                                       BIG01170
C*********************************************************************BIG01180
C                                                                       BIG01190
177   CONTINUE                                                          BIG01200
```

```
C                                                                       BIG01210
      CALL SPECLX(SPD,SPBACK)                                           BIG01220
C                                                                       BIG01230
C                                                                       BIG01240
      IF(ALHFLG.EQ.1) CALL ALUH(CY,AC,XLOAD,TRAILR,PCCO,PCCC,AFSHUM)    BIG01250
C                                                                       BIG01260
C***********************************************************************BIG01270
C                                                                       BIG01280
      FCO = PCCO*.01                                                    BIG01290
      FHC = PCHS*.01                                                    BIG01300
      FCC = PCCC*.01                                                    BIG01310
C                                                                       BIG01320
C**FRACTION COLD OP(NON-CAT): FCO                                       BIG01330
C**FRACTION HOT START(CAT):  FHC                                        BIG01340
C**FRACTION COLD OP(CAT)  :   FCC                                       BIG01350
C**FRACTION HOT START(NON-CAT): FHO                                     BIG01360
      FHO = (FCC-FCO) + FHC                                             BIG01370
C                                                                       BIG01380
      DO 500 I = 1,20                                                   BIG01390
C                                                                       BIG01400
         MY = CY - (20-I)                                               BIG01410
         MYC = MY - 50                                                  BIG01420
         IAGE = CY - MY                                                 BIG01430
         AGE = IAGE                                                     BIG01440
C                                                                       BIG01450
C***********************                                                BIG01460
      IF(MYC.LE.0) STOP                                                 BIG01470
C>>>>>>>>>>>>>>>>>>>>>>>>>                                              BIG01480
C                                                                       BIG01490
      DO 500 IMODE = 1,4                                                BIG01500
C                                                                       BIG01510
      IMOD4 = IMODE                                                     BIG01520
      IF(IMODE.EQ.4) IMOD4 = 6                                          BIG01530
      IAGEM = IAGE                                                      BIG01540
         MAX = MAXAGE(IMODE)                                            BIG01550
      IF(IAGEM.GT.MAX) IAGEM = MAX                                      BIG01560
C                                                                       BIG01570
      IF(IAGEM.EQ.0)VMTAGE=0.0                                          BIG01580
      IF(IAGEM.GT.0)VMTAGE=CUMMIL(IAGEM,IMOD4)*0.0001                   BIG01590
C                                                                       BIG01600
```

```
         IGX = G(MYC,IPEJN,IMODE)                                       BIG01610
C                                                                       BIG01620
      DO 500 IP = 1,3                                                   BIG01630
C                                                                       BIG01640
C**DEFAULT                                                              BIG01650
C                                                                       BIG01660
      IEQN = IEQNAR(MYC,IP,IPEJN,IMODE)                                 BIG01670
C                                                                       BIG01680
C*****************************************                              BIG01690
C                                                                       BIG01700
      IF(ALTFLG.EQ.0) GO TO 777                                         BIG01710
C                                                                       BIG01720
      DO 450 ICH = 1,NMYALT                                             BIG01730
      IF(ALTKOD(1,ICH).EQ.MY) GO TO 877                                 BIG01740
450   CONTINUE                                                          BIG01750
      GO TO 550                                                         BIG01760
877   CONTINUE                                                          BIG01770
C**HERE FOR ALTERATION                                                  BIG01780
      IALT = ALTKOD(IP+1,ICH)                                           BIG01790
C                                                                       BIG01800
      IF(IALT.EQ.0) GO TO 777                                           BIG01810
      IF(IALT.EQ.1) IEQN = 3                                            BIG01820
      IF(IALT.GT.1) IEQN = 4                                            BIG01830
777   CONTINUE                                                          BIG01840
C                                                                       BIG01850
C*****************************************                              BIG01860
C                                                                       BIG01870
550   CONTINUE                                                          BIG01880
C                                                                       BIG01890
      FACL = FCO                                                        BIG01900
      FACM = FHO                                                        BIG01910
      IF(IEQN.LT.2) GO TO 888                                           BIG01920
      FACL = FCC                                                        BIG01930
      FACM = FHC                                                        BIG01940
888   FACR = (1.0 - FACL - FACM)                                        BIG01950
C                                                                       BIG01960
C*****************************************************************     BIG01970
C                                                                       BIG01980
      DENOM = D1(IEQN,IP) + D2(IEQN,IP)*VMTAGE                          BIG01990
C                                                                       BIG02000
```

```
      LEFT = FACL * (EXP(C(IEQN,1,IP)+T*C(IEQN,2,IP)) + C(IEQN,3,IP) +  BIG02010
     *  C(IEQN,4,IP)*VMTAGE) * SPBACK(2,1,IP)/SPBOT(2,1,IP)             BIG02020
C                                                                       BIG02030
C                                                                       BIG02040
      MIDDLE=FACM*(C(IEQN,5,IP)+C(IEQN,6,IP)*VMTAGE)                    BIG02050
     *     *SPBACK(IGX,3,IP)/SPBOT(IGX,3,IP)                            BIG02060
C                                                                       BIG02070
C                                                                       BIG02080
      RIGHT=FACR*(C(IEQN,7,IP)+C(IEQN,8,IP)*VMTAGE)                     BIG02090
     *      *SPBACK(IGX,2,IP)/SPBOT(IGX,2,IP)                           BIG02100
C                                                                       BIG02110
      TOP = LEFT + MIDDLE + RIGHT                                       BIG02120
C                                                                       BIG02130
C                                                                       BIG02140
      FACTOR = TOP/DENOM                                                BIG02150
C                                                                       BIG02160
      IF(ALHFLG.EQ.1) FACTOR = FACTOP*ALHRET(I,IMODE,IP)                BIG02170
C                                                                       BIG02180
      IMOD2 = IMODE                                                     BIG02190
C                                                                       BIG02200
      IF(IMODE.EQ.4) IMOD2 = 6                                          BIG02210
C                                                                       BIG02220
      CFRET(I,IP,IMOD2) = FACTOR                                        BIG02230
C                                                                       BIG02240
500   CONTINUE                                                          BIG02250
C                                                                       BIG02260
C***********************************************************************BIG02270
C                                                                       BIG02280
C                                                                       BIG02290
C   FOR MODES 4 AND 5                                                   BIG02300
C                                                                       BIG02310
C                                                                       BIG02320
      DATA ATR/ 2.42, 1.14, 1.56, 1.54,                                 BIG02330
     *          1.74, 1.54, 1.24, 0.62,                                 BIG02340
     *          .651, .812, .820, .764,                                 BIG02350
     *          0.94, 0.94,  .920, 1.20,                                BIG02360
     *          1.44, 1.44,  1.20, 1.44,                                BIG02370
     *          0.62, 0.62,  0.72, 1.00  /                              BIG02380
C                                                                       BIG02390
      DATA BTR/ -.153, -.065,  -.096, -.071,                            BIG02400
```

```
     *          -.117, -.007,  -.078, -.039,                          BIG02410
     *          .0074, .0094,  .0081, .0120,                          BIG02420
     *          -.055, -.055,  -.051, -.072,                          BIG02430
     *          -.092, -.092,  -.072, -.088,                          BIG02440
     *          -.043, -.043,  -.054, -.074  /                        BIG02450
C                                                                     BIG02460
      DATA CTR/ .0016, .0004, .0009, -.0003,                          BIG02470
     *          .0015, .0010, .0008, .0004,                           BIG02480
     *          .0000, .0000, .0000, .0000,                           BIG02490
     *          .0004, .0004, .0005, .0006,                           BIG02500
     *          .0010, .0010, .0006, .0008,                           BIG02510
     *          .0006, .0006, .0009, .0012  /                         BIG02520
C                                                                     BIG02530
      IF(TRKFLG.EQ.1) CALL TRKOPC(CY,HCWGT,HDWGT,HGCID,HDCID)         BIG02540
C                                                                     BIG02550
C     IF(SP3FLG.EQ.0) SPAV = SPD(1)/(1.3777386 - (PCCC*.30/79.42))    BIG02560
      IF(SP3FLG.EQ.0) SPAV = SP1                                      BIG02570
C     IF(SP3FLG.EQ.1) SPAV = SPD(1)/(1.41114406 - (PCCO*.0041114406))BIG02580
      IF(SP3FLG.EQ.1) SPAV = SPD(2)                                   BIG02590
C     IF(SP3FLG.EQ.1) SPAV = AMIN1(SPD(1),SPD(2))                     BIG02600
C***ELIMINATE TWO LINES ABOVE                                         BIG02610
C                                                                     BIG02620
C                                                                     BIG02630
      DO 700 IMODE = 4,5                                              BIG02640
      IM = IMODE-3                                                    BIG02650
         DO 700 IP = 1,3                                              BIG02660
C                                                                     BIG02670
      IMY1 = CY-19                                                    BIG02680
              DO 700 IMY = IMY1,CY                                    BIG02690
      IMYP = IMY - 50                                                 BIG02700
      IF(IMYP.LT.1) IMYP=1                                            BIG02710
C                                                                     BIG02720
              IX = 20-(CY-IMY)                                        BIG02730
C                                                                     BIG02740
          INDX = MDINDX(IMYP,IPEJN)                                   BIG02750
C                                                                     BIG02760
      IF(IP.EQ.3.AND.IMODE.EQ.4) GO TO 720                            BIG02770
C                                                                     BIG02780
      CPRET(IX,IP,IMODE) = EXP(ATR(INDX,IP,IM) +                      BIG02790
     *  BTR(INDX,IP,IM)*SPAV + CTR(INDX,IP,IM)*SPAV*SPAV)             BIG02800
```

```
C                                                                       BIG02810
      GO TO 688                                                         BIG02820
700   CONTINUE                                                          BIG02830
C                                                                       BIG02840
      CFRET(IX,IP,IMODE) = ATR(INDX,IP,IM) + BTR(INDX,IP,IM)*SFAV       BIG02850
C                                                                       BIG02860
688   CONTINUE                                                          BIG02870
C                                                                       BIG02880
      IF(TRKFLG.EQ.1) CFRET(IX,IP,IMODE) = CFRET(IX,IP,IMODE)*          BIG02890
     *                                     TRKRET(IX,IP,IM)             BIG02900
C                                                                       BIG02910
700   CONTINUE                                                          BIG02920
      RETURN                                                            BIG02930
C                                                                       BIG02940
C     DEBUG SUBCHK                                                      BIG02950
      END                                                               BIG02960
```

```
      SUBROUTINE INITEX( NMYALT , ALTKOD )                              INX00010
C                                                                       INX00020
      COMMON/FLGCOM/ALTFLG,ALHFLG,TRKFLG,IRDFLG,SP3FLG,NMHFLG,IDLFLG,   INX00030
     & UNFFLG,MYNRFG,ICEVFG,PRTFLG,IFCFM                                INX00040
      INTEGER ALHFLG,TRKFLG,ALTFLG,ALTKOD(4,20),CY,MSFLG,SP3FLG,        INX00050
     *        NMHFLG,IDLFLG,UNFFLG,MYNRFG                               INX00060
C                                                                       INX00070
      CALL BEFGEN                                                       INX00080
C                                                                       INX00090
      CALL CCEVAX                                                       INX00100
C                                                                       INX00110
      IF (ALTFLG.EQ.1) CALL EFALTX( NMYALT,ALTKOD )                     INX00120
C                                                                       INX00130
C                                                                       INX00140
      RETURN                                                            INX00150
C                                                                       INX00160
      END                                                               INX00170
```

```
      SUBROUTINE TFCALY                                                 TFC00010
C                                                                       TFC00020
      COMMON/MYMCOM/MYM,MYR,TF                                          TFC00030
      REAL TFNORM(6),MYM(20,6),MYR(20,6),TF(20,6)                       TFC00040
C                                                                       TFC00050
      DO 850 M = 1,6                                                    TFC00060
            TFNORM(M) = 0.0                                             TFC00070
         DO 810 I = 1,20                                                TFC00080
810         TFNORM(M) = TFNORM(M) + MYM(I,M)*MYR(I,M)                   TFC00090
         DO 820 I = 1,20                                                TFC00100
820         TF(21-I,M) = ( MYM(I,M)*MYR(I,M) )/TFNORM(M)                TFC00110
C                                                                       TFC00120
850   CONTINUE                                                          TFC00130
C                                                                       TFC00140
      RETURN                                                            TFC00150
C                                                                       TFC00160
C     DEBUG SUBCHK                                                      TFC00170
C                                                                       TFC00180
      END                                                               TFC00190
```

```
      SUBROUTINE SPECLX(SPARA,SPBACK)                                   SPF00010
      COMMON/ACOM/A                                                     SPF00020
      REAL SPARA(3),SPBACK(18,3,3),A(6,18,3)                            SPF00030
C                                                                       SPF00040
      DO 30 IP = 1,3                                                    SPF00050
         DO 20 IG = 1,18                                                SPF00060
               IGG = IG                                                 SPF00070
            DO 10 IS = 1,3                                              SPF00080
                  SUM = A(6,IGG,IP)                                     SPF00090
                  S = SPARA(IS)                                         SPF00100
               DO 5 I = 1,5                                             SPF00110
                SUM = SUM*S + A(6-I,IGG,IP)                             SPF00120
5              CONTINUE                                                 SPF00130
             SCF=EXP(SUM)                                               SPF00140
             IF(IP.EQ.3) SCF=SUM                                        SPF00150
             SPBACK(IG,IS,IP) = SCF                                     SPF00160
10          CONTINUE                                                    SPF00170
20       CONTINUE                                                       SPF00180
30    CONTINUE                                                          SPF00190
      RETURN                                                            SPF00200
C     DEBUG SUBCHK                                                      SPF00210
      END                                                               SPF00220
```

```
      SUBROUTINE BEFGEN                                                 BEF00010
C                                                                       BEF00020
      COMMON/FLGCOM/ALTFLG,ALHFLG,TRKFLG,IRDFLG,SP3FLG,NMHFLG,IDLFLG,   BEF00030
     *              UNFFLG,MYMRFG,INFLG,ICEVFG,PRTFLG,IFORM             BEF00040
      COMMON/REGCOM/IREJN                                               BEF00050
      COMMON/BEFCOM/BEF                                                 BEF00060
      COMMON/BASECM/BASE3R                                              BEF00070
      COMMON/DELCOM/DEL3R                                               BEF00080
      COMMON/IDLBCM/IDLB3R                                              BEF00090
      COMMON/IDLDCM/IDEL3R                                              BEF00100
      COMMON/IDLCOM/IDLBEF                                              BEF00110
      COMMON/INDXCM/INDX3R                                              BEF00120
      COMMON/MYMCOM/MYM,MYR,TF                                          BEF00130
C                                                                       BEF00140
      REAL BASE3R(10,3,6,3),DEL3R(10,3,6,3)                             BEF00150
      REAL IDLB3R(10,3,6,3),IDEL3R(10,3,6,3)                            BEF00160
      REAL NONMTH(45,6)                                                 BEF00170
      REAL BEF(20,26,3,6),IDLBEF(20,26,3,6)                             BEF00180
      REAL CUMMIL(20,6),MYM(20,6),MYR(20,6),TF(20,6)                    BEF00190
C                                                                       BEF00200
      INTEGER ALHFLG,TRKFLG,ALTFLG,MSFLG,SP3FLG,                        BEF00210
     *        NMHFLG,IDLFLG,UNFFLG,MYMRFG,PRTFLG                        BEF00220
      INTEGER INDX3R(30,3,6,3)                                          BEF00230
      INTEGER MAXAGE(6)                                                 BEF00240
C                                                                       BEF00250
      DATA MAXAGE/19,19,19,19,19,19/                                    BEF00260
      DATA NONMTH/                                                      BEF00270
     *            24*.95 , 21*.95 ,                                     BEF00280
     *            24*.95 , 21*.95 ,                                     BEF00290
     *            24*.95 ,  4*.95 , 17*.85 ,                            BEF00300
     *            24*.95 ,  8*.95 , 13*.85 ,                            BEF00310
     *            45*.95 ,                                              BEF00320
     *            45*.95                      /                         BEF00330
C                                                                       BEF00340
C***********************************************************************BEF00350
C                                                                       BEF00360
      CALL GETCUM(CUMMIL)                                               BEF00370
C                                                                       BEF00380
C*************************************************                      BEF00390
C                                                                       BEF00400
```

```
      DO 100 IMODE=1,6                                                  BEF00410
C                                                                       BEF00420
        MAX = MAXAGE(IMODE)                                             BEF00430
C                                                                       BEF00440
      DO 100 ICY = 70,95                                                BEF00450
      DO 100 IP = 1,3                                                   BEF00460
        IMY1=ICY-19                                                     BEF00470
        IMY13=ICY                                                       BEF00480
        ICYP=ICY-69                                                     BEF00490
C                                                                       BEF00500
      DO 100 MY = IMY1,IMY13                                            BEF00510
        IMYP =  MY - 50                                                 BEF00520
C                                                                       BEF00530
        IAGE=ICY - MY                                                   BEF00540
C                                                                       BEF00550
      IF(IAGE.GT.MAX) IAGE = MAX                                        BEF00560
C                                                                       BEF00570
        AGE = IAGE                                                      BEF00580
        IA2=ICY-MY+1                                                    BEF00590
        IA3=21-IA2                                                      BEF00600
C                                                                       BEF00610
      IF(MY.LT.66) INDX=1                                               BEF00620
      IF(MY.GE.66) INDX=INDXTR(MY-65,IP,IMODE,IREJN)                    BEF00630
C                                                                       BEF00640
      VMTAGE = CUMMIL(IAGE+1,IMODE)*.0001                               BEF00650
C                                                                       BEF00660
      BEF(IA3,ICYP,IP,IMODE) =                                          BEF00670
     *              BASE3R(INDX,IP,IMODE,IREJN) +                       BEF00680
     *                 VMTAGE*DEL3R(INDX,IP,IMODE,IREJN)                BEF00690
C                                                                       BEF00700
      IDLBEF(IA3,ICYP,IP,IMODE) =                                       BEF00710
     *              IDLB3R(INDX,IP,IMODE,IREJN) +                       BEF00720
     *                 VMTAGE*IDEL3R(INDX,IP,IMODE,IREJN)               BEF00730
C                                                                       BEF00740
100   CONTINUE                                                          BEF00750
      DO 450 I=1,20                                                     BEF00760
      DO 450 J=1,26                                                     BEF00770
      DO 450 IM=1,6                                                     BEF00780
      IF(NMHFLG.EQ.1)BEF(I,J,1,IM)=BEF(I,J,1,IM)*NONMTR(I+J-1,IM)       BEF00790
  450 CONTINUE                                                          BEF00800
```

```
C                                                                       BEF00810
C                                                                       BEF00820
C*****************************************************                  BEF00830
C                                                                       BEF00840
C     WRITE(6,444) CUMMIL                                               BEF00850
444   FORMAT(1X,10F8.0)                                                 BEF00860
C                                                                       BEF00870
C     DEBUG SUBCHK                                                      BEF00880
C                                                                       BEF00890
      RETURN                                                            BEF00900
      END                                                               BEF00910
```

```
      SUBROUTINE GETCUM(CUMMIL)                                         GET00010
C                                                                       GET00020
      COMMON/MYMCOM/MYMDUM,MYR,TF                                       GET00030
C                                                                       GET00040
      REAL CUMMIL(20,6),MYM(20,6),MYMDUM(20,6),MYR(20,6),TF(20,6)       GET00050
      REAL FRAC(4,6)                                                    GET00060
      DATA FRAC/                                                        GET00070
     *          .75 , .375 , .25 , .875 ,                               GET00080
     *          .75 , .375 , .25 , .875 ,                               GET00090
     *          .75 , .375 , .25 , .875 ,                               GET00100
     *          .50 , .25  , .50 , .75  ,                               GET00110
     *          .50 , .25  , .50 , .75  ,                               GET00120
     *          .50 , .25  , .50 , .75     /                            GET00130
C                                                                       GET00140
      DO 10 I1=1,6                                                      GET00150
      DO 10 I2=1,20                                                     GET00160
10    MYM(I2,I1) = MYMDUM(I2,I1)*100000.                                GET00170
C                                                                       GET00180
      DO 100 IM=1,6                                                     GET00190
         CUMMIL(1,IM)=MYM(1,IM)*FRAC(2,IM)                              GET00200
      DO 90 I=2,20                                                      GET00210
         SUM=0.                                                         GET00220
         SUM2=0.                                                        GET00230
         II=I-1                                                         GET00240
      DO 50 J=1,II                                                      GET00250
80    SUM=SUM+MYM(J,IM)                                                 GET00260
         III=II-1                                                       GET00270
         IF(III.EQ.0) GO TO 74                                          GET00280
      DO 70 L=1,III                                                     GET00290
70    SUM2=SUM2+MYM(L,IM)                                               GET00300
C                                                                       GET00310
74    CONTINUE                                                          GET00320
         T1=SUM+MYM(I,IM)*FRAC(2,IM)                                    GET00330
         T2=SUM2+MYM(II,IM)*FRAC(4,IM)                                  GET00340
         T1=T1*FRAC(1,IM)                                               GET00350
         T2=T2*FRAC(3,IM)                                               GET00360
         TSUM=T1+T2                                                     GET00370
90    CUMMIL(I,IM)=TSUM                                                 GET00380
100   CONTINUE                                                          GET00390
C                                                                       GET00400
```

```
C       WRITE(6,444) CUMMIL                                     GET00410
444     FORMAT(1X,10F2.0)                                       GET00420
C                                                               GET00430
        RETURN                                                  GET00440
C                                                               GET00450
C       DEBUG SUBCHK                                            GET00460
        END                                                     GET00470
```

```
      SUBROUTINE EFALTX( NMYALT , ALTKOD )                              EFA00010
C                                                                       EFA00020
      COMMON/BEFCOM/BEF                                                 EFA00030
C                                                                       EFA00040
      INTEGER ALTKOD(4,20),NMYALT,CYI,P,ISTD1(3),MYI,ICY                EFA00050
C                                                                       EFA00060
      REAL BEF(20,26,3,6),EFREP(20,4,3)                                 EFA00070
C                                                                       EFA00080
      DATA EFREP/1.0,1.2,1.4,1.6,1.8,2.0,2.2,2.4,2.6,2.8,3.0,3.0,3.0,   EFA00090
     *  0.6,0.7,0.8,1.0,1.1,1.2,1.3,1.4,1.6,1.7, 3*1.8,                 EFA00100
     *  0.27,0.32,0.38,0.43,0.49,0.54,0.59,0.65,0.70,0.76,3*0.81,13*0., EFA00110
     *  23.7,26.8,30.0,33.1,36.2,39.4,42.5,45.7,48.8,4*51.9,            EFA00120
     *  6.98,10.12,13.26,16.40,19.54,22.68,25.82,28.96,32.10,4*35.24,   EFA00130
     *  2.2,4.16,6.12,8.08,10.04,12.,13.96,15.92,17.88,4*19.84,         EFA00140
     *  13*0.0,                                                         EFA00150
     *  2.00,2.06,2.12,2.18,2.24,2.30,2.36,2.42,2.48,2.54,3*2.60,       EFA00160
     *  1.50,1.56,1.62,1.68,1.74,1.80,1.86,1.92,1.98,2.04,3*2.10,       EFA00170
     *  1.00,1.04,1.08,1.12,1.16,1.20,1.24,1.28,1.32,1.36,3*1.40,       EFA00180
     *  0.24,0.29,0.34,0.40,0.56,0.73,0.90,1.1,1.3,1.5,3*1.7,84*0. /    EFA00190
C                                                                       EFA00200
      DO 200 I = 1,NMYALT                                               EFA00210
C                                                                       EFA00220
        MYI1      = ALTKOD(1,I) - 69                                    EFA00230
        ISTD1(1) = ALTKOD(2,I)                                          EFA00240
        ISTD1(2) = ALTKOD(3,I)                                          EFA00250
        ISTD1(3) = ALTKOD(4,I)                                          EFA00260
C                                                                       EFA00270
         DO 110 P = 1,3                                                 EFA00280
              ICODE1 = ISTD1(P)                                         EFA00290
              IF(ICODE1.EQ.0) GO TO 110                                 EFA00300
           DO 100 K = 1,20                                              EFA00310
              IREV = 21-K                                               EFA00320
              IM1 = K-1                                                 EFA00330
              ICY = MYI1 + IM1                                          EFA00340
              IF(ICY.GT.26) GO TO 100                                   EFA00350
              BEF(IREV,ICY,P,1) = EFREP(K,ICODE1,P)                     EFA00360
100        CONTINUE                                                     EFA00370
C                                                                       EFA00380
110      CONTINUE                                                       EFA00390
C                                                                       EFA00400
```

```
100   CONTINUE                                                    EFA00410
C                                                                 EFA00420
      RETURN                                                      EFA00430
C                                                                 EFA00440
C     DEBUG SUBCHK                                                EFA00450
C                                                                 EFA00460
      END                                                         EFA00470
```

```
      SUBROUTINE CCEVAX                                                 CCE00010
C                                                                       CCE00020
      COMMON/CEVCOM/CCEVRT                                              CCE00030
      COMMON/REGCOM/IREJN                                               CCE00040
C                                                                       CCE00050
      INTEGER CY , MY                                                   CCE00060
      REAL CCEV(45,6,3),CCEVRT(45,6),LDGSWD(45),CCCAL(45,4,3)           CCE00070
C     DATA LDGSWD/17*.80,.77,.77,3*.74,2*.72,21*.55/                    CCE00080
C                                                                       CCE00090
      DATA CCEV/12*6.63,5*3.33,3*2.53,7*1.76,2*0.60,16*0.15,            CCE00100
     *          12*6.63,5*3.33,3*2.53,7*1.76,2*0.60,16*0.15,            CCE00110
     *          17*7.70,11*2.53,1*0.60,16*0.15,                         CCE00120
     *          17*7.70,13*2.00,15*0.30,                                CCE00130
     *          45*0.,                                                  CCE00140
     *          27*1.60,18*0.,                                          CCE00150
     *                                                                  CCE00160
     *          10*6.63,3*3.33,7*2.53,7*1.76,2*0.60,16*0.15,            CCE00170
     *          10*6.63,3*3.33,7*2.53,7*1.76,2*0.60,16*0.15,            CCE00180
     *          17*7.70,10*2.53,2*0.60,16*0.15,                         CCE00190
     *          17*7.70,7*2.00,2*1.19,3*0.87,16*0.30,                   CCE00200
     *          45*0.,                                                  CCE00210
     *          27*1.60,18*0.,                                          CCE00220
     *                                                                  CCE00230
     *          12*9.00,5*4.52,3*3.44,6*2.39,1*1.76,2*0.81,4*.2,12*.15, CCE00240
     *          12*9.00,5*4.52,3*3.44,6*2.39,1*1.76,2*0.81,4*.2,12*.15, CCE00250
     *          17*10.46,9*2.72,2.00,2.72,0.81,4*0.2,12*0.15,           CCE00260
     *          17*10.46,16*2.72,12*0.30,                               CCE00270
     *          45*0.,                                                  CCE00280
     *          27*2.17,18*0.       /                                   CCE00290
C                                                                       CCE00300
C                                                                       CCE00310
      DO 100 IMODE = 1,6                                                CCE00320
      DO 90 MY = 1,45                                                   CCE00330
C                                                                       CCE00340
      CCEVRT(MY,IMODE) = CCEV(MY,IMODE,IREJN)                           CCE00350
C                                                                       CCE00360
90    CONTINUE                                                          CCE00370
100   CONTINUE                                                          CCE00380
C                                                                       CCE00390
C                                                                       CCE00400
```

```
      RETURN                                                            CCE00410
C     DEBUG SUBCHK                                                      CCE00420
      END                                                               CCE00430
```

```
      SUBROUTINE LDVIMX
C
      COMMON/BEFCOM/BEF(20,26,3,6)
      COMMON/IMCOM/ICYIM,ISTRN,IMTFLG,MODYR1,MODYR2
      COMMON/IMCRED/PCIM
C
      INTEGER CY,CYP,CYIMP,CYIP1,DELY,DEL2,DEL
      INTEGER PCIM(19,5,4,2,2)
C**IN PCIM: DEL,ISTRIN,ITECH,IPROGM,IPOL
C
C*****************************************************************
C
      IPROGM = 1 + IMTFLG
C     ISTRIN = .1*ISTRN
      IF(ISTRN.LT.10.OR.ISTRN.GT.50) STOP
      IREM = ISTRN - (ISTRN/10)*10
      REM=IREM*.1
C     ISTRIN = ISTRN*.1
      STRN = (ISTRN + 0.5)*0.1
      ISTRIN = STRN
      CYIMP = ICYIM
      CYIP1 = ICYIM + 1
C
      DO 100 IPOL=1,2
      DO 100 CY = CYIP1,95
C
           CYP = CY-69
           DELY = CY - CYIMP
C
      DO 90 I = 1,20
         MY = CY - (20-I)
         MYP = MY - 50
         DEL2 = MY - CYIMP
C
      IF(MY.LT.MODYR1.OR.MY.GT.MODYR2) GO TO 90
C
      IF(MY.LE.CYIMP) DEL = DELY
      IF(MY.GT.CYIMP) DEL = DELY - DEL2
      IF(DEL.GT.19) DEL = 19
C
```

```
C*TEST FOR TECH. I, II, III, OR IV                                      LDV00410
      IF(MY.LT.75) ITECH = 1                                            LDV00420
      IF(MY.GE.75.AND.MY.LE.79) ITECH = 2                               LDV00430
      IF(MY.EQ.80) ITECH = 3                                            LDV00440
      IF(MY.GE.81) ITECH = 4                                            LDV00450
C                                                                       LDV00460
      IF(DEL.EQ.0) PCRED = 0.0                                          LDV00470
      IF(DEL.GT.0.AND.ISTRIN.LT.5)                                      LDV00480
     * PCRED = PCIM(DEL,ISTRIN,ITECH,IPROGM,IPOL)*.01                   LDV00490
     *   + REM*(PCIM(DEL,ISTRIN+1,ITECH,IPROGM,IPOL) -                  LDV00500
     *          PCIM(DEL,ISTRIN,ITECH,IPROGM,IPOL) )*.01                LDV00510
C                                                                       LDV00520
      IF(DEL.GT.0.AND.ISTRIN.EQ.5)                                      LDV00530
     * PCRED = PCIM(DEL,ISTRIN,ITECH,IPROGM,IPOL)*.01                   LDV00540
C                                                                       LDV00550
C*PERHAPS SKIP FOR LDT?                                                 LDV00560
      PCLEFT = 1. - PCRED                                               LDV00570
C                                                                       LDV00580
      PEF(I,CYP,IPOL,1) = PEF(I,CYP,IPOL,1)*PCLEFT                      LDV00590
C                                                                       LDV00600
90    CONTINUE                                                          LDV00610
100   CONTINUE                                                          LDV00620
C                                                                       LDV00630
      RETURN                                                            LDV00640
C                                                                       LDV00650
C     DEBUG SUBCHK                                                      LDV00660
C                                                                       LDV00670
      END                                                               LDV00680
```

```
      SUBROUTINE ALUH(CY,AC,YLOAD,TRAILR,PCCO,PCCC,ABSHUM)
C
      COMMON/RET1/ALHRET(20,4,3)
C
      INTEGER CY
      REAL COMPEF(20,3,6)
      REAL XLOAD(3)
      REAL PCW(20),CFNIX(20,3,6),COMCCC(6),CFLET(20,3,6),TF(20,6)
      REAL A(20,3,3),U(20,3,3),L(3,3),H(3)
      REAL TCFA(3),TCFB(3),TCFC(3)
      REAL CFLD(3)
      REAL ACCF(3),PCWAC(45)
C
      DATA ACCF/1.13,1.18,1.18/
      DATA CFLD/1.06,1.20,1.03/
      DATA PCWAC/15*.54,3*.66,4*.75,2*.81,21*.81/
      DATA TCFA/1.32,2.15,1.16/,TCFB/.75,1.55,1.28/,TCFC/.43,.39,.92/
C  IN TCFA ABOVE, 2ND VALUE SHOULD BE 2.15..IT IS 1.15 FOR ORE. CF.
C
      PCCOLD=PCCC*.01
C
      DO 207 IP=1,3
      H(IP)=1.
      DO 207 IM=1,3
      L(IP,IM)=1.
      DO 207 IT=1,20
      A(IT,IP,IM)=1.
      U(IT,IP,IM)=1.
      ALHRET(IT,4,IP) = 1.
207   CONTINUE
C
      DO 500 IP = 1,3
C
      DO 400 IMODE=1,3
C
      IF(IMODE.EQ.3) GO TO 15
C*****A:  AIR CONDITIONING CORRECTION FACTOR
      DO 90 I=1,20
      IMY = CY - (20-I) - 50
      PCW(I)=PCWAC(IMY)
```

```
      A(I,IMODE,IP) = AC*PCKAC(IMY)*(ACCF(IP)-1.) + 1.0                 ALU00410
20    CONTINUE                                                          ALU00420
C                                                                       ALU00430
      IF(IMODE.EQ.2) GO TO 15                                           ALU00440
C*****U:  TRAILER TOWING CORRECTION FACTOR                              ALU00450
      DO 91 I=1,20                                                      ALU00460
      IMY=CY-(20-I)-50                                                  ALU00470
      IF(IMY.GT.24) GO TO 77                                            ALU00480
      CFA=TCFA(IP)                                                      ALU00490
      GO TO 78                                                          ALU00500
77    CONTINUE                                                          ALU00510
      CFA=(PCCOLD*TCFA(IP) + (1.-PCCOLD)*TCFB(IP))/                     ALU00520
     *            (PCCOLD + (1.-PCCOLD)*TCFC(IP))                       ALU00530
78    CONTINUE                                                          ALU00540
C                                                                       ALU00550
      U(I,IMODE,IP) = TRAILR*(CFA-1.) + 1.0                             ALU00560
C                                                                       ALU00570
91    CONTINUE                                                          ALU00580
C                                                                       ALU00590
C                                                                       ALU00600
15    CONTINUE                                                          ALU00610
C*****L:  ADDITIONAL LOADING CORRECTION FACTOR                           ALU00620
      L(IMODE,IP) = XLOAD(IMODE)*(CFLD(IP)-1.) + 1.                     ALU00630
C                                                                       ALU00640
400   CONTINUE                                                          ALU00650
500   CONTINUE                                                          ALU00660
C*****H:  HUMIDITY CORRECTION FACTOR                                     ALU00670
      DO 675 IMODE=1,3                                                  ALU00680
      H(IMODE) = 1. - .0047*(APSHUM - 75.)                              ALU00690
675   CONTINUE                                                          ALU00700
C                                                                       ALU00710
C                                                                       ALU00720
      DO 800 IP=1,3                                                     ALU00730
      DO 800 IMODE=1,3                                                  ALU00740
      DO 800 IY=1,20                                                    ALU00750
      ALHTMP = A(IY,IMODE,IP)*U(IY,IMODE,IP)*L(IMODE,IP)                ALU00760
      IF(IP.EQ.3) ALHTMP = ALHTMP*H(IMODE)                              ALU00770
      ALHRET(IY,IMODE,IP) = ALHTMP                                      ALU00780
800   CONTINUE                                                          ALU00790
C                                                                       ALU00800
```

```
      RETURN                                                    ALU00810
C     DEBUG SUBCHK                                              ALU00820
      END                                                       ALU00830
```

```
      SUBROUTINE TRKOPC(CY,HGWGT,HDWGT,HGCID,HDCID)                     TRK00010
C                                                                       TRK00020
      COMMON/RET2/TRKRET(20,3,2)                                        TRK00030
      INTEGER CY,INDXAR(45)                                             TRK00040
C**IN B0,B1,AND B2, 1ST INDEX IS YEAR GROUP, 2ND IS VEHTYPE,3RD IS POL  TRK00050
      REAL B0(3,2,3),B1(3,2,3),B2(3,2,3)                                TRK00060
C                                                                       TRK00070
      DATA B0/1.302,-.584,.762,2.058,2.058,.893,                        TRK00080
     *        .814,.354,.320,-.533,-.533,-.299,                         TRK00090
     *        .869,.887,.943,.085,.085,.138      /                      TRK00100
C                                                                       TRK00110
      DATA B1/.177,.124,.131,-.005,-.005,.015,                          TRK00120
     *        -.036,.106,.149,-.03,-.03,.03,                            TRK00130
     *        .172,.016,.008,.02,.02,.023        /                      TRK00140
C                                                                       TRK00150
      DATA B2/-.065,-.01,-.047,-.014,-.014,-.006,                       TRK00160
     *        .016,-.024,-.045,.043,.043,.003,                          TRK00170
     *        -.054,-.003,-.012,.002,.002,.001   /                      TRK00180
C                                                                       TRK00190
      DATA INDXAR/19*1,4*2,22*3/                                        TRK00200
C                                                                       TRK00210
      DO 100 IMY=1,20                                                   TRK00220
C                                                                       TRK00230
         MY = CY - (20-IMY) - 50                                        TRK00240
         IX=INDXAR(MY)                                                  TRK00250
C                                                                       TRK00260
      DO 100 IPOL = 1,3                                                 TRK00270
C                                                                       TRK00280
      TRKRET(IMY,IPOL,1) = B0(IX,1,IPOL) + B1(IX,1,IPOL)*HGWGT*.001 +   TRK00290
     *                     B2(IX,1,IPOL)*HGWGT/HGCID                    TRK00300
C                                                                       TRK00310
      TRKRET(IMY,IPOL,2) = B0(IX,2,IPOL) + B1(IX,2,IPOL)*HDWGT*.001 +   TRK00320
     *                     B2(IX,2,IPOL)*HDWGT/HDCID                    TRK00330
C                                                                       TRK00340
100   CONTINUE                                                          TRK00350
C                                                                       TRK00360
      RETURN                                                            TRK00370
C     DEBUG SUBCHK                                                      TRK00380
      END                                                               TRK00390
```

```
 BLOCK DATA                                                          BL100010
 COMMON /LINK/ NLINK,NLANE(200),X1(200),Y1(200),X2(200),             BL100020
1Y2(200),LCAP(200),DIST(200),VEL(200),LCON(200,3),                   BL100030
2NSTOPS(200)                                                         BL100040
 INTEGER NSTOPS/200*0/                                               BL100050
 COMMON /INTRST/ NINS,LIN(70,4),ITYPC(70),ICYCL(70),                 BL100060
1PH(70,4),CI(70),GCAP(70,4),QUE(70,4)                                BL100070
 REAL QUE/280*0.0/                                                   BL100080
 COMMON /PARKZ/ PZ,PV(50),PLA(50),VZ(50),ZV(50),PD,PLS,              BL100090
1PLBO,PNOS(50),PLL(50)                                               BL100100
 REAL PV/50*0.0/,PLA/50*0.0/,PLS/15.0/,PLBO/12.0/                    BL100110
 COMMON /VEHTYP/ VT,VTP(3),VTE(3),VTMULT(2),S7VTP(3,2)               BL100120
 REAL VTP/1.4,14.0,1.0/                                              BL100130
 END                                                                 BL100140
```

```
      BLOCK DATA                                                        BL200010
      COMMON/COEFMM/BAD1,BAD2,BAD3,BAD4                                 BL200020
      COMMON/DET/DETER                                                  BL200030
      REAL DETER(20),BAD1(12,5),BAD2(12,5),BAD3(12,5),BAD4(12,5)        BL200040
      DATA BAD1/                                                        BL200050
     &  0.10210419E+01,-0.46106476E-01, 0.12750413E-01, 0.19804664E-01, BL200060
     &  0.10487151E-02,-0.15174127E+00, 0.19398060E-01,-0.79918522E-04, BL200070
     & -0.27256622E-03, 0.33066291E+00,-0.24132896E-02, 0.32060244E-03, BL200080
     &  0.63576120E+00,-0.24510358E-01, 0.43812852E-01, 0.49628541E-02, BL200090
     &  0.44956130E-03,-0.47977134E-01, 0.61538182E-02, 0.10994070E-04, BL200100
     & -0.67511966E-04, 0.30338836E+00,-0.27355787E-02, 0.13473761E-03, BL200110
     &  0.26043031E+00,-0.47583580E-02, 0.43524541E-01, 0.13808063E-02, BL200120
     &  0.10754808E-03,-0.21531656E-02, 0.93090301E-03, 0.11250070E-04, BL200130
     & -0.19257031E-05, 0.22921783E+00,-0.40214434E-02, 0.12248126E-03, BL200140
     &  0.49613750E+00,-0.14627926E-01, 0.61504323E-01,-0.10729437E-02, BL200150
     &  0.21254545E-03,-0.68023652E-02, 0.97647379E-03, 0.10525828E-03, BL200160
     &  0.37405029E-04, 0.26291054E+00,-0.18189936E-02, 0.84008745E-04, BL200170
     &  0.50066710E+00,-0.17586585E-01, 0.59339337E-01,-0.18133272E-02, BL200180
     &  0.22920808E-03,-0.12368988E-01, 0.11492474E-02, 0.74215845E-04, BL200190
     &  0.14108437E-04, 0.30531263E+00,-0.70244968E-02, 0.10785337E-03/ BL200200
      DATA BAD2/                                                        BL200210
     &  0.39664203E+00,-0.13213880E-01, 0.58552373E-01,-0.19716325E-02, BL200220
     &  0.16135792E-03,-0.43251887E-02, 0.21358313E-03, 0.86380664E-04, BL200230
     &  0.40023762E-04, 0.24617422E+00,-0.68550445E-02, 0.10254829E-03, BL200240
     &  0.36676234E+00,-0.14121491E-01, 0.68226576E-01,-0.32857144E-02, BL200250
     &  0.18291990E-03, 0.22933292E-02, 0.19526284E-04, 0.13627447E-03, BL200260
     &  0.54617223E-04, 0.22517568E+00,-0.46133995E-02, 0.69841262E-04, BL200270
     &  0.10577946E+01,-0.67509890E-01, 0.62050894E-01, 0.11177700E-01, BL200280
     &  0.14054242E-02,-0.17354453E+00, 0.21822363E-01, 0.12219173E-03, BL200290
     & -0.29136962E-03, 0.30515999E+00,-0.65815225E-02, 0.36294828E-03, BL200300
     &  0.68093568E+00,-0.48967101E-01, 0.90703070E-01, 0.41544996E-02, BL200310
     &  0.97224675E-03,-0.92593908E-01, 0.13378501E-01, 0.19100340E-03, BL200320
     & -0.13010977E-03, 0.19975567E+00,-0.77630468E-02, 0.33748685E-03, BL200330
     &  0.10887880E+01,-0.71653783E-01, 0.56213565E-01, 0.12077432E-01, BL200340
     &  0.14173663E-02,-0.18283314E+00, 0.22780791E-01, 0.11450195E-03, BL200350
     & -0.30368124E-03, 0.31737489E+00,-0.97049810E-02, 0.31322357E-03/ BL200360
      DATA BAD3/                                                        BL200370
     &  0.12207556E+01,-0.89525878E-01, 0.95823226E-01, 0.61595142E-02, BL200380
     &  0.17752303E-02,-0.19405907E+00, 0.24501167E-01, 0.30660117E-03, BL200390
     & -0.31120144E-03, 0.30530828E+00,-0.67905560E-02, 0.22574072E-03, BL200400
```

```
&  0.14931040E+01,-0.10349673E+00, 0.72227307E-02, 0.59160888E-02, BL200410
&  0.20053601E-02,-0.23721021E+00, 0.26225589E-01, 0.23920002E-04, BL200420
& -0.39849989E-03, 0.32876998E+00,-0.91728717E-02, 0.29819994E-03, BL200430
&  0.11040087E+01,-0.79453886E-01, 0.56303188E-01, 0.17077201E-02, BL200440
&  0.16181599E-02,-0.17064410E+00, 0.19803409E-01, 0.20100000E-03, BL200450
& -0.26381994E-03, 0.25955880E+00,-0.75969696E-02, 0.26727002E-03, BL200460
&  0.28853571E+00,-0.12315691E-01, 0.92407882E-01,-0.31504999E-02, BL200470
&  0.16614000E-03, 0.70702210E-02, 0.46436000E-03, 0.14813000E-03, BL200480
&  0.56589997E-04, 0.15976292E+00,-0.65233000E-02, 0.10359001E-03, BL200490
&  0.20974290E+00,-0.12848910E-01, 0.96859634E-01,-0.66359304E-02, BL200500
&  0.18598999E-03, 0.21924630E-01,-0.15685901E-02, 0.27883006E-03, BL200510
&  0.12293000E-03, 0.11387944E+00,-0.28685799E-02, 0.47189998E-04/ BL200520
 DATA BAD4/                                                        BL200530
&  0.57665521E+00,-0.36277410E-01, 0.21571960E-01, 0.29442401E-02, BL200540
&  0.63650007E-03,-0.52853432E-01, 0.69414489E-02,-0.17560000E-04, BL200550
& -0.94470000E-04, 0.25530094E+00,-0.76948516E-02, 0.13530999E-03, BL200560
&  0.54005557E+00,-0.39594091E-01, 0.28321180E-01, 0.14426301E-02, BL200570
&  0.70013991E-03,-0.57096951E-01, 0.73297806E-02, 0.53550000E-04, BL200580
& -0.85880005E-04, 0.20601881E+00,-0.58017895E-02, 0.85580003E-04, BL200590
&  0.21578521E+00,-0.12577798E-01, 0.51477298E-01,-0.23425999E-02, BL200600
&  0.16780000E-03,-0.15755999E-02, 0.28229994E-03, 0.12529999E-03, BL200610
&  0.48500006E-04, 0.11655778E+00,-0.46298988E-02, 0.69899994E-04, BL200620
&  0.21578521E+00,-0.12577798E-01, 0.51477298E-01,-0.23425999E-02, BL200630
&  0.16780000E-03,-0.15755999E-02, 0.28229994E-03, 0.12529999E-03, BL200640
&  0.48500006E-04, 0.11655778E+00,-0.46298988E-02, 0.69899994E-04, BL200650
&  0.21578521E+00,-0.12577798E-01, 0.51477298E-01,-0.23425999E-02, BL200660
&  0.16780000E-03,-0.15755999E-02, 0.28229994E-03, 0.12529999E-03, BL200670
& 0.48500006E-04, 0.11655778E+00,-0.46298988E-02, 0.69899994E-04/  BL200680
 DATA DETER/.146,.214,.558,.671,.708,.824,1.033,.312,.290,.327,     BL200690
& .351,.218,.217,.497,.524,.558,.584,.339,.144,-0.037/               BL200700
 END                                                                BL200710
```

```
      BLOCK DATA                                                        BL300010
C                                                                       BL300020
      COMMON/ACOM/AIN1,AIN2,AIN3                                        BL300030
C                                                                       BL300040
      REAL AIN1(6,18),AIN2(6,18),AIN3(6,18)                             BL300050
C                                                                       BL300060
C**HC SPEED CORRECTION FACTOR COEFFICIENTS                              BL300070
      DATA AIN1/                                                        BL300080
     *2.2461,-.29097,0.015889,-.47249E-03,0.69408E-05,-.39280E-07,      BL300090
     *2.3103,-.28957,0.015299,-.44669E-03,0.64818E-05,-.36346E-07,      BL300100
     *2.1656,-.26999,0.014420,-.43364E-03,0.65074E-05,-.37810E-07,      BL300110
     *2.3973,-.29998,0.016135,-.48749E-03,0.72909E-05,-.41977E-07,      BL300120
     *2.4087,-.30819,0.016817,-.50684E-03,0.75385E-05,-.43160E-07,      BL300130
     *2.2322,-.28499,0.015383,-.45674E-03,0.67349E-05,-.38380E-07,      BL300140
     *2.2522,-.28778,0.015682,-.47318E-03,0.70795E-05,-.40846E-07,      BL300150
     *2.0278,-.27305,0.015360,-.46030E-03,0.67853E-05,-.38488E-07,      BL300160
     *2.1506,-.28362,0.015380,-.44214E-03,0.62873E-05,-.34631E-07,      BL300170
     *2.2302,-.29365,0.016240,-.48415E-03,0.71159E-05,-.40286E-07,      BL300180
     *2.1223,-.29107,0.016910,-.52615E-03,0.80271E-05,-.47012E-07,      BL300190
     *2.1536,-.28345,0.015700,-.46976E-03,0.69383E-05,-.39471E-07,      BL300200
     *2.0735,-.28935,0.017300,-.55471E-03,0.86420E-05,-.51311E-07,      BL300210
     *2.3495,-.30496,0.016842,-.50962E-03,0.75952E-05,-.43496E-07,      BL300220
     *2.1134,-.28568,0.016320,-.50079E-03,0.75507E-05,-.43719E-07,      BL300230
     *2.1194,-.29863,0.018450,-.61654E-03,0.99206E-05,-.60402E-07,      BL300240
     *2.6638,-.34463,0.019542,-.62572E-03,0.97844E-05,-.58337E-07,      BL300250
     *2.3954,-.33578,0.021161,-.73155E-03,0.12072E-04,-.74857E-07/      BL300260
C                                                                       BL300270
C**CO SPEED CORRECTION FACTOR COEFFICIENTS                              BL300280
      DATA AIN2/                                                        BL300290
     *  1.8198,-.25466,.015235,-.48740E-03,.75821E-05,-.44951E-07,      BL300300
     *  2.3399,-.29698,.016007,-.47740E-03,.70675E-05,-.40398E-07,      BL300310
     *  2.4415,-.29147,.014296,-.38785E-03,.52978E-05,-.28244E-07,      BL300320
     *  2.4655,-.30502,.016050,-.47397E-03,.69908E-05,-.39976E-07,      BL300330
     *  2.7780,-.31913,.015318,-.42233E-03,.58495E-05,-.31497E-07,      BL300340
     *  2.7890,-.32711,.016294,-.46757E-03,.67191E-05,-.37440E-07,      BL300350
     *  2.7074,-.33131,.017618,-.53858E-03,.81740E-05,-.47780E-07,      BL300360
     *  1.8692,-.27668,.017233,-.55828E-03,.87168E-05,-.51698E-07,      BL300370
     *  1.8213,-.27205,.017030,-.55202E-03,.86254E-05,-.51144E-07,      BL300380
     *  2.0142,-.29519,.018635,-.62161E-03,.99366E-05,-.59978E-07,      BL300390
     *  2.0453,-.31062,.020485,-.70853E-03,.11621E-04,-.71569E-07,      BL300400
```

```
     *  2.3187,-.34115,.020945,-.66589E-03,.10223E-04,-.59827E-07,      BL300410
     *  2.5752,-.32889,.018975,-.62826E-03,.10092E-04,-.61273E-07,      BL300420
     *  2.6845,-.33282,.017628,-.52412E-03,.77222E-05,-.43702E-07,      BL300430
     *  2.1549,-.32912,.021011,-.68906E-03,.10839E-04,-.64712E-07,      BL300440
     *  2.5456,-.36285,.023277,-.81504E-03,.13623E-04,-.85591E-07,      BL300450
     *  2.8393,-.36876,.021078,-.67644E-03,.10627E-04,-.63641E-07,      BL300460
     *  2.4875,-.39156,.027072,-.97618E-03,.16527E-04,-1.0432E-07/      BL300470
C                                                                       BL300480
C**NOX SPEED CORRECTION FACTOR COEFFICIENTS                             BL300490
      DATA AIN3/                                                        BL300500
     *  2.4442,-.250110,1.3829E-02,-.28703E-03,.20758E-05,0.0,          BL300510
     *  1.6863,-.118300,.65497E-02,-.13714E-03,.10085E-05,0.0,          BL300520
     *  1.1265,-.039340,.26864E-02,-.60802E-04,.47729E-06,0.0,          BL300530
     *  1.2268,-.044499,.26248E-02,-.56715E-04,.43429E-06,0.0,          BL300540
     *  1.0174,-.011896,.91437E-03,-.21574E-04,.18230E-06,0.0,          BL300550
     *  .98760,-.019567,.16964E-02,-.40400E-04,.32800E-06,0.0,          BL300560
     *  1.1592,-.044454,.29643E-02,-.66899E-04,.52236E-06,0.0,          BL300570
     *  1.8866,-.161290,.90499E-02,-.18561E-03,.13256E-05,0.0,          BL300580
     *  1.5578,-.113030,.67183E-02,-.14341E-03,.10608E-05,0.0,          BL300590
     *  2.0452,-.194010,.11074E-01,-.23175E-03,.16837E-05,0.0,          BL300600
     *  1.6326,-.121860,.70302E-02,-.14629E-03,.10614E-05,0.0,          BL300610
     *  1.4482,-.122440,.79502E-02,-.17108E-03,.12578E-05,0.0,          BL300620
     *  .24597,.084195,-.34084E-02,.62988E-04,-.41397E-06,0.0,          BL300630
     *  1.2817,-.080487,.53574E-02,-.11889E-03,.90106E-06,0.0,          BL300640
     *  1.5345,-.125670,.78592E-02,-.16943E-03,.12549E-05,0.0,          BL300650
     *  .70481,.038153,-.17391E-02,.32614E-04,-.20385E-06,0.0,          BL300660
     *  .78384,.32855E-03,.10603E-02,-.31935E-04,.29039E-06,0.0,        BL300670
     *  .94213,-.042324,.38625E-02,-.93985E-04,.75388E-06,0.0   /       BL300680
C                                                                       BL300690
      END                                                               BL300700
```

```
      BLOCK DATA                                                        BL400010
C                                                                       BL400020
      COMMON/MYMCOM/MYM,MYR,TF                                          BL400030
      COMMON/LNKCOM/SPD,TEMP,PCCO,PCHS,PCCC,MS                          BL400040
C                                                                       BL400050
      REAL SPD(3),MS(6),MYM(20,6),MYR(20,6),TF(20,6)                    BL400060
C                                                                       BL400070
C**DEFAULT MODAL SPLIT                                                  BL400080
      DATA MS/.803,.059,.058,.045,.031,.005/                             BL400090
C                                                                       BL400100
C**ANNUAL AVERAGE MILEAGE ACCUMULATION (DIVIDED BY 100000)              BL400110
      DATA MYM/                                                         BL400120
     *0.159,0.150,0.140,0.131,0.122,0.113,0.103,0.094,0.085,0.076,      BL400130
     *0.067,0.066,0.062,0.059,0.055,0.051,0.050,0.047,0.044,0.044,      BL400140
     *0.159,0.150,0.140,0.131,0.122,0.113,0.103,0.094,0.085,0.076,      BL400150
     *0.067,0.066,0.062,0.059,0.055,0.051,0.050,0.047,0.044,0.044,      BL400160
     *0.157,0.157,0.141,0.126,0.113,0.102,0.094,0.086,0.080,0.075,      BL400170
     *0.071,0.066,0.063,0.060,0.055,0.052,0.050,0.047,0.044,0.041,      BL400180
     *0.190,0.190,0.179,0.165,0.150,0.135,0.120,0.106,0.095,0.086,      BL400190
     *0.078,0.070,0.063,0.059,0.053,0.049,0.047,0.046,0.044,0.042,      BL400200
     *0.736,0.736,0.699,0.633,0.566,0.500,0.456,0.412,0.382,0.360,      BL400210
     *0.346,0.338,0.331,0.324,0.309,0.287,0.257,0.213,0.184,0.154,      BL400220
     *.0201,.0251,.0207,.0189,.0172,.0162,.0155,.0149,.0143,.0139,      BL400230
     *.0135,.0132,.0129,.0127,.0125,.0123,.0122,.0121,.0120,.0119/      BL400240
C                                                                       BL400250
C**VEHICLE REGISTRATION DISTRIBUTIONS                                   BL400260
      DATA MYR/                                                         BL400270
     *0.075,0.107,0.107,0.106,0.100,0.092,0.085,0.077,0.066,0.052,      BL400280
     *0.039,0.027,0.018,0.014,0.009,0.006,0.005,0.005,0.005,0.004,      BL400290
     *0.061,0.095,0.094,0.103,0.083,0.076,0.076,0.063,0.054,0.043,      BL400300
     *0.036,0.024,0.030,0.028,0.026,0.024,0.022,0.020,0.018,0.016,      BL400310
     *0.037,0.070,0.078,0.086,0.075,0.075,0.075,0.068,0.059,0.053,      BL400320
     *0.044,0.032,0.038,0.036,0.034,0.032,0.030,0.028,0.026,0.024,      BL400330
     *0.037,0.070,0.078,0.086,0.075,0.075,0.075,0.068,0.059,0.053,      BL400340
     *0.044,0.032,0.038,0.036,0.034,0.032,0.030,0.028,0.026,0.024,      BL400350
     *0.077,0.135,0.134,0.131,0.099,0.090,0.082,0.062,0.045,0.033,      BL400360
     *0.025,0.015,0.013,0.011,0.010,0.008,0.007,0.006,0.005,0.004,      BL400370
     *0.105,0.225,0.206,0.149,0.097,0.062,0.046,0.033,0.029,0.023,      BL400380
     *0.008,0.005,0.013,0.0  ,0.0  ,0.0  ,0.0  ,0.0  ,0.0  ,0.0  /      BL400390
C                                                                       BL400400
```

```
      END                                                               GL400410
```

```
      BLOCK DATA                                                        BL500010
C                                                                       BL500020
      COMMON/INDXCM/INDX49,INDXCA,INDXHI                                BL500030
      COMMON/BASECM/BASE49,BASECA,BASEHI                                BL500040
      COMMON/DELCOM/ DEL49, DELCA, DELHI                                BL500050
C                                                                       BL500060
      INTEGER INDX49(30,3,6),INDXCA(30,3,6),INDXHI(30,3,6)              BL500070
      REAL BASE49(10,3,6),BASECA(10,3,6),BASEHI(10,3,6)                 BL500080
      REAL DEL49 (10,3,6),DELCA (10,3,6),DELHI (10,3,6)                 BL500090
C                                                                       BL500100
C     6 6 6 6 7 7 7 7 7 7 7 7 7 7 8 8 8 8 8 8 8 8 8 8 9 9 9 9 9 9        BL500110
C     6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5        BL500120
      DATA INDX49/                                                      BL500130
     *1,1,2,2,2,2,2,2,2,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,     BL500140
     *1,1,2,2,2,2,2,2,2,3,3,3,3,3,4,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,     BL500150
     *1,1,2,2,2,2,2,3,3,4,4,5,5,5,5,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,     BL500160
     *1,1,2,2,2,2,2,2,2,3,3,3,3,4,4,4,4,5,5,5,5,5,5,5,5,5,5,5,5,5,5,     BL500170
     *1,1,2,2,2,2,2,2,2,3,3,3,3,4,4,4,4,5,5,5,5,5,5,5,5,5,5,5,5,5,5,     BL500180
     *1,1,2,2,2,2,2,3,3,4,4,4,4,5,5,5,5,5,5,6,6,6,6,6,6,6,6,6,6,6,6,     BL500190
     *1,1,1,1,2,2,2,2,2,2,2,2,2,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,     BL500200
     *1,1,1,1,2,2,2,2,2,2,2,2,2,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,     BL500210
     *1,1,1,1,2,2,2,2,2,2,2,2,2,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,     BL500220
     *1,1,1,1,2,2,2,2,3,3,3,3,3,4,4,4,4,5,5,5,5,5,5,5,5,5,5,5,5,5,5,     BL500230
     *1,1,1,1,2,2,2,2,3,3,3,3,3,4,4,4,4,5,5,5,5,5,5,5,5,5,5,5,5,5,5,     BL500240
     *1,1,1,1,2,2,2,2,3,3,3,3,3,4,4,4,4,4,4,5,5,5,5,5,5,5,5,5,5,5,5,     BL500250
     *1,1,1,1,1,1,1,1,2,2,2,2,2,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,     BL500260
     *1,1,1,1,1,1,1,1,2,2,2,2,2,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,     BL500270
     *1,1,1,1,1,1,1,1,2,2,2,2,2,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,     BL500280
     *1,1,1,1,1,1,1,1,1,1,1,1,2,2,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,     BL500290
     *1,1,1,1,1,1,1,1,1,1,1,1,2,2,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,     BL500300
     *1,1,1,1,1,1,1,1,1,1,1,1,2,2,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4 /   BL500310
C                                                                       BL500320
C     6 6 6 6 7 7 7 7 7 7 7 7 7 7 8 8 8 8 8 8 8 8 8 8 9 9 9 9 9 9        BL500330
C     6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5        BL500340
      DATA INDXCA/                                                      BL500350
     *2,2,2,2,2,2,2,2,2,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,     BL500360
     *2,2,2,2,2,2,2,2,2,3,3,4,4,4,4,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,     BL500370
     *2,2,2,2,2,2,2,3,3,4,4,5,5,5,6,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,     BL500380
     *1,1,2,2,2,2,2,2,2,3,3,3,4,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,     BL500390
     *1,1,2,2,2,2,2,2,2,3,3,3,4,5,5,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,     BL500400
```

```
     *1,1,2,2,2,2,2,3,3,4,4,4,4,5,5,6,6,7,7,7,7,7,7,7,7,7,7,7,7,7,      BL500410
     *1,1,1,2,2,2,2,2,2,2,2,2,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,      BL500420
     *1,1,1,2,2,2,2,2,2,2,2,2,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,      BL500430
     *1,1,1,2,2,2,2,2,2,2,2,2,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,      BL500440
     *1,1,1,2,2,2,2,3,3,4,4,5,5,5,6,6,6,7,7,7,7,7,7,7,7,7,7,7,7,7,      BL500450
     *1,1,1,2,2,2,2,3,3,4,4,5,5,5,5,5,5,6,6,6,6,6,6,6,6,6,6,6,6,6,      BL500460
     *1,1,1,2,2,2,2,3,3,4,4,5,5,5,6,6,6,6,6,7,7,7,7,7,7,7,7,7,7,7,      BL500470
     *1,1,1,1,1,1,1,2,2,2,2,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,      BL500480
     *1,1,1,1,1,1,1,2,2,2,2,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,      BL500490
     *1,1,1,1,1,1,1,2,2,2,2,3,3,3,4,4,4,4,4,5,5,5,5,5,5,5,5,5,5,5,      BL500500
     *1,1,1,1,1,1,1,1,1,1,1,1,2,2,3,3,4,5,5,5,5,5,5,5,5,5,5,5,5,5,      BL500510
     *1,1,1,1,1,1,1,1,1,1,1,1,2,2,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,      BL500520
     *1,1,1,1,1,1,1,1,1,1,1,1,2,2,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4 /    BL500530
C                                                                      BL500540
C     6 6 6 6 7 7 7 7 7 7 7 7 7 7 7 8 8 8 8 8 8 8 8 8 9 9 9 9 9 9       BL500550
C     6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5       BL500560
      DATA INDXHI/                                                     BL500570
     *1,1,2,2,2,2,2,2,2,3,3,4,5,5,6,7,7,7,8,8,8,8,8,8,8,8,8,8,8,8,      BL500580
     *1,1,2,2,2,2,2,2,2,3,3,4,5,5,6,7,7,7,8,8,8,8,8,8,8,8,8,8,8,8,      BL500590
     *1,1,2,2,2,2,2,3,3,4,4,5,6,6,7,8,8,8,9,9,9,9,9,9,9,9,9,9,9,9,      BL500600
     *1,1,2,2,2,2,2,2,2,3,3,4,5,6,6,6,6,7,8,8,8,8,8,8,8,8,8,8,8,8,      BL500610
     *1,1,2,2,2,2,2,2,2,3,3,4,5,6,6,6,6,7,8,8,8,8,8,8,8,8,8,8,8,8,      BL500620
     *1,1,2,2,2,2,2,3,3,4,4,5,6,7,7,7,7,7,8,9,9,9,9,9,9,9,9,9,9,9,      BL500630
     *1,1,1,1,2,2,2,2,2,2,2,2,2,3,3,3,3,4,5,5,5,5,5,5,5,5,5,5,5,5,      BL500640
     *1,1,1,1,2,2,2,2,2,2,2,2,2,3,3,3,3,4,5,5,5,5,5,5,5,5,5,5,5,5,      BL500650
     *1,1,1,1,2,2,2,2,2,2,2,2,2,3,3,3,3,3,4,5,5,5,5,5,5,5,5,5,5,5,      BL500660
     *1,1,1,1,2,2,2,2,3,3,3,3,3,4,4,4,4,5,6,6,6,6,6,6,6,6,6,6,6,6,      BL500670
     *1,1,1,1,2,2,2,2,3,3,3,3,3,4,4,4,4,5,6,6,6,6,6,6,6,6,6,6,6,6,      BL500680
     *1,1,1,1,2,2,2,2,3,3,3,3,3,4,4,4,4,4,5,6,6,6,6,6,6,6,6,6,6,6,      BL500690
     *1,1,1,1,1,1,1,1,2,2,2,2,2,3,3,3,3,4,5,5,5,5,5,5,5,5,5,5,5,5,      BL500700
     *1,1,1,1,1,1,1,1,2,2,2,2,2,3,3,3,3,4,5,5,5,5,5,5,5,5,5,5,5,5,      BL500710
     *1,1,1,1,1,1,1,1,2,2,2,2,2,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,      BL500720
     *1,1,1,1,1,1,1,1,1,1,1,1,2,2,3,3,3,4,5,5,5,5,5,5,5,5,5,5,5,5,      BL500730
     *1,1,1,1,1,1,1,1,1,1,1,1,2,2,3,3,3,4,5,5,5,5,5,5,5,5,5,5,5,5,      BL500740
     *1,1,1,1,1,1,1,1,1,1,1,1,2,2,3,3,3,4,5,5,5,5,5,5,5,5,5,5,5,5 /    BL500750
C                                                                      BL500760
C***EXHAUST EMISSION FACTOR INTERCEPT: 49 STATE                        BL500770
      DATA BASE49/                                                     BL500780
     *  4.45,  2.43,  1.13,  0.13,  0.13,  0.13, 0.0 , 0.0 , 0.0 , 0.0 ,BL500790
     * 68.30, 31.14, 18.60,  3.00,  1.40,  1.40, 0.0 , 0.0 , 0.0 , 0.0 ,BL500800
```

```
     * 3.58,  4.43,  2.98,  2.42,  1.50,  0.29, 0.0 , 0.0 , 0.0 , 0.0 ,BL500810
     * 4.45,  2.43,  1.11,  0.94,  0.31,  0.31, 0.0 , 0.0 , 0.0 , 0.0 ,BL500820
     * 68.30, 31.14, 16.10, 14.50,  3.87,  3.87, 0.0 , 0.0 , 0.0 , 0.0 ,BL500830
     * 3.58,  4.43,  2.98,  2.45,  1.73,  0.41, 0.0 , 0.0 , 0.0 , 0.0 ,BL500840
     * 5.99,  2.90,  0.94,  0.31,  0.31,  0.31, 0.0 , 0.0 , 0.0 , 0.0 ,BL500850
     * 78.70, 32.40, 14.50,  3.87,  3.87,  3.87, 0.0 , 0.0 , 0.0 , 0.0 ,BL500860
     * 6.49,  5.04,  1.73,  0.41,  0.41,  0.41, 0.0 , 0.0 , 0.0 , 0.0 ,BL500870
     * 23.90, 18.54, 22.02,  5.22,  1.46,  1.46, 0.0 , 0.0 , 0.0 , 0.0 ,BL500880
     *272.90,212.70,218.80,191.90, 15.38, 15.38, 0.0 , 0.0 , 0.0 , 0.0 ,BL500890
     * 8.80, 12.80, 10.50,  9.10,  3.99,  3.99, 0.0 , 0.0 , 0.0 , 0.0 ,BL500900
     * 4.30,  4.50,  4.50,  2.85,  2.85,  2.85, 0.0 , 0.0 , 0.0 , 0.0 ,BL500910
     * 35.10, 27.00, 27.00, 27.00, 27.00, 27.00, 0.0 , 0.0 , 0.0 , 0.0 ,BL500920
     * 21.40, 20.10, 19.90,  5.35,  5.35,  5.35, 0.0 , 0.0 , 0.0 , 0.0 ,BL500930
     * 8.96,  4.70,  3.82,  0.29,  0.0 ,  0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL500940
     * 34.40, 20.27, 14.86,  2.71,  0.0 ,  0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL500950
     * 0.14,  0.28,  0.56,  0.04,  0.0 ,  0.0 , 0.0 , 0.0 , 0.0 , 0.0 /BL500960
C                                                                       BL500970
C***EXHAUST EMISSION FACTOR INTERCEPT: CALIFORNIA                       BL500980
      DATA BASECA/                                                      BL500990
     * 4.45,  2.00,  0.29,  0.15,  0.15,  0.15, 0.0 , 0.0 , 0.0 , 0.0 ,BL501000
     * 68.30, 30.42, 3.80,  3.80,  3.00,  3.00, 0.0 , 0.0 , 0.0 , 0.0 ,BL501010
     * 3.58,  3.92,  3.12,  1.50,  1.10,  0.29, 0.21, 0.0 , 0.0 , 0.0 ,BL501020
     * 4.45,  2.43,  1.11,  0.29,  0.15,  0.15, 0.15, 0.0 , 0.0 , 0.0 ,BL501030
     * 68.30, 31.14, 16.10, 13.70,  3.80,  3.80, 3.80, 0.0 , 0.0 , 0.0 ,BL501040
     * 3.58,  4.43,  2.98,  1.50,  1.28,  0.54, 0.21, 0.0 , 0.0 , 0.0 ,BL501050
     * 5.99,  2.90,  0.29,  0.16,  0.16,  0.16, 0.16, 0.0 , 0.0 , 0.0 ,BL501060
     * 78.70, 32.40, 13.70,  3.80,  3.80,  3.80, 3.80, 0.0 , 0.0 , 0.0 ,BL501070
     * 6.49,  5.04,  1.73,  0.44,  0.44,  0.44, 0.44, 0.0 , 0.0 , 0.0 ,BL501080
     * 23.90, 18.54, 22.02, 18.16,  5.22,  3.50, 1.46, 0.0 , 0.0 , 0.0 ,BL501090
     *272.90,212.70,218.80,209.00,191.90, 15.38,15.38, 0.0 , 0.0 , 0.0 ,BL501100
     * 8.80, 12.80, 10.50,  7.50,  8.04,  6.20, 3.99, 0.0 , 0.0 , 0.0 ,BL501110
     * 4.30,  4.50,  4.50,  2.85,  2.85,  2.85, 2.85, 0.0 , 0.0 , 0.0 ,BL501120
     * 35.10, 27.00, 27.00, 27.00, 27.00, 27.00,27.00, 0.0 , 0.0 , 0.0 ,BL501130
     * 21.40, 20.10, 18.61, 16.30,  5.35,  5.35, 5.35, 0.0 , 0.0 , 0.0 ,BL501140
     * 8.96,  4.70,  3.82,  0.82,  0.29,  0.29, 0.29, 0.29, 0.29, 0.29,BL501150
     * 34.40, 20.27, 14.86,  2.71,  2.71,  2.71, 2.71, 2.71, 2.71, 2.71,BL501160
     * 0.14,  0.28,  0.56,  0.04,  0.04,  0.04, 0.04, 0.04, 0.04, 0.04/BL501170
C                                                                       BL501180
C***EXHAUST EMISSION FACTOR INTERCEPT: HIGH ALTITUDE                    BL501190
      DATA BASEHI/                                                      BL501200
```

```
     *  6.03,  4.07,  1.83,  0.70,  1.83,  0.21, 0.21, 0.13, 0.13, 0.13,BL501210
     *110.04, 76.73, 38.31, 10.30, 38.31,  6.18, 2.88, 1.40, 1.40, 1.40,BL501220
     *  1.96,  3.00,  1.94,  1.50,  1.90,  1.50, 0.93, 0.18, 0.29, 0.29,BL501230
     *  6.03,  4.07,  1.79,  1.11,  1.79,  1.52, 0.47, 0.31, 0.0 , 0.0 ,BL501240
     *110.04, 76.73, 33.15, 16.10, 33.15, 29.90, 8.00, 3.87, 0.0 , 0.0 ,BL501250
     *  1.96,  3.00,  1.94,  1.52,  2.45,  1.52, 1.07, 1.73, 0.41, 0.0 ,BL501260
     *  8.12,  4.86,  1.52,  0.47,  0.31,  0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL501270
     *127.30, 79.90, 29.90,  8.00,  3.87,  0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL501280
     *  3.55,  3.41,  1.07,  1.73,  0.41,  0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL501290
     * 32.40, 31.05, 36.77,  8.74,  2.35,  1.46, 0.0 , 0.0 , 0.0 , 0.0 ,BL501300
     *441.50,344.20,353.20,310.90, 31.70, 15.38, 0.0 , 0.0 , 0.0 , 0.0 ,BL501310
     *  4.82,  8.67,  6.54,  5.92,  5.64,  3.99, 0.0 , 0.0 , 0.0 , 0.0 ,BL501320
     *  5.80,  6.10,  6.10,  3.85,  2.85,  0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL501330
     * 56.80, 43.70, 43.70, 43.70, 27.00,  0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL501340
     * 11.80, 13.10, 12.90,  5.35,  0.0 ,  0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL501350
     * 12.10,  7.97,  6.38,  0.45,  0.29,  0.29, 0.29, 0.29, 0.29, 0.29,BL501360
     * 47.10, 27.80, 20.40,  5.58,  2.71,  2.71, 2.71, 2.71, 2.71, 2.71,BL501370
     *  0.08,  0.18,  0.36,  0.03,  0.04,  0.04, 0.04, 0.04, 0.04,  0.0/BL501380
C                                                                      BL501390
C***EXHAUST EMISSION FACTOR SLOPE: 49 STATE                            BL501400
      DATA DEL49/                                                      BL501410
     * 0.58, 0.53, 0.23, 0.23, 0.23, 0.23, 0.0 , 0.0 , 0.0 , 0.0 ,     BL501420
     * 3.06, 6.15, 2.80, 2.30, 2.00, 2.00, 0.0 , 0.0 , 0.0 , 0.0 ,     BL501430
     * 0.0 , 0.0 , 0.0 , 0.00, 0.16, 0.22, 0.0 , 0.0 , 0.0 , 0.0 ,     BL501440
     * 0.58, 0.53, 0.41, 0.41, 0.23, 0.23, 0.0 , 0.0 , 0.0 , 0.0 ,     BL501450
     * 3.06, 6.15, 5.34, 5.34, 2.00, 2.00, 0.0 , 0.0 , 0.0 , 0.0 ,     BL501460
     * 0.0 , 0.0 , 0.0 , 0.0 , 0.11, 0.22, 0.0 , 0.0 , 0.0 , 0.0 ,     BL501470
     * 0.58, 0.53, 0.41, 0.23, 0.23, 0.23, 0.0 , 0.0 , 0.0 , 0.0 ,     BL501480
     * 3.06, 6.15, 5.34, 2.00, 2.00, 2.00, 0.0 , 0.0 , 0.0 , 0.0 ,     BL501490
     * 0.0 , 0.0 , 0.11, 0.22, 0.22, 0.22, 0.0 , 0.0 , 0.0 , 0.0 ,     BL501500
     * 0.58, 0.53, 0.53, 0.53, 1.06, 1.06, 0.0 , 0.0 , 0.0 , 0.0 ,     BL501510
     * 3.06, 6.15, 6.15, 6.15,10.54,10.54, 0.0 , 0.0 , 0.0 , 0.0 ,     BL501520
     * 0.0 , 0.0 , 0.0 , 0.0 , 0.34, 0.34, 0.0 , 0.0 , 0.0 , 0.0 ,     BL501530
     * 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,     BL501540
     * 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,     BL501550
     * 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,     BL501560
     * 1.17, 1.03, 1.03, 0.23, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,     BL501570
     * 1.54, 4.00, 4.00, 2.00, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,     BL501580
     * 0.0 , 0.0 , 0.0 , 0.22, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0   /   BL501590
C                                                                      BL501600
```

```
C***EXHAUST EMISSION FACTOR SLOPE: CALIFORNIA                          BL501610
      DATA DELCA/                                                      BL501620
     * 0.58, 0.70, 0.23, 0.23, 0.23, 0.23, 0.0 , 0.0 , 0.0 , 0.0 ,     BL501630
     * 3.06, 6.40, 2.33, 2.33, 2.00, 2.00, 0.0 , 0.0 , 0.0 , 0.0 ,     BL501640
     * 0.0 , 0.0 , 0.0 , 0.16, 0.16, 0.22, 0.22, 0.0 , 0.0 , 0.0 ,     BL501650
     * 0.58, 0.53, 0.41, 0.23, 0.23, 0.23, 0.23, 0.0 , 0.0 , 0.0 ,     BL501660
     * 3.06, 6.15, 5.34, 5.34, 2.33, 2.00, 2.00, 0.0 , 0.0 , 0.0 ,     BL501670
     * 0.0 , 0.0 , 0.0 , 0.11, 0.22, 0.22, 0.22, 0.0 , 0.0 , 0.0 ,     BL501680
     * 0.58, 0.53, 0.23, 0.23, 0.23, 0.23, 0.23, 0.0 , 0.0 , 0.0 ,     BL501690
     * 3.06, 6.15, 5.34, 2.00, 2.00, 2.00, 2.00, 0.0 , 0.0 , 0.0 ,     BL501700
     * 0.0 , 0.0 , 0.11, 0.22, 0.22, 0.22, 0.22, 0.0 , 0.0 , 0.0 ,     BL501710
     * 0.58, 0.53, 0.53, 0.53, 0.53, 0.53, 1.06, 0.0 , 0.0 , 0.0 ,     BL501720
     * 3.06, 6.15, 6.15, 6.15, 6.15,10.54,10.54, 0.0 , 0.0 , 0.0 ,     BL501730
     * 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.34, 0.0 , 0.0 , 0.0 ,     BL501740
     * 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,     BL501750
     * 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,     BL501760
     * 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,     BL501770
     * 0.58, 0.53, 0.53, 0.23, 0.23, 0.23, 0.23, 0.23, 0.23, 0.23,     BL501780
     * 3.06, 6.15, 6.15, 2.00, 2.00, 2.00, 2.00, 2.00, 2.00, 2.00,     BL501790
     * 0.0 , 0.0 , 0.0 , 0.22, 0.22, 0.22, 0.22, 0.22, 0.22, 0.22 /    BL501800
C                                                                      BL501810
C***EXHAUST EMISSION FACTOR SLOPE: HIGH ALTITUDE                       BL501820
      DATA DELHI/                                                      BL501830
     * 0.55, 0.55, 0.23, 0.23, 0.23, 0.23, 0.23, 0.23, 0.23, 0.23,     BL501840
     * 2.81, 4.24, 2.80, 2.80, 2.80, 2.30, 2.00, 2.00, 2.00, 2.00,     BL501850
     * 0.0 , 0.0 , 0.0 , 0.08, 0.16, 0.16, 0.16, 0.22, 0.22, 0.22,     BL501860
     * 0.55, 0.55, 0.41, 0.41, 0.41, 0.41, 0.23, 0.23, 0.0 , 0.0 ,     BL501870
     * 2.81, 4.24, 5.34, 5.34, 5.34, 5.34, 2.00, 2.00, 0.0 , 0.0 ,     BL501880
     * 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.11, 0.11, 0.22, 0.0 ,     BL501890
     * 0.55, 0.55, 0.41, 0.23, 0.23, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,     BL501900
     * 2.81, 4.24, 5.34, 2.00, 2.00, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,     BL501910
     * 0.0 , 0.0 , 0.11, 0.11, 0.22, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,     BL501920
     * 0.55, 0.55, 0.55, 0.55, 1.06, 1.06, 0.0 , 0.0 , 0.0 , 0.0 ,     BL501930
     * 2.81, 4.24, 4.24, 4.24,10.54,10.54, 0.0 , 0.0 , 0.0 , 0.0 ,     BL501940
     * 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.34, 0.0 , 0.0 , 0.0 , 0.0 ,     BL501950
     * 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,     BL501960
     * 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,     BL501970
     * 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,     BL501980
     * 0.55, 0.55, 0.55, 0.23, 0.23, 0.23, 0.23, 0.23, 0.23, 0.23,     BL501990
     * 2.81, 4.24, 4.24, 2.00, 2.00, 2.00, 2.00, 2.00, 2.00, 2.00,     BL502000
```

```
      * 0.0 , 0.0 , 0.0 , 0.22, 0.22, 0.22, 0.22, 0.22, 0.22, 0.22 /      BL502010
C                                                                          BL502020
      END                                                                  BL502030
```

```
      BLOCK DATA                                                        BL600010
C                                                                       BL600020
      COMMON/IDLBCM/IDLB49,IDLBCA,IDLBHI                                BL600030
      COMMON/IDLDCM/IDEL49,IDELCA,IDELHI                                BL600040
C                                                                       BL600050
      REAL IDLB49(10,3,6),IDLBCA(10,3,6),IDLBHI(10,3,6)                 BL600060
      REAL IDEL49(10,3,6),IDELCA(10,3,6),IDELHI(10,3,6)                 BL600070
C                                                                       BL600080
C***IDLE EMISSION FACTOR INTERCEPT: 49 STATE                            BL600090
      DATA IDLB49/                                                      BL600100
     * 2.01, 0.68, 0.27, 0.03, 0.07, 0.07, 0.0 , 0.0 , 0.0 , 0.0 ,BL600110
     * 16.42, 12.73, 5.43, 0.89, 0.41, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL600120
     * 0.16, 0.26, 0.16, 0.36, 0.22, 0.04, 0.2 , 0.0 , 0.0 , 0.0 ,BL600130
     * 2.01, 0.68, 0.28, 0.24, 0.08, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL600140
     * 16.42, 12.73, 2.02, 1.82, 0.49, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL600150
     * 0.16, 0.26, 0.16, 0.25, 0.18, 0.04, 0.0 , 0.0 , 0.0 , 0.0 ,BL600160
     * 2.93, 1.86, 0.24, 0.08, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL600170
     * 17.24, 18.62, 1.82, 0.49, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL600180
     * 0.18, 0.27, 0.18, 0.04, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL600190
     * 3.85, 0.71, 3.09, 0.73, 0.20, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL600200
     * 24.63, 15.70, 21.92, 19.23, 1.54, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL600210
     * 0.02, 0.04, 0.02, 0.02, 0.01, 0.01, 0.0 , 0.0 , 0.0 , 0.0 ,BL600220
     * 0.50, 0.40, 0.40, 0.25, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL600230
     * 1.32, 0.66, 0.66, 0.66, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL600240
     * 1.11, 1.00, 0.99, 0.27, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL600250
     * 4.05, 2.12, 1.72, 0.13, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL600260
     * 8.27, 4.87, 3.57, 0.65, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL600270
     * 0.01, 0.02, 0.04, 0.01, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 /BL600280
C                                                                       BL600290
C***IDLE EMISSION FACTOR INTERCEPT: CALIFORNIA                          BL600300
      DATA IDLBCA/                                                      BL600310
     * 2.01, 0.93, 0.04, 0.02, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL600320
     * 16.42, 15.02, 0.18, 0.18, 0.14, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL600330
     * 0.16, 0.24, 0.07, 0.11, 0.08, 0.02, 0.02, 0.0 , 0.0 , 0.0 ,BL600340
     * 2.01, 0.68, 0.28, 0.07, 0.04, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL600350
     * 16.42, 12.73, 2.02, 1.72, 0.48, 0.48, 0.0 , 0.0 , 0.0 , 0.0 ,BL600360
     * 0.16, 0.26, 0.16, 0.25, 0.21, 0.09, 0.04, 0.0 , 0.0 , 0.0 ,BL600370
     * 2.93, 1.86, 0.07, 0.04, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL600380
     * 17.24, 18.62, 1.72, 0.48, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL600390
     * 0.18, 0.27, 0.25, 0.09, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL600400
```

```
     *  3.85,  0.71,  3.09,  0.18,  0.05,  0.03, 0.01, 0.0 , 0.0 , 0.0 ,BL600410
     * 24.63, 15.70, 21.92,  4.40,  4.04,  0.32, 0.0 , 0.0 , 0.0 , 0.0 ,BL600420
     *  0.02,  0.04,  0.02,  0.05,  0.05,  0.04, 0.03, 0.0 , 0.0 , 0.0 ,BL600430
     *  0.50,  0.40,  0.40,  0.25,  0.0 ,  0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL600440
     *  1.32,  0.66,  0.66,  0.66,  0.0 ,  0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL600450
     *  1.11,  1.00,  0.93,  0.81,  0.27,  0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL600460
     *  3.46,  1.81,  1.48,  0.32,  0.11,  0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL600470
     * 11.57,  6.82,  5.00,  0.91,  0.0 ,  0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL600480
     *  0.01,  0.02,  0.04,  0.01,  0.0 ,  0.0 , 0.0 , 0.0 , 0.0 , 0.0 /BL600490
C                                                                       BL600500
C***IDLE EMISSION FACTOR INTERCEPT: HIGH ALTITUDE                       BL600510
      DATA IDLBHI/                                                      BL600520
     *  2.29,  1.04,  0.24,  0.17,  0.24,  0.05, 0.05, 0.03, 0.0 , 0.0 ,BL600530
     * 20.70, 20.71,  4.48,  3.01,  4.48,  1.81, 0.84, 0.41, 0.0 , 0.0 ,BL600540
     *  0.26,  0.19,  0.08,  0.06,  0.28,  0.06, 0.14, 0.02, 0.04, 0.0 ,BL600550
     *  2.29,  1.04,  0.22,  0.28,  0.22,  0.39, 0.12, 0.08, 0.0 , 0.0 ,BL600560
     * 20.70, 20.71,  5.60,  2.02,  5.60,  3.75, 1.01, 0.49, 0.0,  0.0, BL600570
     *  0.26,  0.19,  0.08,  0.04,  0.25,  0.04, 0.11, 0.18, 0.04, 0.0 ,BL600580
     *  3.40,  3.50,  0.39,  0.12,  0.08,  0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL600590
     * 38.84, 48.25,  3.75,  1.01,  0.49,  0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL600600
     *  0.10,  0.17,  0.11,  0.18,  0.04,  0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL600610
     *  4.46,  1.34,  5.82,  1.22,  0.32,  0.20, 0.0 , 0.0 , 0.0 , 0.0 ,BL600620
     * 55.59, 40.68, 56.80, 71.15,  3.17,  1.54, 0.0 , 0.0 , 0.0 , 0.0 ,BL600630
     *  0.01,  0.03,  0.02,  0.01,  0.01,  0.01, 0.0 , 0.0 , 0.0 , 0.0 ,BL600640
     *  0.58,  0.75,  0.75,  0.34,  0.25,  0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL600650
     *  2.97,  1.71,  1.71,  1.71,  0.66,  0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL600660
     *  0.64,  0.64,  0.64,  0.27,  0.0 ,  0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL600670
     *  4.01,  3.59,  2.87,  0.20,  0.13,  0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL600680
     * 26.07,  6.68,  4.90,  1.34,  0.65,  0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,BL600690
     *  0.01,  0.01,  0.03,  0.01,  0.04,  0.0 , 0.0 , 0.0 , 0.0 , 0.0 /BL600700
C                                                                       BL600710
C***IDLE EMISSION FACTOR SLOPE: 49 STATE                                BL600720
      DATA IDEL49/                                                      BL600730
     * 0.18, 0.20, 0.07, 0.07, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL600740
     * 2.55, 2.92, 0.83, 0.67, 0.59, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL600750
     * 0.0 , 0.0 , 0.0 , 0.03, 0.06, 0.08, 0.0 , 0.0 , 0.0 , 0.0 ,      BL600760
     * 0.18, 0.20, 0.19, 0.19, 0.06, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL600770
     * 2.55, 2.92, 1.56, 1.56, 0.25, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL600780
     * 0.0 , 0.0 , 0.0 , 0.02, 0.01, 0.02, 0.0 , 0.0 , 0.0 , 0.0 ,      BL600790
     * 0.18, 0.20, 0.19, 0.06, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL600800
```

```
     * 2.55, 2.92, 1.56, 0.25, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL600810
     * 0.0 , 0.0 , 0.01, 0.02, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL600820
     * 0.18, 0.20, 0.20, 0.20, 0.40, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL600830
     * 2.55, 2.92, 2.92, 2.92, 5.00, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL600840
     * 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL600850
     * 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL600860
     * 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL600870
     * 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL600880
     * 0.36, 0.62, 0.62, 0.10, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL600890
     * 1.28, 1.12, 1.12, 0.48, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL600900
     * 0.0 , 0.0 , 0.0 , 0.01, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0   /    BL600910
C                                                                       BL600920
C***IDLE EMISSION FACTOR SLOPE: CALIFORNIA                              BL600930
      DATA IDELCA/                                                      BL600940
     * 0.18, 0.31, 0.01, 0.01, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL600950
     * 2.55, 3.15, 0.08, 0.08, 0.06, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL600960
     * 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL600970
     * 0.18, 0.20, 0.19, 0.05, 0.05, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL600980
     * 2.55, 2.92, 1.56, 1.56, 0.29, 0.29, 0.0 , 0.0 , 0.0 , 0.0 ,      BL600990
     * 0.0 , 0.0 , 0.0 , 0.02, 0.04, 0.04, 0.04, 0.0 , 0.0 , 0.0 ,      BL601000
     * 0.18, 0.20, 0.05, 0.05, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL601010
     * 2.55, 2.92, 1.56, 0.29, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL601020
     * 0.0 , 0.0 , 0.02, 0.04, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL601030
     * 0.18, 0.20, 0.20, 0.20, 0.20, 0.20, 0.40, 0.0 , 0.0 , 0.0 ,      BL601040
     * 2.55, 2.92, 2.92, 2.92, 2.92, 5.00, 0.0 , 0.0 , 0.0 , 0.0 ,      BL601050
     * 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL601060
     * 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL601070
     * 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL601080
     * 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL601090
     * 0.31, 0.28, 0.28, 0.12, 0.12, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL601100
     * 1.80, 3.61, 3.61, 1.18, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL601110
     * 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0   /    BL601120
C                                                                       BL601130
C***IDLE EMISSION FACTOR SLOPE: HIGH ALTITUDE                           BL601140
      DATA IDELHI/                                                      BL601150
     * 0.21, 0.14, 0.02, 0.07, 0.07, 0.07, 0.07, 0.07, 0.0 , 0.0 ,      BL601160
     * 0.53, 1.15, 0.98, 0.83, 0.83, 0.67, 0.59, 0.59, 0.0 , 0.0 ,      BL601170
     * 0.0 , 0.0 , 0.0 , 0.04, 0.06, 0.06, 0.06, 0.08, 0.08, 0.0 ,      BL601180
     * 0.21, 0.14, 0.08, 0.19, 0.19, 0.19, 0.06, 0.06, 0.0 , 0.0 ,      BL601190
     * 0.53, 1.15, 1.24, 1.56, 1.56, 1.56, 0.25, 0.25, 0.0 , 0.0 ,      BL601200
```

```
     * 0.0 , 0.0 , 0.0 , 0.07, 0.02, 0.02, 0.01, 0.01, 0.02, 0.0 ,      BL601210
     * 0.21, 0.14, 0.19, 0.06, 0.06, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL601220
     * 0.53, 1.15, 1.56, 0.25, 0.25, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL601230
     * 0.0 , 0.0 , 0.01, 0.01, 0.02, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL601240
     * 0.21, 0.14, 0.14, 0.20, 0.40, 0.40, 0.0 , 0.0 , 0.0 , 0.0 ,      BL601250
     * 0.53, 1.15, 1.15, 2.92, 5.00, 5.00, 0.0 , 0.0 , 0.0 , 0.0 ,      BL601260
     * 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL601270
     * 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL601280
     * 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL601290
     * 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL601300
     * 0.36, 0.62, 0.62, 0.10, 0.10, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL601310
     * 0.66, 1.12, 1.12, 0.47, 0.48, 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,      BL601320
     * 0.0 , 0.0 , 0.0 , 0.0 , 0.01, 0.0 , 0.0 , 0.0 , 0.0 , 0.0  /     BL601330
C                                                                       BL601340
      END                                                               BL601350
```

```
      BLOCK DATA                                                        BL700010
C                                                                       BL700020
      COMMON/IMCRED/PCIMHN,PCIMHY,PCIMCN,PCIMCY                         BL700030
C                                                                       BL700040
      INTEGER PCIMHN(19,5,4),PCIMHY(19,5,4),PCIMCN(19,5,4),             BL700050
     *  PCIMCY(19,5,4)                                                  BL700060
C                                                                       BL700070
C                                                                       BL700080
      DATA PCIMHN/ 3, 6,11,15,19,24,27,31,35,39,41,42,44,46,47,48,49,51,BL700090
     * 52,   4, 8,13,18,22,28,34,36,40,42,45,46,49,50,51,52,54,56,57,   BL700100
     *       7,14,20,28,34,39,43,46,49,52,54,55,57,58,59,60,61,62,63,   BL700110
     *       8,15,22,30,36,41,46,49,52,54,56,58,59,61,62,63,64,65,65,   BL700120
     *       8,16,23,31,37,42,46,50,53,55,57,59,60,61,62,63,64,65,66,   BL700130
     *       4, 6, 9,11,16,19,20,23,25,28,29,30,32,33,35,36,36,37,38,   BL700140
     *       6,11,16,20,23,27,31,33,36,38,40,41,43,44,45,46,47,48,49,   BL700150
     *      10,18,24,28,33,36,40,42,45,47,48,50,51,53,55,55,55,57,58,   BL700160
     *      12,20,26,32,37,40,43,46,49,51,52,54,55,56,57,59,59,60,61,   BL700170
     *      13,22,28,34,39,43,46,49,51,54,55,57,58,59,61,61,62,63,64,   BL700180
     *       4,11,22,32,42,49,54,58,62,64,66,68,69,71,72,73,74,74,75,   BL700190
     *       4,13,25,37,46,53,58,62,65,67,69,71,72,73,74,75,76,77,77,   BL700200
     *       4,15,28,40,49,55,60,64,67,69,71,72,74,75,76,77,77,78,79,   BL700210
     *       5,17,31,43,51,58,63,66,69,71,73,74,75,76,77,78,79,79,80,   BL700220
     *       5,18,32,44,53,59,63,67,70,72,73,75,76,77,78,79,79,80,81,   BL700230
     *       3, 7,14,23,31,38,44,49,53,56,58,60,62,63,65,66,67,68,69,   BL700240
     *       3, 7,15,24,33,40,46,51,54,57,60,62,63,65,66,67,68,69,70,   BL700250
     *       3, 7,16,25,34,41,47,51,55,58,60,62,64,65,67,68,69,70,71,   BL700260
     *       3, 8,17,26,35,42,48,52,56,59,61,63,65,66,68,69,70,71,71,   BL700270
     *       3, 8,17,27,36,43,49,54,57,60,62,64,66,67,68,69,70,71,72/   BL700280
C                                                                       BL700290
      DATA PCIMHY/ 6,13,17,20,25,32,35,40,45,47,49,50,53,53,54,56,57,57,BL700300
     * 58,   9,14,19,25,31,36,43,45,50,51,53,54,55,56,58,58,61,63,63,   BL700310
     *      12,19,27,35,41,44,51,53,55,57,59,61,61,63,65,65,66,66,67,   BL700320
     *      13,24,33,40,46,50,54,57,59,61,63,65,65,66,68,68,69,70,71,   BL700330
     *      14,25,35,42,48,53,56,59,62,64,65,66,68,69,70,70,71,72,72,   BL700340
     *       9,13,15,18,22,26,28,31,33,35,37,39,40,42,44,44,44,46,48,   BL700350
     *      13,16,20,26,31,34,38,41,43,44,45,47,48,50,52,52,53,54,55,   BL700360
     *      14,21,28,33,38,41,45,47,50,52,53,54,56,57,59,60,61,61,62,   BL700370
     *      15,24,31,36,41,44,48,50,52,54,56,58,58,60,61,62,63,63,65,   BL700380
     *      16,26,33,39,43,47,50,53,55,57,59,60,61,63,64,65,65,66,67,   BL700390
     *      10,33,51,59,67,70,74,76,78,80,80,81,82,83,84,84,85,85,86,   BL700400
```

```
     *       13,36,53,62,69,73,76,78,80,81,82,83,83,84,85,85,86,86,87,   BL700410
     *       15,38,54,64,70,74,77,79,81,82,83,84,85,85,86,86,87,87,87,   BL700420
     *       16,39,54,64,70,74,77,80,81,83,84,85,85,86,87,87,87,88,88,   BL700430
     *       16,39,54,64,70,74,77,80,81,83,84,85,86,86,87,87,88,88,88,   BL700440
     *        7,24,36,52,54,65,66,69,72,75,75,75,77,79,79,80,80,81,81,   BL700450
     *       11,29,45,56,63,68,70,74,76,76,78,80,80,81,82,82,83,84,84,   BL700460
     *       12,32,46,58,65,69,72,75,76,77,80,81,81,82,83,83,84,85,85,   BL700470
     *       14,36,52,62,68,73,76,78,80,81,82,83,84,85,85,86,86,87,87,   BL700480
     *       15,38,54,64,70,74,77,79,81,83,84,85,85,86,87,87,88,88,88/   BL700490
C                                                                       BL700500
C                                                                       BL700510
      DATA PCIMCN/ 7,13,17,22,26,30,32,35,37,40,41,42,44,45,46,47,48,50,BL700520
     * 50,    9,17,22,27,31,35,37,40,42,44,46,48,49,51,52,53,54,55,56,   BL700530
     *       12,21,27,32,36,40,44,47,49,52,53,55,56,58,59,60,61,62,62,   BL700540
     *       14,23,29,35,39,43,47,50,52,55,56,58,59,60,61,62,63,64,65,   BL700550
     *       15,25,31,36,41,45,49,52,54,56,58,59,61,62,63,64,65,65,66,   BL700560
     *       10,15,20,23,26,28,31,32,34,35,37,37,38,40,41,41,42,43,44,   BL700570
     *       17,25,29,32,36,38,41,42,44,46,47,48,49,50,51,52,52,53,54,   BL700580
     *       21,31,36,40,44,47,49,51,53,55,56,57,58,59,60,61,62,62,63,   BL700590
     *       23,34,39,43,46,49,52,54,56,57,59,60,61,62,63,63,64,65,65,   BL700600
     *       25,36,41,45,49,52,54,56,58,60,61,62,63,64,65,66,67,67,68,   BL700610
     *        0, 4, 8,14,20,26,30,33,35,39,42,44,45,48,49,50,50,53,55,   BL700620
     *        0, 5,10,18,23,28,32,37,40,43,45,47,49,51,52,53,54,56,57,   BL700630
     *        1, 6,13,21,27,33,38,42,45,48,50,52,54,56,57,59,60,61,62,   BL700640
     *        1, 6,14,22,28,34,39,43,47,49,52,54,55,57,58,60,61,62,63,   BL700650
     *        2, 7,14,22,29,35,40,44,47,50,52,54,56,58,59,60,61,62,63,   BL700660
     *        0, 2, 7,13,19,24,29,33,37,40,43,45,47,49,50,52,53,54,55,   BL700670
     *        0, 4,11,19,26,32,37,42,45,48,51,53,55,56,58,59,60,62,62,   BL700680
     *        1, 5,13,21,28,34,40,44,48,51,53,55,57,58,60,61,62,63,64,   BL700690
     *        2, 8,18,28,36,42,47,52,55,58,60,62,63,65,66,67,68,69,70,   BL700700
     *        3,12,23,33,42,48,53,57,60,63,65,66,68,69,70,71,72,73,74/   BL700710
C                                                                       BL700720
      DATA PCIMCY/11,20,26,30,34,40,43,47,50,52,53,55,56,57,58,59,61,60,BL700730
     * 62,   16,23,29,35,40,43,49,51,54,55,57,58,59,60,62,62,64,66,66,   BL700740
     *       20,29,37,42,47,49,55,57,59,61,62,64,64,66,67,68,68,68,70,   BL700750
     *       22,34,41,46,51,55,56,61,63,64,65,67,68,69,70,70,71,72,73,   BL700760
     *       23,35,42,48,53,57,60,62,65,66,68,69,70,71,72,72,73,74,74,   BL700770
     *       17,25,27,31,35,39,40,43,44,45,49,50,50,53,54,54,55,56,58,   BL700780
     *       26,33,36,41,46,48,51,54,56,57,58,59,60,61,63,63,64,65,66,   BL700790
     *       30,41,47,50,54,57,59,61,63,65,65,66,67,68,69,70,71,71,71,   BL700800
```

```
     *      72,45,50,54,57,60,63,64,65,67,68,69,70,71,72,72,73,73,74,    BL700810
     *      73,47,52,56,60,62,65,67,68,69,70,71,72,73,74,74,75,75,76,    BL700820
     *       5,24,40,48,56,60,65,67,69,72,72,74,75,76,77,78,78,79,79,    BL700830
     *       8,27,42,51,58,63,66,69,71,73,74,76,76,77,78,79,79,80,80,    BL700840
     *      11,29,43,52,59,64,67,70,72,74,75,76,77,78,79,80,80,81,81,    BL700850
     *      13,30,43,52,59,64,68,71,73,75,76,77,78,79,80,80,81,82,82,    BL700860
     *      14,30,43,52,59,64,68,71,73,75,76,77,78,79,80,81,81,82,82,    BL700870
     *       9,30,41,56,58,68,68,71,74,77,77,78,79,81,81,81,81,82,83,    BL700880
     *      15,37,51,60,66,70,72,76,77,78,80,82,82,82,83,84,84,85,85,    BL700890
     *      18,39,52,62,68,71,74,77,78,79,82,82,82,84,84,84,85,86,86,    BL700900
     *      21,43,57,66,71,75,78,80,81,83,84,85,85,86,86,87,87,88,88,    BL700910
     *      23,44,58,66,72,76,79,81,83,84,85,86,86,87,88,88,88,89,89/    BL700920
C                                                                        BL700930
      END                                                                BL700940
```

SECTION 8

REFERENCES

1. Dabbert, W.F. and R.C. Sandys and P.A. Buder, ISMAP - A Traffic/Emissions/Dispersion Model for Mobile Pollution Sources. User's Manual. Prepared for California Business Properties Association by Stanford Research Institute, Menlo Park, California 94025.

2. Automobile Exhaust Emission Modal Analysis Model, EPA-460/3-74-005, January 1974.

3. Newell, G.F., Approximate Methods for Queues with Application to the Fixed Cycle Traffic Light, S.I.A.M. Review. Vol. 7, No. 2.

4. Webster, F.V., Traffic Signal Settings. Road Research Technical Paper No. 39, Road Research Laboratory, H.M. Stationary Office, London, England.

5. Technical Guidelines for the Review of Indirect Sources (Draft). Stanford Research Institute, Menlo Park, California. Prepared for the U.S. Environmental Protection Agency, Research Triangle Park, N.C. June 1976.

6. Benesh, F. User's Manual for the Intersection-Midblock Model. GCA/Technology Division, Bedford, Massachusetts. Prepared for the U.S. Environmental Protection Agency, Research Triangle Park, N.C. July 1978.

7. Johnson, W.B., W.F. Dabberdt, F.L. Ludwig, and R.J. Allen. Field Study for Initial Evaluation of an Urban Diffusion Model for Carbon Monoxide. Coordinating Research Council and Environmental Protection Agency, Contract CAPA-3-68 (1-69), Stanford Research Institute, Menlo Park, California. 1971.

8. Mancuso, R.L. and F.L. Ludwig. User's Manual for the APRAC-1A Urban Diffusion Model Computer Program. Contract CAPA-3-68 (1-69), Stanford Research Institute, Menlo Park, California. 119 pp, (NTIS-PB 213 091). 1972.

9. Highway Research Board, Highway Capacity Manual 1965, Special Report 87, NAS-NRC, Washington, D.C. 1965.

10. Ludwig, F. L., W. B. Johnson, A. E. Moon, and R. L. Mancuso. A Practical Multipurpose Diffusion Model for Carbon Monoxide. Final Report, Contracts CAPA-3-68 and CPA 22-69-64, Stanford Research Institute, Menlo Park, Califronia. 184 pp., (NTIS-PB 196 003). 1970.

11. Johnson, W. B., W. F. Dabberdt, F. L. Ludwig, and R. J. Allen. Field Study for Initial Evaluation of an Urban Diffusion Model for Carbon Monoxide. Comprehensive Report, Contract CAPA-3-68 (1-69), Stanford Research Institute, Menlo Park, California. 240 pp., (NTIS-PB 203 469). 1971.

12. Ludwig, F. L. and W. F. Dabberdt. Evaluation of the APRAC-1A Urban Diffusion Model for Carbon Monoxide. Final Report, Contract CAPA-3-68 (1-69), Stanford Research Institute, Menlo Park, California. 167 pp., (NTIS-PB 210 819). 1972.

13. Compilation of Air Pollutant Emission Factors, Supplement 5 (AP-42). U.S. Environmental Protection Agency, Research Triangle Park, N.C. 27711. April 1975.

14. Guthman, L. User's Guide to MOBILE1. U.S. EPA, Washington, D. C. 1978.

15. Ludwig, F., et al. User's Manual for the APRAC-2. Emissions and Diffusion Model. Stanford Research Institute, Menlo Park, California. 1977.

16. Wiltsee, K and Benesh, F. Documentation of Modifications to APRAC-2 for Incorporating 1978 Emission Factors. GCA/Technology Division, Bedford, Massachusetts. Prepared for Atlanta Regional Commission, Atlanta, Ga. (Model modifications have been completed and is currently being used; documentation will be published September 1978).

TECHNICAL REPORT DATA
(Please read Instructions on the reverse before completing)

1. REPORT NO. EPA-450/3-78-040	2.	3. RECIPIENT'S ACCESSION NO.
4. TITLE AND SUBTITLE Carbon Monoxide Hot Spot Guidelines Volume VI: User's Manual for the Modified ISMAP Model		5. REPORT DATE August 1978
		6. PERFORMING ORGANIZATION CODE
7. AUTHOR(S) Frank Benesh		8. PERFORMING ORGANIZATION REPORT NO. GCA-TR-78-32-G(6)
9. PERFORMING ORGANIZATION NAME AND ADDRESS GCA Corporation GCA/Technology Division Burlington Road Bedford, Massachusetts 01730		10. PROGRAM ELEMENT NO. 2AF643
		11. CONTRACT/GRANT NO. 68-02-2539
12. SPONSORING AGENCY NAME AND ADDRESS U.S. Environmental Protection Agency Office of Air Quality Planning and Standards Research Triangle Park, North Carolina 27711		13. TYPE OF REPORT AND PERIOD COVERED
		14. SPONSORING AGENCY CODE

15. SUPPLEMENTARY NOTES

16. ABSTRACT

A modified version of the ISMAP model has been developed for the analysis of carbon monoxide hot spot locations. The original version of ISMAP (Indirect Source Model for Air Pollution) was developed by Stanford Research Institute to be used in the evaluation of indirect source impact upon ambient carbon monoxide concentrations. Due to the size and complexity of the original version of the model, the modifications were made without changing the basic structure of the program. Since the analysis of parking lot traffic and emissions was an integral part of the original model structure, it was left in the modified version, but the code was modified to negate their effect upon traffic flow and air quality. In the modified ISMAP, internal zones (indirect sources) are used together with external zones to generate flow within a traffic network. This modified version of the model sets parking lot emissions equal to zero and provides a near infinite parking lot capacity so that vehicle routing between internal zones will not occur because of parking lot overcrowding. Other modifications made to ISMAP include the incorporation of a street canyon submodel. Version 2 of modified ISMAP utilized the most recent (1978) motor vehicle emission correction factors and modal analysis model coefficients and deterioration. In other aspects, it is unchanged from Version 1, written by Michael T. Mills.

17. KEY WORDS AND DOCUMENT ANALYSIS

a. DESCRIPTORS	b. IDENTIFIERS/OPEN ENDED TERMS	c. COSATI Field/Group
	Carbon Monoxide Models Air Quality Models Traffic Models	

18. DISTRIBUTION STATEMENT RELEASE UNLIMITED	19. SECURITY CLASS *(This Report)* UNCLASSIFIED	21. NO. OF PAGES 230
	20. SECURITY CLASS *(This page)* UNCLASSIFIED	22. PRICE

EPA Form 2220-1 (9-73)