EPA-600/3-81-044
May 1983

DOCUMENTATION FOR
WATER QUALITY ANALYSIS SIMULATION PROGRAM (WASP)
AND
MODEL VERIFICATION PROGRAM (MVP)

by

Dominic M. DiToro*
James J. Fitzpatrick*
Robert V. Thomann*
Hydroscience, Inc., Westwood, N.J. 07675
*(Presently at Hydroqual Inc., Mahwah, New Jersey 07430)

In Partial Fulfillment of
Contract No. 68-01-3872

Project Officer

William L. Richardson
Large Lakes Research Station
U.S. Environmental Protection Agency
Grosse Ile, Michigan 48138

ENVIRONMENTAL RESEARCH LABORATORY
OFFICE OF RESEARCH AND DEVELOPMENT
U.S. ENVIRONMENTAL PROTECTION AGENCY
DULUTH, MINNESOTA 55804

NOTICE

This document has been reviewed in accordance with U.S. Environmental Protection Agency policy and approved for publication. Mention of trade names or commercial products does not constitute endorsement or recommendation for use.

FOREWORD

One of the primary goals of the U.S. EPA's research program on Great Lakes is to develop methodologies for simulating and predicting transport and fate of contaminants in aquatic systems. Although methodologies have specific research applications to the Great Lakes, they may be useful for other water bodies. The computer programs documented in this report have been used by researchers at Manhattan College and the EPA Large Lakes Research Station for research ranging from transport of conservative tracers like chloride to complex interactions and transport of phytoplankton and nutrients, and finally of toxic substances with suspended solids.

In documenting these programs, our intent is to allow researchers and engineers to develop new theories and apply existing models to their problem. Operating complex, dynamic models should be approached with caution, however. An experienced Fortran programmer should be employed to operate and modify the computer programs to fit the computer and application confronted. Experienced environmental scientists and engineers and preferably a team consisting of biologists, limnologists, and hydrodynamicists would ideally be involved in a program to develop and apply models and interpret results. Modeling research should dovetail with surveillance and experimental research which provides calibration and verification data and estimates of model process rates.

No claims are made that the programs are applicable to every problem no that they are error free. Procedures for obtaining copies of model codes can be obtained by writing the project officer or contacting the EPA Water Quality Modeling Center, ERL-Athens.

William L. Richardson
Environmental Scientist
Large Lakes Research Station
9311 Groh Road
Grosse Ile, Michigan 48138

ABSTRACT

A generalized water quality modeling program and a model verification analysis program have been developed that have application to a wide variety of water resource management problems. The Water Quality Analysis Simulation Program, WASP, is based on the flexible compartment approach. It may be applied to water bodies in a one, two, or three-dimensional configuration, and kinetic models may be structured to include linear and non-linear reactions. The user may choose, via input options, to employ constant or time-variable transport and kinetic processes, as well as point and non-point waste discharges. The Model Verification Program, MVP, may be used as an indicator of "goodness of fit" or adequacy of the model as a representation of the real world.

To date, WASP has been applied to over twenty water resource management problems. These applications have included one, two and three-dimensional configurations and a number of different physical, chemical and biological modeling frameworks, such as BOD-DO, eutrophication, and toxic substances.

A user's manual and program listings have been prepared. The user's manual was oriented towards the system analyst, whose responsibility it would be to design, develop and debug new kinetic models for end users, as well as the end user who must prepare the data input to the program.

CONTENTS

Appendices

FIGURES

TABLES

ACKNOWLEDGMENTS

The authors are especially grateful to William L. Richardson for his suggestions and assistance in implementing WASP and MVP on the EPA computer systems, and for his hospitality during their visits to Grosse Ile. The assistance provided by Dennis Parrott and Steve Golus in implementing WASP and MVP is also acknowledged.

Appreciation is given to the following Hydroscience staff members who contributed to this project: Richard Askins, for developing MVP; Richard Winfield, for his assistance in implementing the LAKE 1 kinetics in WASP and the verification data sets in MVP; Eric Ringewald, for assistance in implementing WASP and MVP on the EPA COMNET system, Kathleen Whartenby, for typing this document.

Recognition is also given Joan Nies for assisting in editing this documentation report and to Debra L. Caudill for retyping and formatting the draft for final publication.

SECTION 1

SUMMARY AND CONCLUSIONS

This report provides detailed documentation for the Water Quality Analysis Simulation Program (WASP) and the Model Verification Program (MVP). These two programs provide a generalized computational framework to be used as part of the decision making process in water resource management problems.

This report is addressed to the needs of the systems analyst, whose responsibility it will be to design, develop and debug new kinetic models, as well as the end user, whose main task it might be to implement an already developed kinetic subroutine for a new river, lake, estuarine or ocean environment. As such, the report will detail the input formats needed to run the WASP and MVP programs for the end user, and provide a complete description of the program logic of WASP and a detailed procedure to following for the programming of new kinetic routines for the systems analyst.

CONCLUSIONS

WASP has proved itself to be a very versatile program, capable of studying time variable or steady state, one, two or three dimensional, linear or non-linear kinetic water quality problems. To date WASP has been employed in over twenty modeling applications that have included river, lake, estuarine and ocean environments and that have investigated dissolved oxygen, bacterial, eutrophication and toxic substance problem contexts. MVP provides a statistical framework to assist a user in determining whether a model developed utilizing WASP is a "reasonable" representation of the real world. MVP permits the user to make a detailed comparison of the model to observed field data, so as to made a judgment concerning the adequacy of the model. Naturally, the firmer the faith in the adequacy of the model the more likely it will be able to play a useful role in the water resource management decision making process.

Model Limitation

WASP does not compute hydrodynamics. As a result, the transport mechanisms, both advective and dispersive, must be specified by the user. The user may choose to generate the transport input data from a separate program capable of performing hydrodynamic computations; or by calibrating the model against conservative substances such as salinity, electrical conductivity, or dye from a dye tracer study.

WASP cannot solve directly for steady state applications. Instead the program must numerically integrate the differential equations involved in a model structure until steady state is achieved, a very CPU time consumina approach.

SECTION 2

RECOMMENDATIONS

1. The Water Quality Analysis Simulation Program (WASP) should be put to use in as many applications, both research and real problems, as possible. The evaluation of WASP and the development of new kinetic structures are important to demonstrate the use of WASP as an acceptable tool in addressing water quality modeling problems.

2. The Model Verification Program (MVP) can be used as one methodology for judging model "accuracy" during the calibration and verification stages of model development.

3. Due to the generality of the WASP program, the user should be knowledgeable in water quality modeling techniques, and in particular knowledgeable in the associated physical, chemical and biological principles that are to be incorporated in the modeling framework. The ultimate responsibility for a model's success lies entirely with the modeler, as WASP just provides a computational framework for the development of a model.

4. The model developer, including both the system analyst and the programmer member of an administrator/engineering/programmer team, should be knowledgeable in FORTRAN programming and operating systems interfacing.

5. The end user should review this report in some detail to fully understand the principal underlying assumptions of the WASP/MVP package.

SECTION 3

INTRODUCTION

BACKGROUND

The application of mathematical modeling techniques to water quality problems has proved to be a powerful tool in water resource management. As a diagnostic tool, it permits the abstraction of a highly complex real world. Realizing that one can never fully expect to detail all the physical phenomena that comprise our natural world, one attempts to identify and include only the phenomena, be they natural or man-made, that are relevant to the water quality problem under consideration. As a predictive tool, mathematical modeling permits the forecasting and evaluation of the effects of changes in the surrounding environment on water quality. Although engineering insight and political and socio-economic concerns play important roles in water resource management, some water quality problems are of such a highly complex nature that the predictive capability of mathematical models provides the only real means for screening the myriad number of management alternatives.

It is important for a computer program that is to serve as the basis for the mathematical modeler to be very general in nature. The program should be flexible enough to provide the modeler with the mechanisms to describe the kinetic process and the inputs to these processes, as well as the transport processes and the geophysical morphology or setting, that go into the framework of the model. Transport processes, basically hydrodynamic in nature, include advection, turbulent diffusion, and, when spatial averaging is included, dispersion. Kinetic (or reactive) processes are the sources and sinks which act upon a particular water quality parameter and may be physical, chemical or biological: for example, sedimentation and flocculation of organics, the assimilative capacity of a water body to receive an acid waste discharge and the predator-prey relationship of zooplankton-phytoplankton.

WASP

While many of the water quality modeling programs available today, DOSAG-I (Texas Water Quality Board, 1970), ES001 (EPA/Hydroscience, 1970), QUAL-I (Masch, 1971), QUAL-II (WRE, 1974), the RECEIV block of SWMM (EPA/-WRE), DEM (WRE), and the Hydrocomp Simulation Model (Hydrocomp), provide flexibility in some of the aforementioned areas, no one program provides the increased flexibility afforded by the Water Quality Analysis Simulation Program (WASP) developed by Hydroscience in 1970. WASP permits the modeler to structure one, two, and three-dimensional models; allows the specification of time-variable exchange coefficients, advective flows, waste loads and water quality boundary conditions; permits the structuring of the kinetic processes, within the larger modeling framework, without having to write or rewrite large sections of computer code. Although WASP's multi-dimensionality and time-variable input capabilities are strong points, it is probably the ease with which one may develop new kinetic or reactive structures that is WASP's main strength. However, WASP's generality requires an additional measure of judgment and insight on the part of the modeler. The kinetic and transport structures are not "hard wired" in WASP (i.e., the equations are not "fixed" and "buried" in the code). Therefore, the burden is on the modeler (perhaps together with a programmer) to write the applicable kinetic equations (or use those already implemented) for a given problem context.

APPLICATIONS

WASP has been used in many modeling frameworks since 1970. Table 1 is a partial listing of the applications made to date of the WASP program. As shown, a variety of problem contexts, varying from the more traditional D.O. and bacterial problems, to eutrophication problems and the fate of hazardous substances, have been studied. Spatial scales have ranged from meters to 100 km and time scales have ranged from minute to minute simulation of a discharge of sulfuric acid to a 10 year long year to year analysis of phytoplankton and nutrients in Lake Ontario. Kinetic structures have encompassed simple linear kinetics, interactive linear kinetics, and a wide range of non-linear interactive kinetic frameworks. One of the first applications of WASP was in modeling eutrophication or algal population dynamics in the Western Delta-Suisun Bay region of San Francisco Bay (1,2). The model segmentation and the systems diagram of the model are shown in Figures 1 and 2 respectively. The model incorporated time variable (monthly averaged) flows, nutrient waste loadings, time-variable boundary conditions, and spatially, as well as temporally, resident extinction coefficients. In addition, the bulk dispersion coefficients were adjusted as a function of net Delta outflow. Figure 3 presents some model calibration results for the principal state variables.

The basic WASP structure has also been applied to eutrophication analyses of the Great Lakes, specifically:

a. Lake Ontario, including the Rochester Embayment
b. Lake Erie
c. Lake Huron, including Saginaw Bay
d. Lake Michigan

A variety of spatial scales have been utilized in the framework as illustrated in Figure 4, where for Lake Ontario, the range in longitudinal scale was from 10-100 km^2 (Rochester embayment model) to 13,000 km^2 (the whole lake model). Details on these analyses are given in a series of reports on Great Lakes eutrophication models (3,4,5,6,7).

TABLE 1. PARTIAL LIST OF APPLICATIONS OF WASP PROGRAM (1)

Geographical Area	Spatial Dimensionality	Number of Spatial Segments	Number of State Variables	Principal State Variables	Time Scale	Remarks
Eutrophication Problem Contexts						
Western Delta-Suisun Bay Area of San Francisco Bay	2 Dimensional Laterally	39	11	Chlorophyll, Nitrogen, Silica, DO	Week-Week	Original problem setting for development of WASP-1970
Potomac Estuary	2 Dimensional	36	8	Chlorophyll, Nitrogen, Phosphorus	Week-Week	
GREAT LAKES						
Lake Ontario	1 Dimensional Vertically	2	13	Chlorophyll, Zooplankton, Phosphorus, Nitrogen, Silica	Week-Week and Year-Year	Lake1 Model
Lake Ontario	3 Dimensional	67	8	Chlorophyll, Zooplankton, Phosphorus, Nitrogen, Silica	Week-Week and Year-Year	Lake3 Model
Rochester Embayment	3 Dimensional	72	8	Chlorophyll, Zooplankton, Phosphorus, Nitrogen, Silica	Week-Week and Year-Year	

TABLE 1. (CONT.)

Geographical Area	Spatial Dimensionality	Number of Spatial Segments	Number of State Variables	Principal State Variables	Time Scale	Remarks
Eutrophication Problem Contexts, Cont.						
Lake Huron	3 Dimensional					
Saginaw Bay	2 Dimensional Laterally	5	8	Chlorophyll, Zoop-plankton, Phosphorus, Nitrogen, Silica	Week-Week and Year-Year	
Lake Erie	3 Dimensional					Includes sediment interactions
Chesapeake Bay	2 Dimensional Laterally	49	5	Chlorophyll, Nitrogen, Phosphorus	Steady State	Simplified eutrophication kinetics
Trinity River						
Tennessee Colony	2 Dimensional Vertically	15	7	Chlorophyll, Nitrogen, Phosphorus		
Lake Livingston	2 Dimensional	14	8	Chlorophyll, Nitrogen, Phosphorus		
Upper Mississippi Minneapolis 208	3 Dimensional	120	12	Chlorophyll, Nitrogen, Phosphorus, DO, Bacteria	Steady State	

TABLE 1. (CONT.)

Geographical Area	Spatial Dimensionality	Number of Spatial Segments	Number of State Variables	Principal State Variables	Time Scale	Remarks
Dissolved Oxygen, Bacteria Problem Contexts						
Upper Delaware River	1 Dimensional	120	3	BOD, DO	Hour-Hour	Included rooted aquatic plants
New York Harbor (208)	3 Dimensional	425	6	BOD, DO, Bacteria	Day-Day	Interfaced to landside simulator
Milwaukee River	1 Dimensional	67	3	BOD, DO, Bacteria	Hour-Hour	
Toxic Substance Problem Contexts						
Hudson River - NY Harbor PCB	1 Dimensional	21	3	Suspended solids, Dissolved PCBs, Particulate PCBs	Steady State	
Quarry - DDE and Lindane	1 Dimensional Vertically	15	5	Suspended solids, Dissolved and Particulate DDE and Lindane	Minute-Minute	
River - pH Spill	2 Dimensional Laterally	78	2	Bicarbonate equilibrium - pH	Minute-Minute	
Ocean Disposal - Acid Waste	2 Dimensional Vertically	120	1	Acid Waste	Minute-Minute	Included stratification due to thermocline

TABLE 1. (CONT.)

Geographical Area	Spatial Dimensionality	Number of Spatial Segments	Number of State Variables	Principal State Variables	Time Scale	Remarks
Biological Waste Treatment Problem Contexts						
Anaerobic Filter		20	12	COD, DO, pH, Organic Nitrogen	Steady State	3 Phases-Solids, Liquid, Gas
Trickling Filter		20	2	Substrate, DO	Steady State	1 or 2 reactors in series, recycle
Pure Oxygen		4	6	COD, DO, Nitrogen, pH	Steady	Gas - Liquid phase

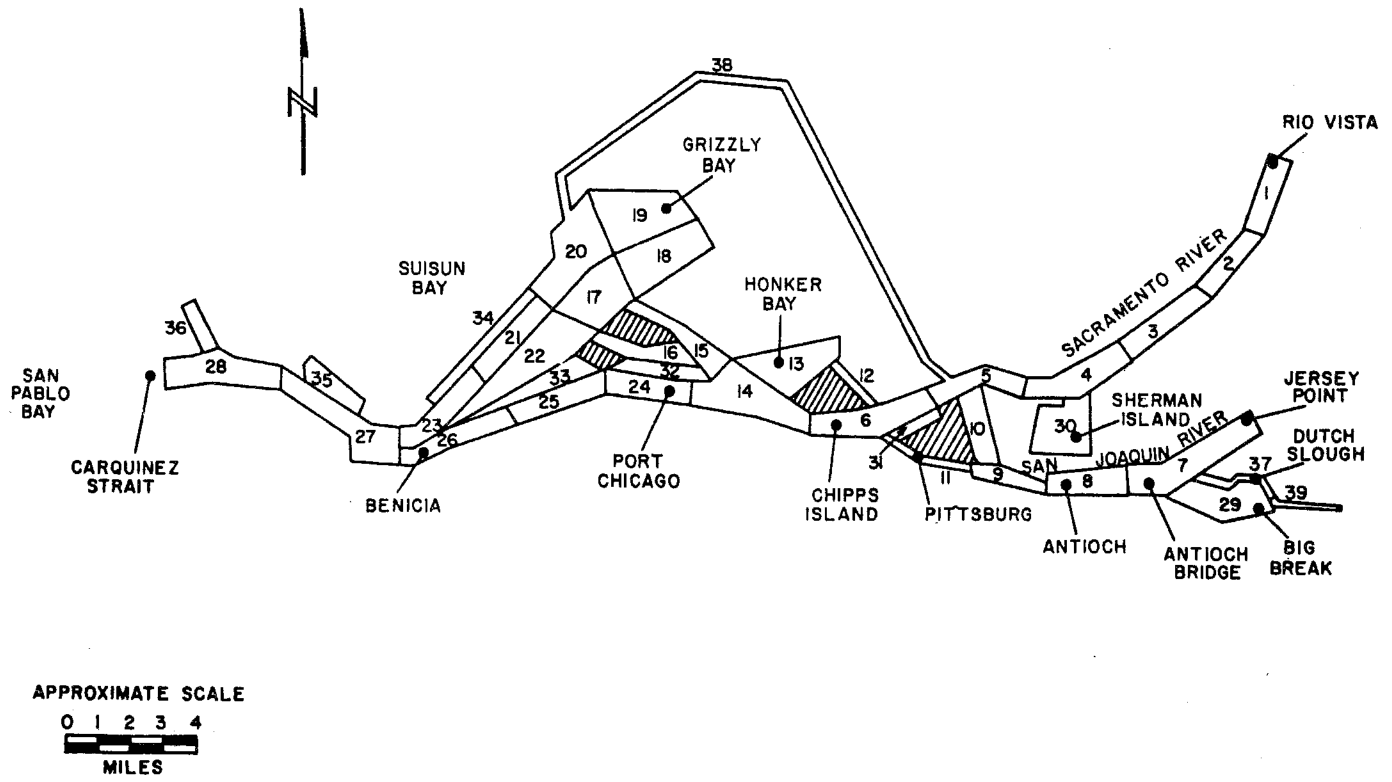


Figure 1. Segmentation map for the western delta-suisun bay study.

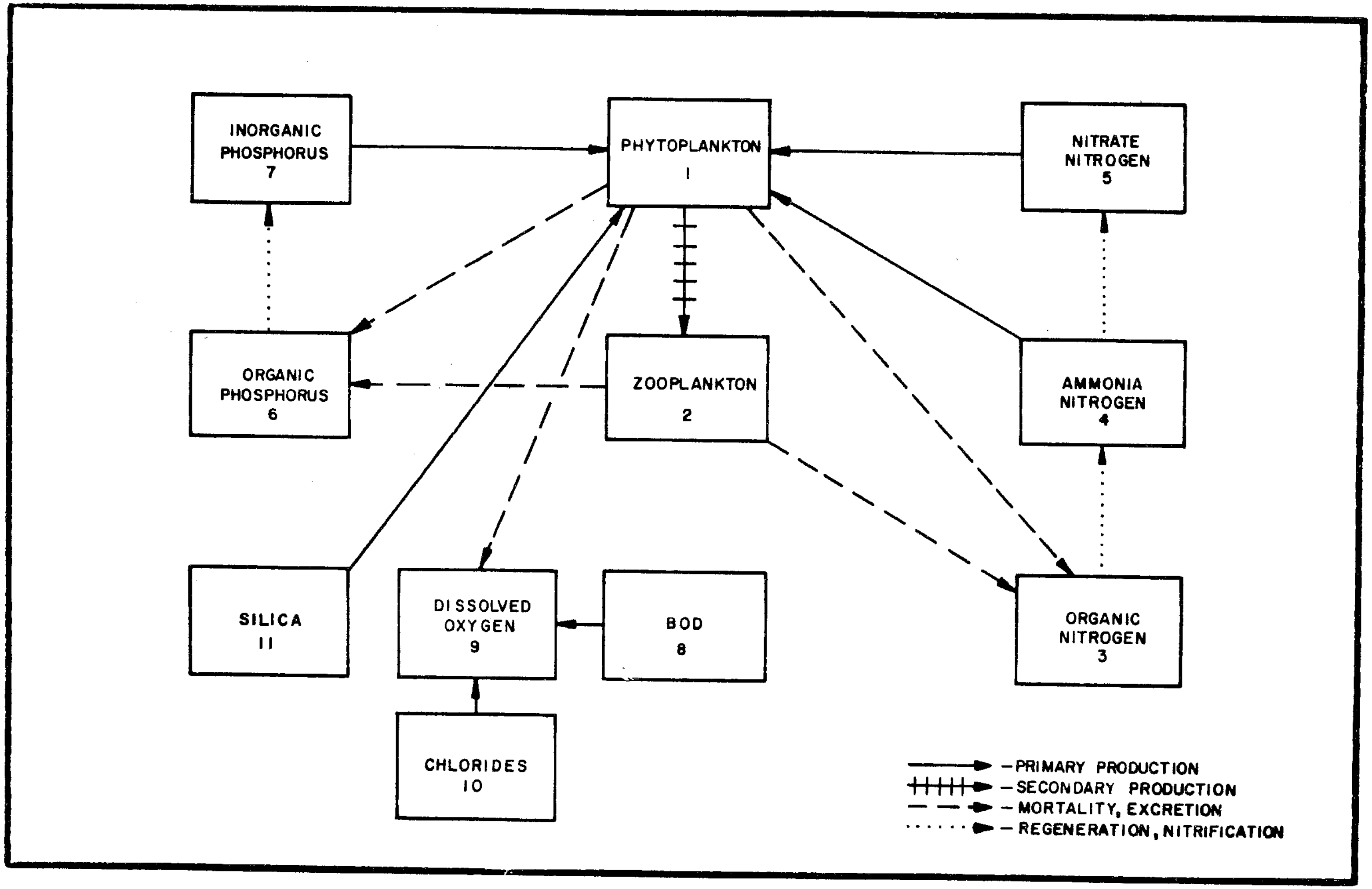


Figure 2. Systems diagram for the western delta-suisun bay study.

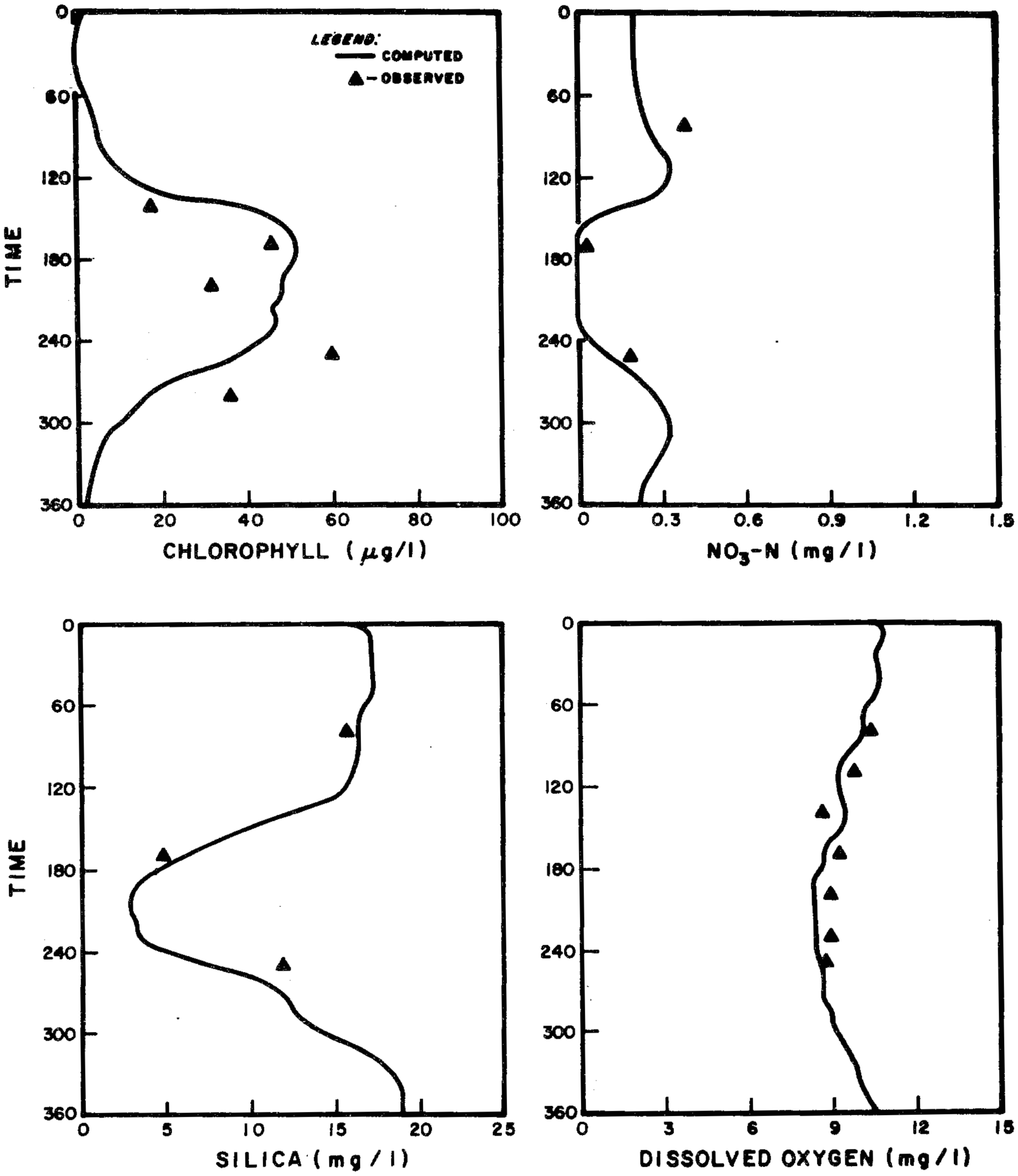


Figure 3. Model calibration observed vs. computed.

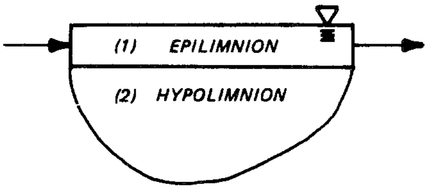
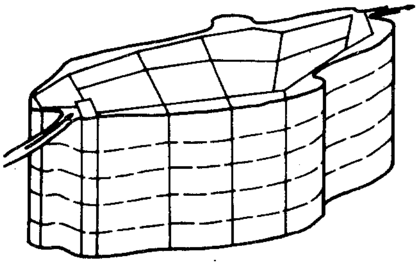
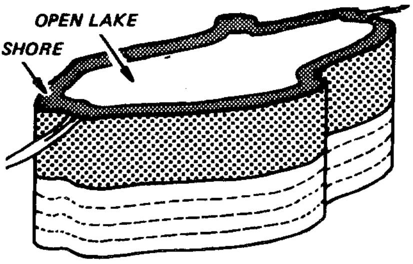
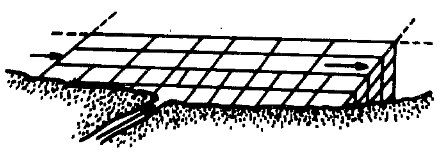
MODEL DESIGNATION		NUMBER OF SEGMENTS	HORIZONTAL SCALE (km^2) EPILIMNION SEGMENTS
LAKE 1		2	13,000
LAKE 3		67	200–1000
LAKE 3 (AGGREGATED)		8	6000–13,000
ROCHESTER EMBAYMENT		72	10–100

Figure 4. Spacial scales used in the Lake Ontario analysis.

As an illustration of the application of WASP to lake models, the earlier work in Lake Ontario can be considered. This work utilized a whole lake analysis, vertically segmented at the average thermocline depth. This model, designated the LAKE1 model to indicate its vertical one-dimensional nature, extended the earlier work on eutrophication of estuaries to a large lake such as Lake Ontario.

Figure 5 shows the major physical features of the LAKE1 model. The principal physical features are a) horizontal transport, b) time variable vertical exchange between the epilimnion and the hypolimnion to permit conditions of a vertically mixed lake or a vertically stratified lake and c) vertical setting of phytoplankton and other particulate nutrient forms. The systems diagram for the LAKE1 model is shown in Figure 6 where eight variables were implemented: the nitrogen and phosphorus variables, chlorophyll and the two zooplankton groups. This kinetic structure was later expanded in the Lake Erie work (6), and also subsequently incorporated in further work on Lake Ontario. The updated Lake Erie and Lake Ontario kinetic structure is shown in Figure 7. This expansion included the addition of silica as a state variable (in two forms) and the division of the phytoplankton variable into "Diatom" and "Non-Diatom". In each case, the coupling between the variables is both linear and nonlinear and the parameters may be specified as time dependent straight line functions. A typical calibration using the updated kinetics, LAKE1A, is shown in Figure 8.

The WASP structure proved particularly useful in all of the Great Lakes work by permitting the ready expansion of models in the spatial dimension as well as expansion of the kinetic structures. In each application for each lake system, the principal effort was in deciding on the spatial configuration to be used, together with the appropriate kinetics. Once the decision was made, WASP permitted the formulation of a "new" lake model by requiring only a minimum amount of effort to write the specific kinetics for insertion into the program and in the preparation of any new input on transport and dispersion.

WASP has also been used as part of studies on Lake Ontario in a long term, year to year analysis of lake behavior (8). Such an analysis provides a basis for estimating lake responses under different external nutrient loadings. In this mode, the seasonal dynamics were computed for each year and the output after a two-year run became the initial conditions for a subsequent two-year run. Output from each two-year run is saved on a permanent file for computations using the Model Verification Program, MVP.

In the multi-year analysis, various kinetic schemes were employed to arrive at the best overall model that represented the long-term data. The WASP framework proved particularly useful in this type of study. Figure 9 shows the comparison between the observed chlorophyll data in the epilimnion for 1967-76 and the calculated values using the LAKE1A kinetics. It should be noted that the results shown in Figure 9 were calculated by setting initial conditions in 1966 and continuing the calculation to 1976 without any resetting of conditions. Actual year to year temperature data and loads were used.

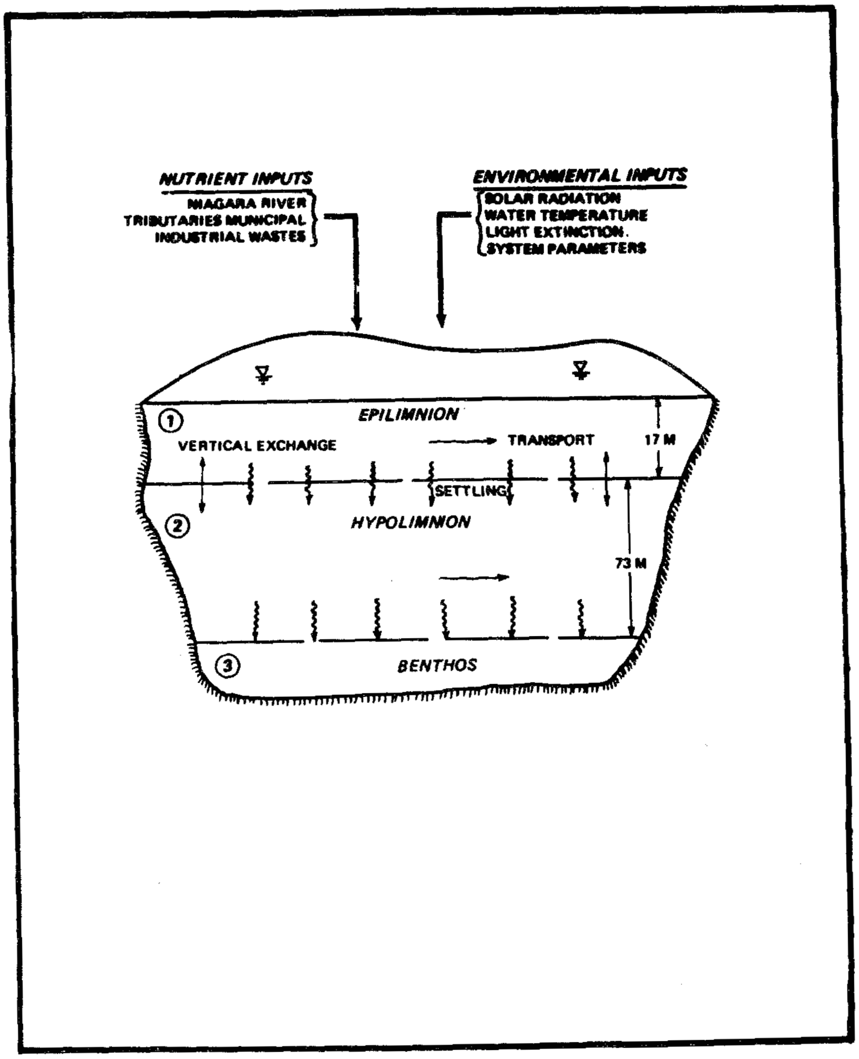


Figure 5. Major physical features included in the LAKE1 model.

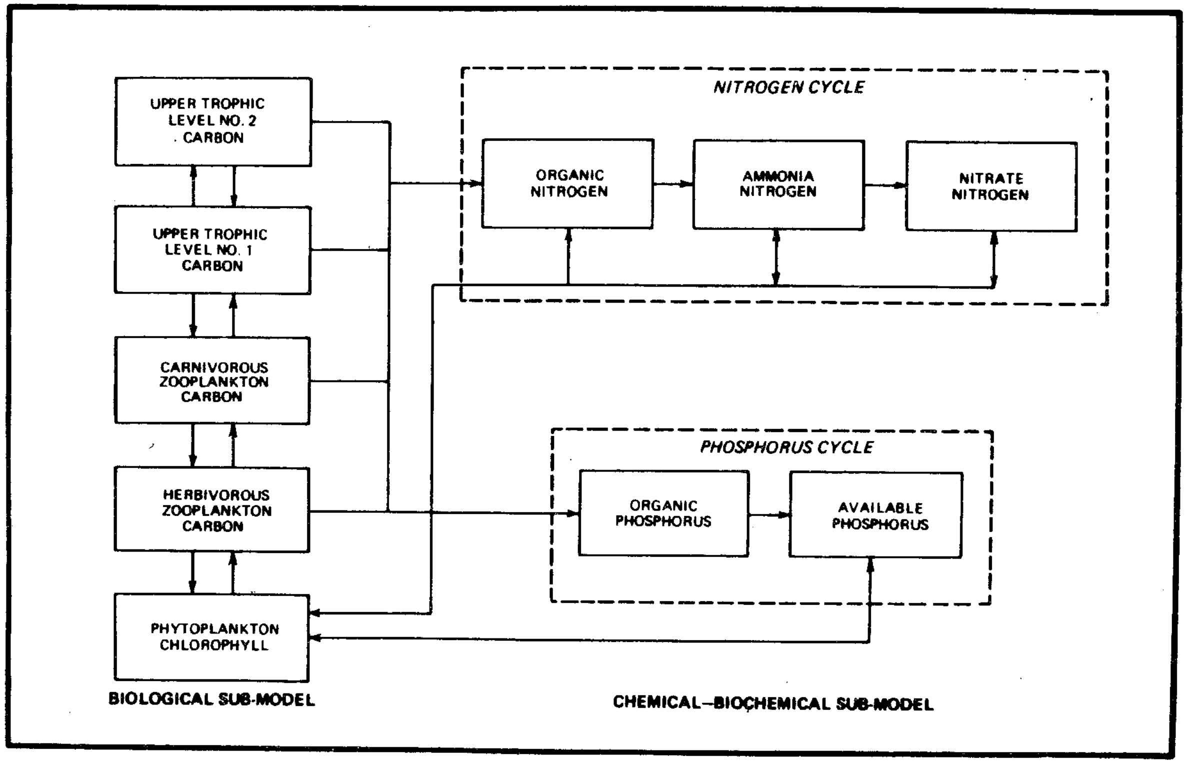

NITROGEN CYCLE
ORGANIC NITROGEN
AMMONIA NITROGEN
NITRATE NITROGEN
PHOSPHORUS CYCLE
ORGANIC PHOSPHORUS
AVAILABLE PHOSPHORUS
UPPER TROPHIC LEVEL NO. 2 CARBON
UPPER TROPHIC LEVEL NO. 1 CARBON
CARNIVOROUS ZOOPLANKTON CARBON
HERBIVOROUS ZOOPLANKTON CARBON
PHYTOPLANKTON CHLOROPHYLL
BIOLOGICAL SUB-MODEL
CHEMICAL–BIOCHEMICAL SUB-MODEL

Figure 6. Systems diagram - LAKE1 model.

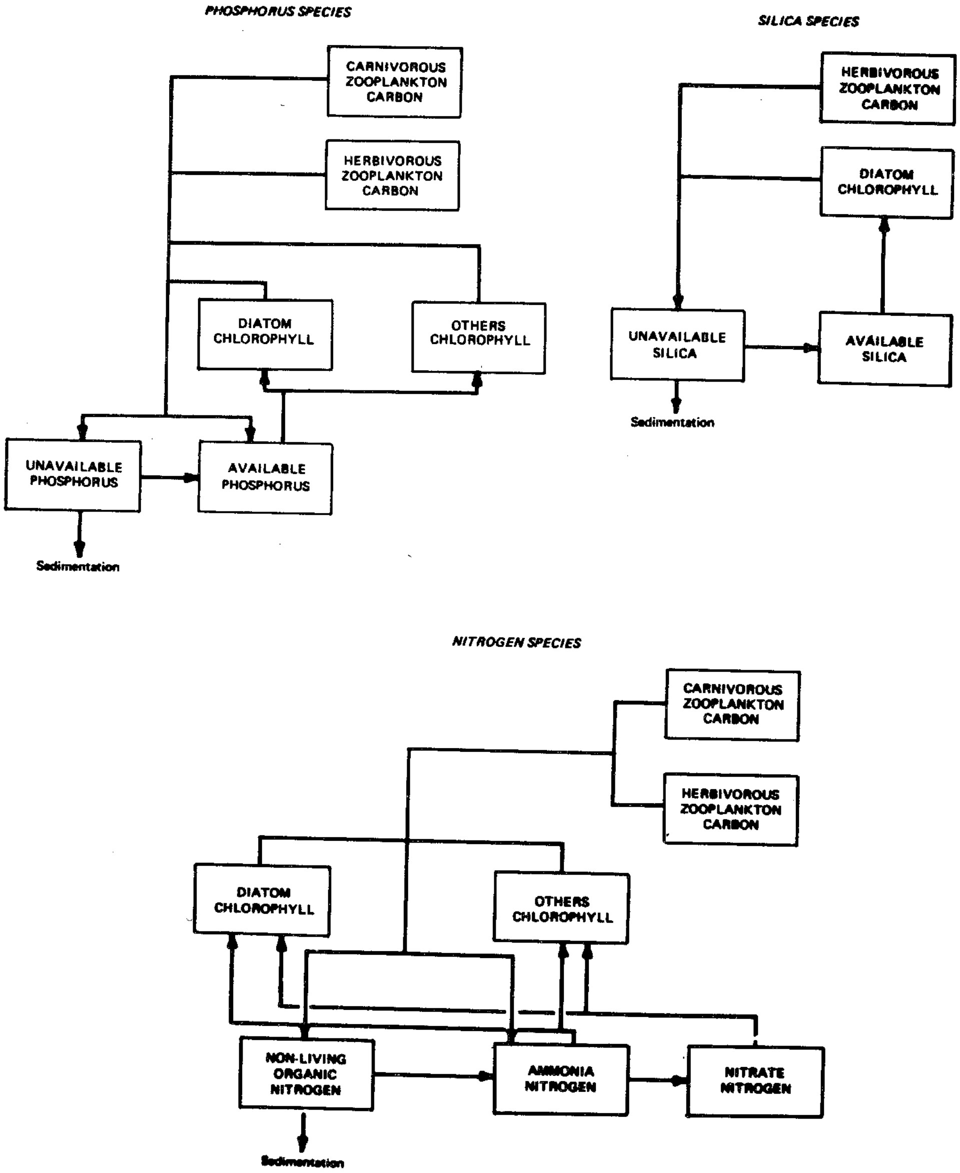


Figure 7. Systems diagram - updated lake kinetics.

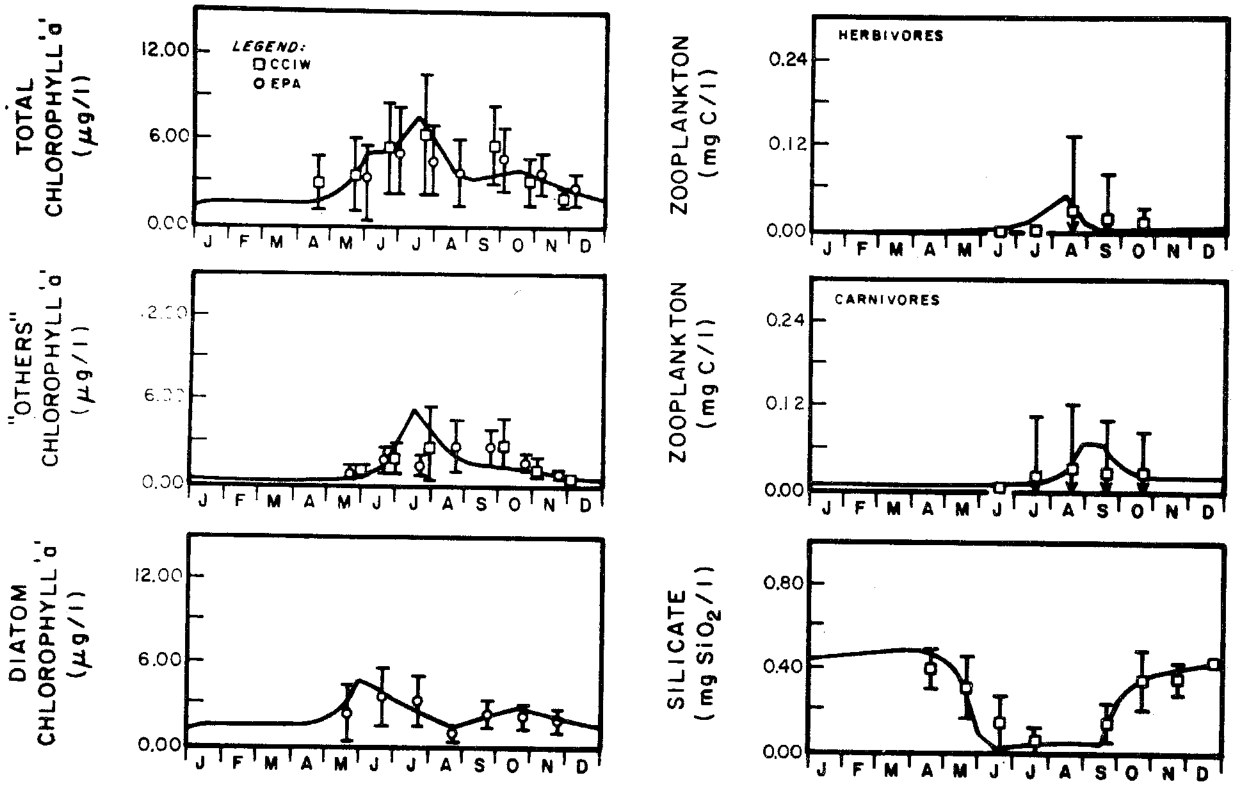


Figure 8. LAKE1A model calibration results (1972).

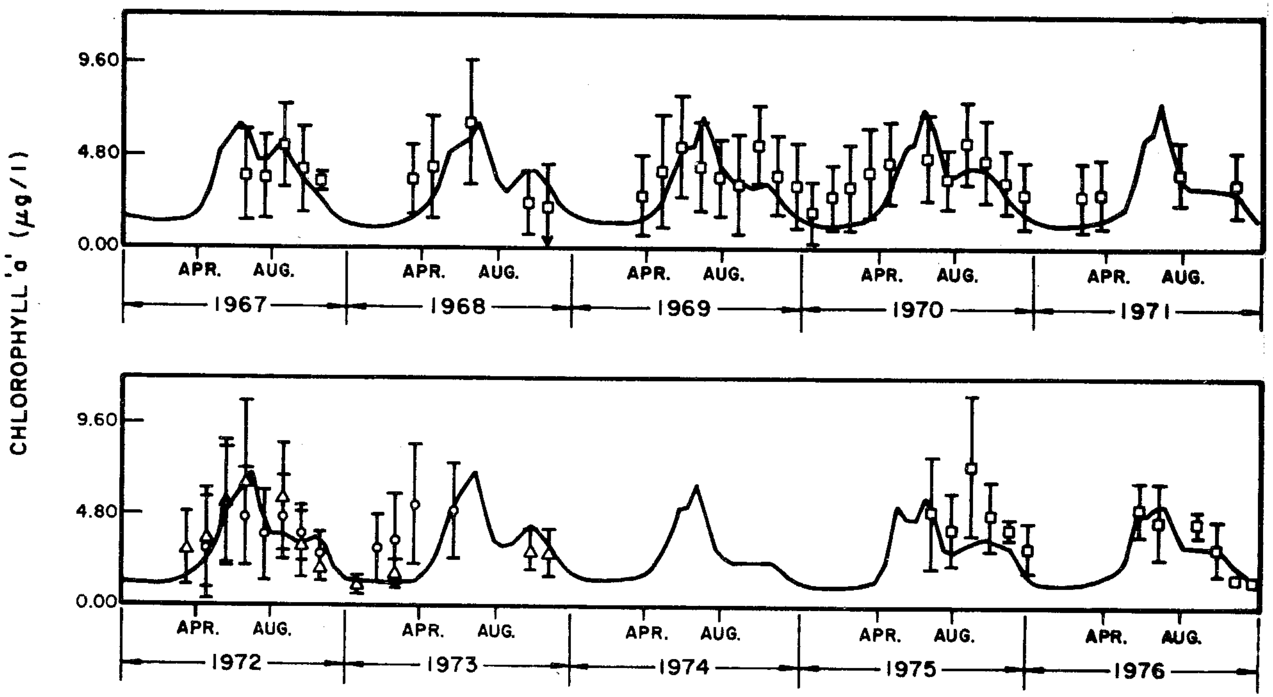


Figure 9. Long term chlorophyll 'a' calibration, epilimnion.

As noted, the output was saved and the MVP program was accessed to provide estimates of the verification status of the model. For chlorophyll, some results of the verification analyses are shown in Figure 10 where regression analyses and the relative error are shown across all years for the two kinetic systems. Statistical comparison results such as these provide the analyst with quantitative measures of model status. Figure 11 shows the median relative error for 1967-1976 for each variable and across all variables. For the latter, the results show that the inclusion of the more complicated LAKE1A kinetics did improve overall model status by about one-third. The overall median error of 22% represents a useful measure when discussing overall model credibility. Finally, MVP also produces verification scores (using a Student's "t" test) and some results across all years and variables are shown in Figure 12. At the given standard errors of the mean, the verification score for LAKE1A kinetics was 70%. In other words, for the months and variables where a comparison could be made between observed and computed values, there was no difference between observed and computed means 70% of the time at a 90% confidence interval. If the estimated standard errors are in question, then a new score is easily computed as shown in Figure 12 for values of 1/2 and 1-1/2 of the given standard errors. These results illustrate the utility of the WASP-MVP package - in this case for a long term, multi-variable model analysis.

WASP has also demonstrated its flexibility as a modeling tool in investigating the spread and impact of hazardous materials in receiving waters. A particular application was to determine the receiving water pH response to a spill of a strong acid (9). The model's kinetic framework was based upon the carbon dioxide-bicarbonate-carbonate equilibria (since this buffer system is the predominant buffering system in natural waters), and the exchange of carbon dioxide with the atmosphere under supersaturated conditions. As a check on the model kinetics (carbonate-bicarbonate buffering), a simulation of the laboratory titration of a sample volume of river water with a strong acid was performed. Figure 13 shows a good comparison between lab data and model results. Figures 14 and 15 show receiving water response to a 72 minute spill of a strong acid. Figure 14 also indicates the type of segment grid which one may use to model spills of hazardous materials, a fine mesh at the immediate point of discharge and an expanding mesh as one is further removed from the point of discharge.

WASP was also employed in a recent study (10) for the State of New York, Department of Environmental Conservation, concerning the need for remedial action and the impact of such action on achieving polychlorinated biphenyl (PCB) environmental objectives in the Hudson River ecosystem. A modeling framework was developed which defined the major interactions that lead to the PCB distribution in the biotic and abiotic sectors of a given aquatic environment (Figure 16). A simplifying assumption was made which permitted the decoupling of the biotic and abiotic sectors (i.e., if the food chain, above the phytoplankton, is viewed as a whole, then its uptake and excretion produce sinks and sources of PCB in the abiotic sector which are negligible when compared to those within the abiotic sector alone). WASP was used then to compute PCB distributions in the abiotic sector which were dependent upon advective and dispersive transport, sedimentation and resuspension, sediment release, evaporation, absorption and de-absorption kinetics and external

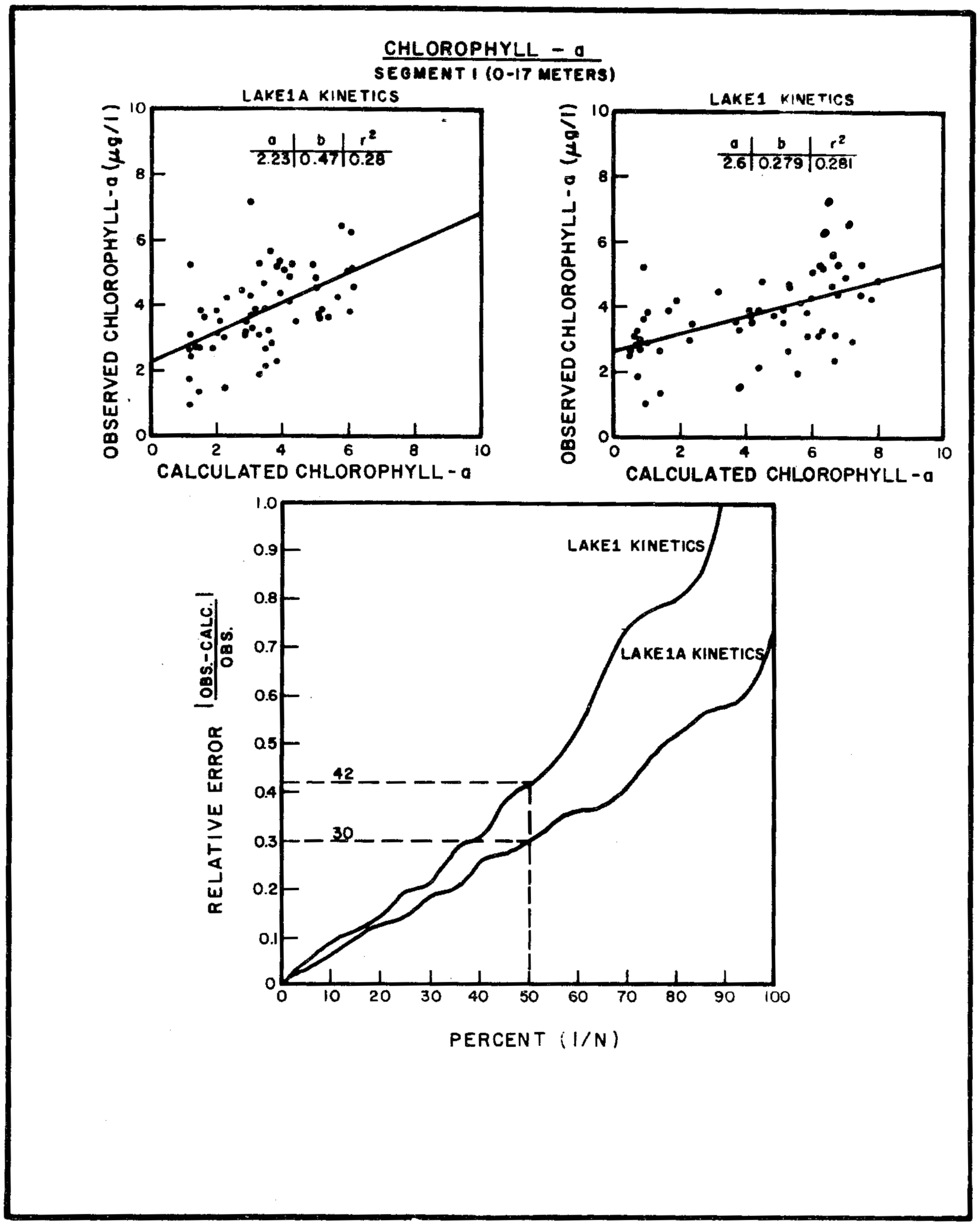


Figure 10. Regression analyses and relative error analyses of LAKE1 and LAKE1A model results versus field data (chlorophyll).

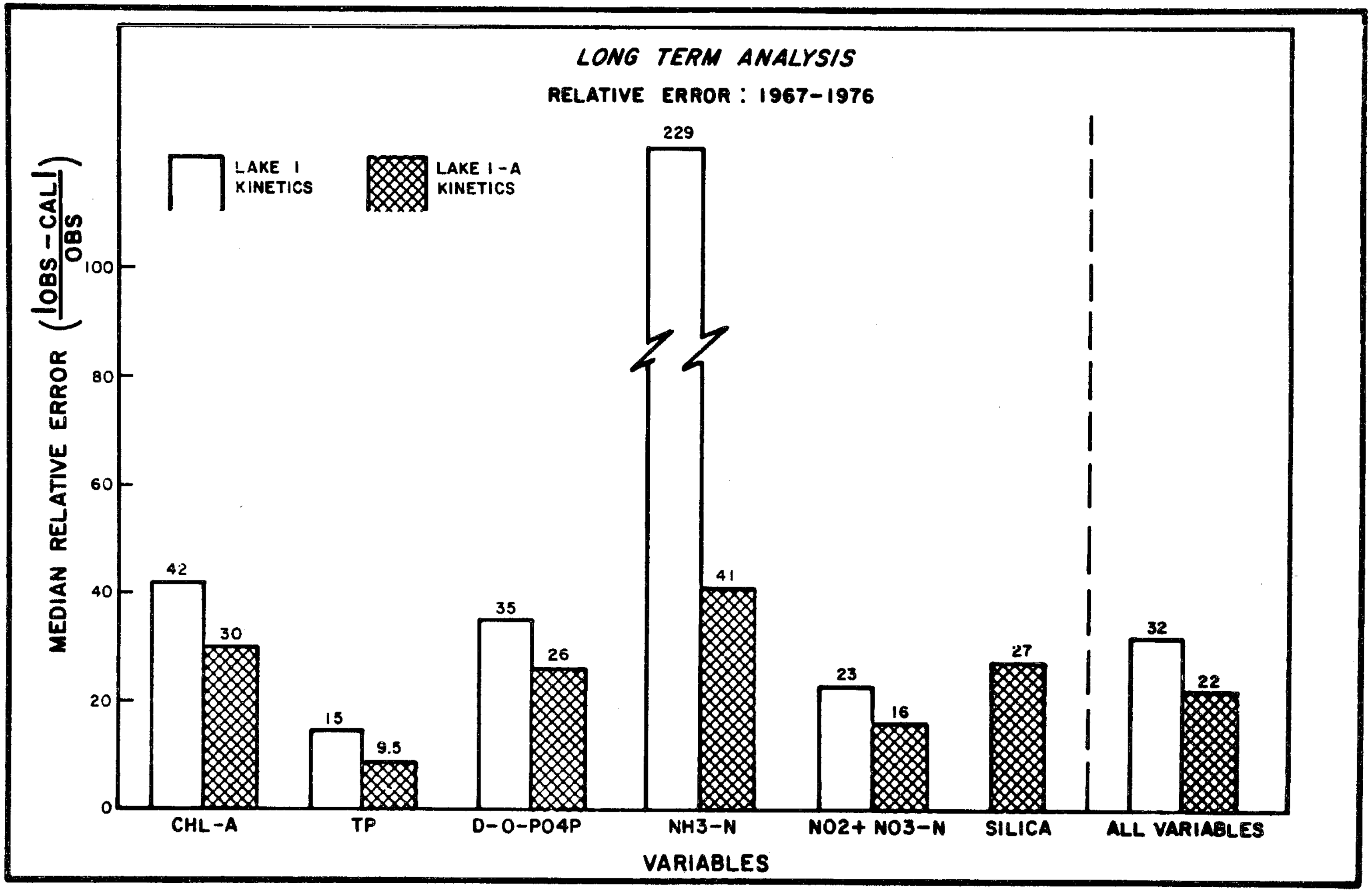


Figure 11. Median relative error analyses of LAKE1A model for several state-variables.

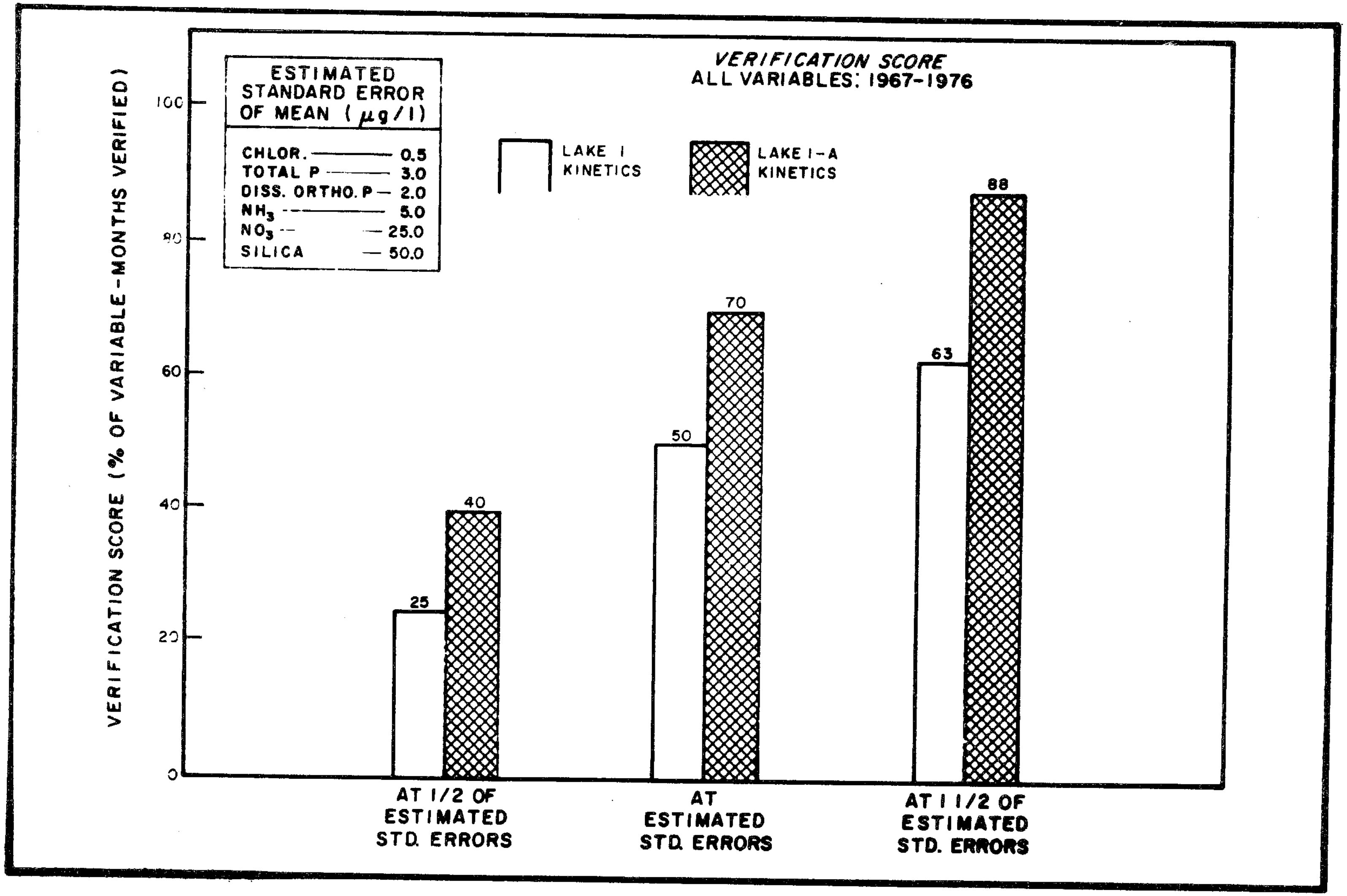


Figure 12. Student's "t" test analyses of LAKE1 and LAKE1A models.

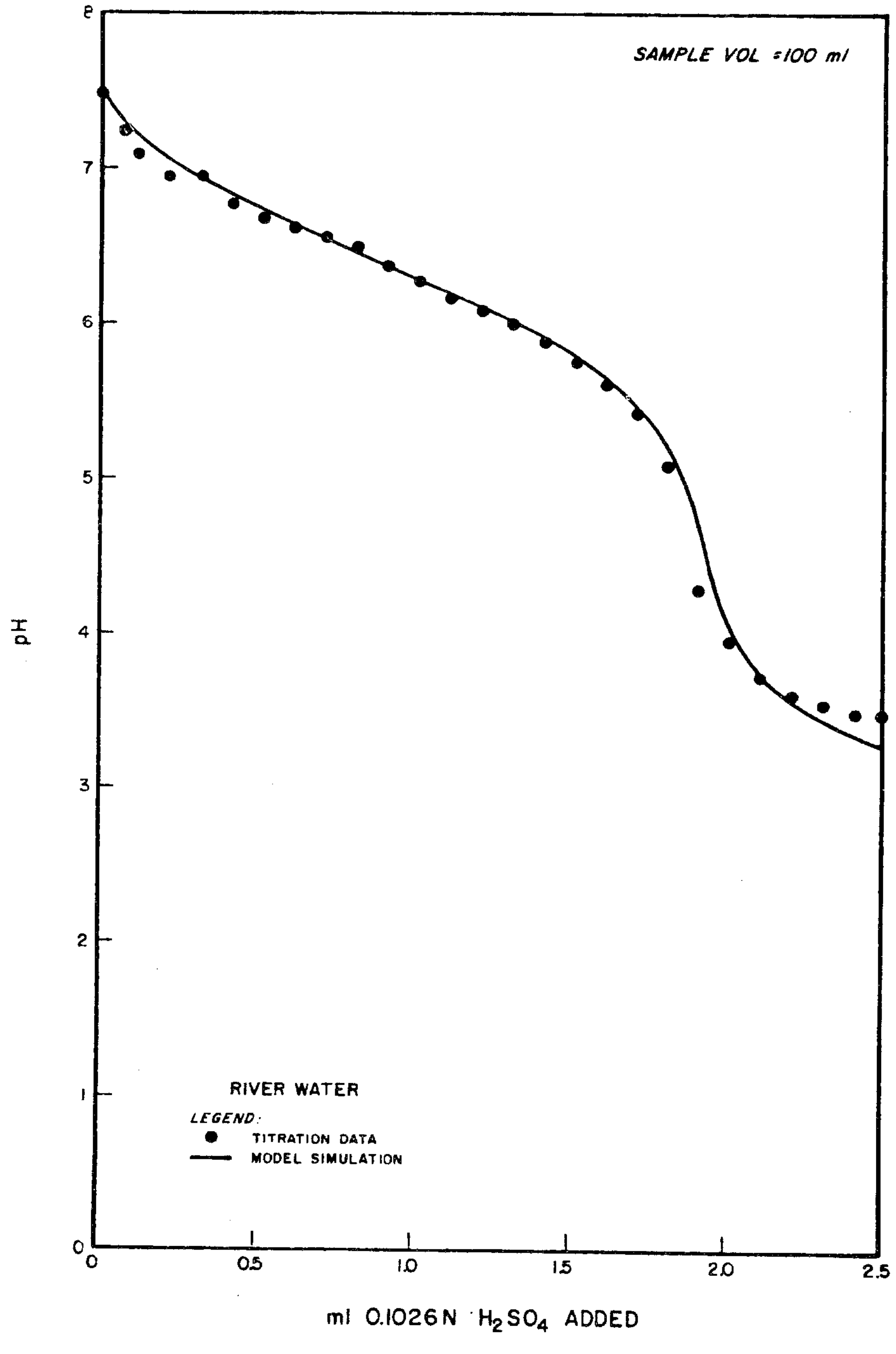


Figure 13. Acid spill problem lab data versus model output.

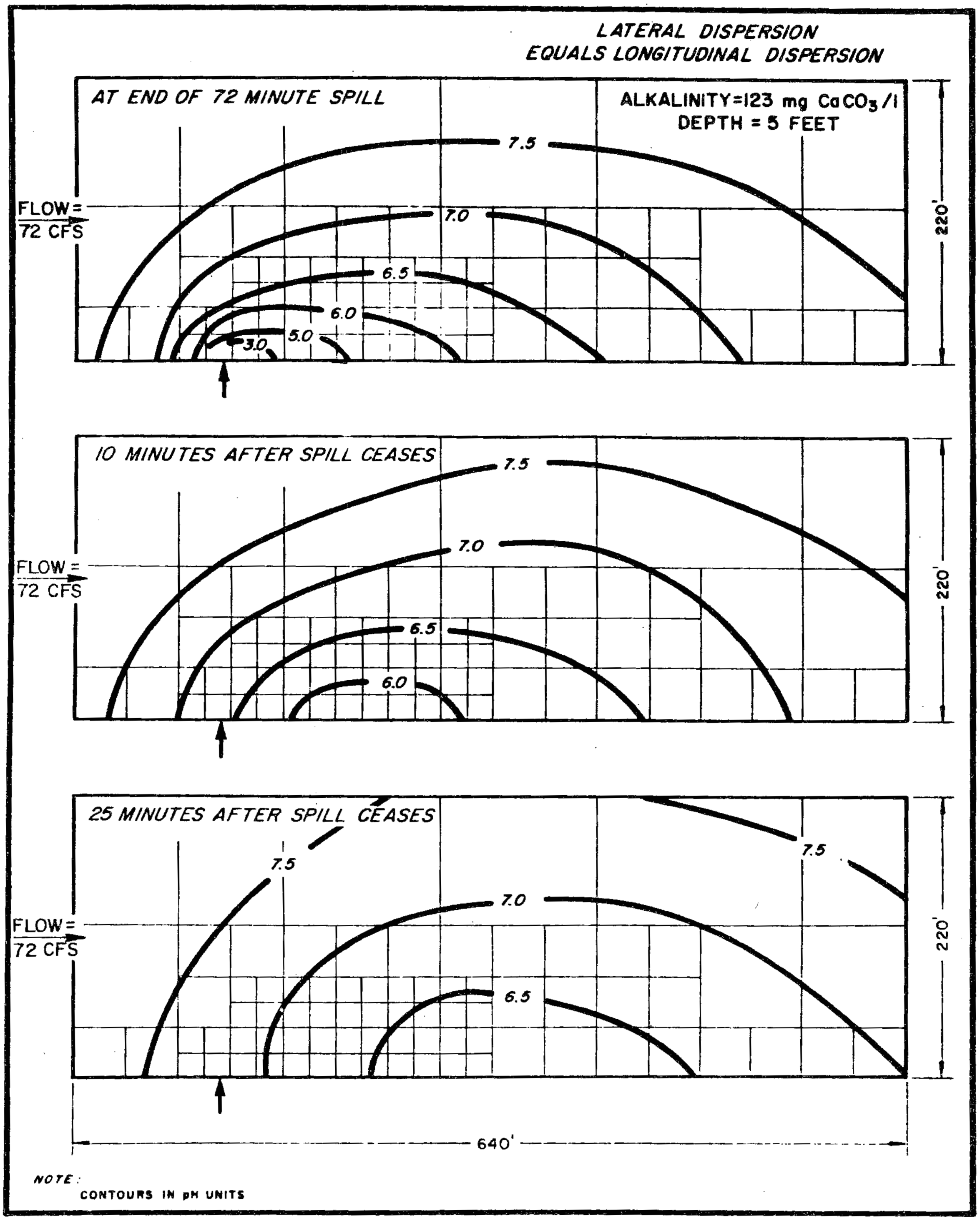


Figure 14. Spatial river pH response to a 72 minute spill of 96 percent sulfuric acid at 0.8 gpm.

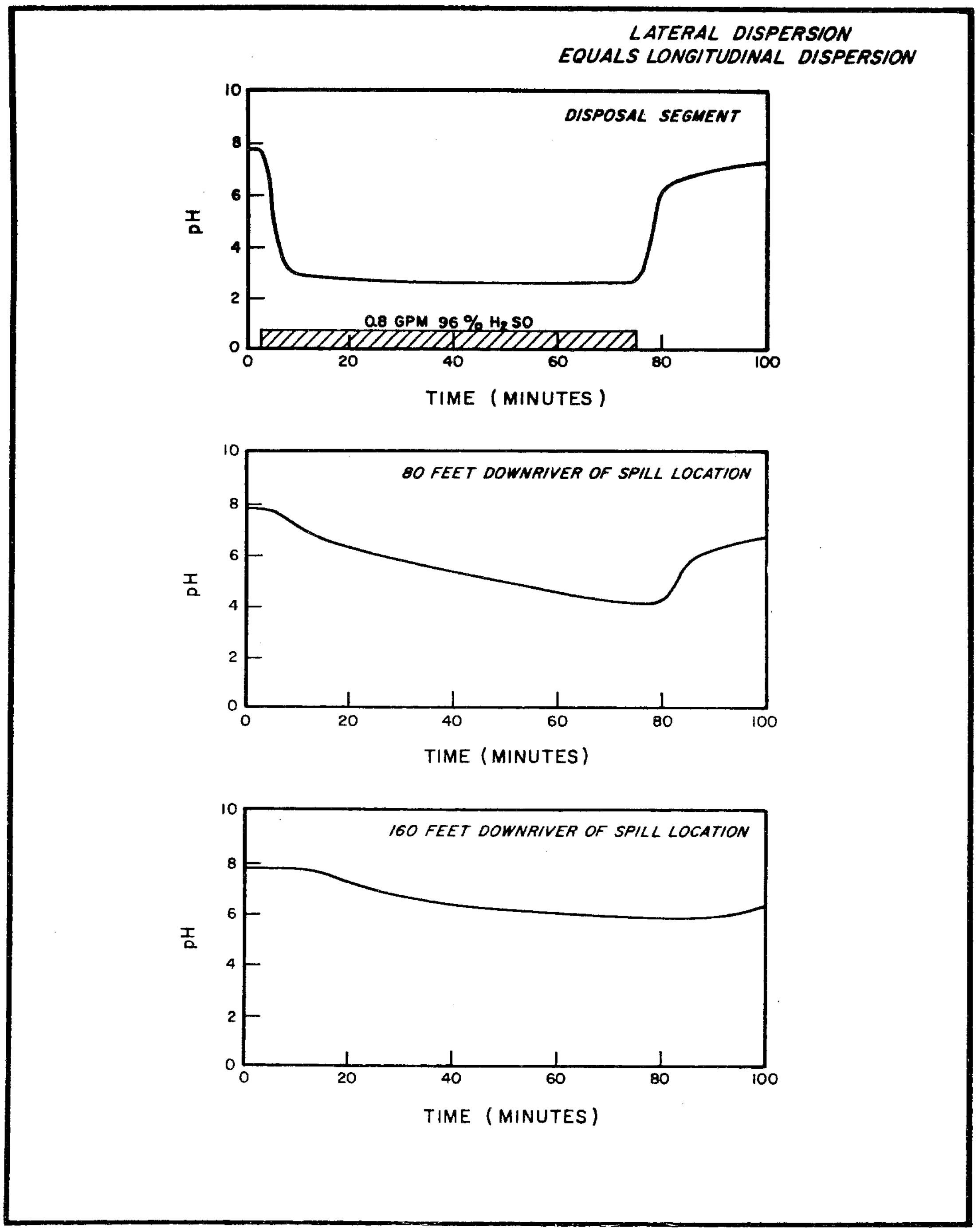


Figure 15. Temporal river pH response to a 72 minute spill of 96 percent sulfuric acid at 0.8 gpm.

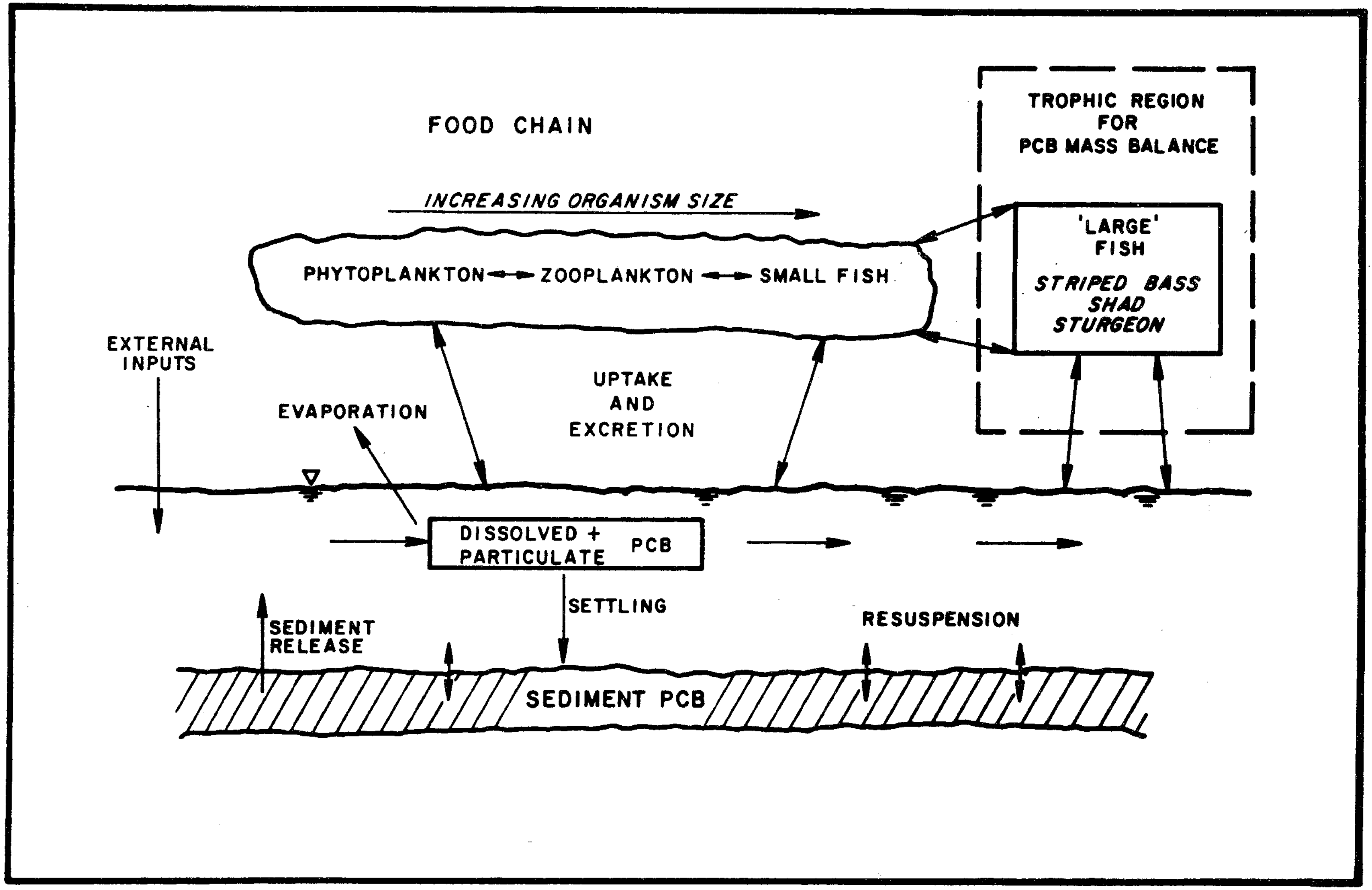


Figure 16. Schematic of major features in PCB distribution throughout food chain.

waste sources. Some model results for the upper Hudson river are presented in Figure 17.

Although not initially thought of as a tool for modeling biological treatment processes, WASP is capable of being programmed so as to provide this capability. WASP has been applied to a number of research efforts aimed at developing useful kinetic models for the anaerobic filter, rotating biological contactor and and the pure oxygen treatment system to name a few (11,12).

Even though WASP is essentially a stand-alone program, i.e., not executed in conjunction with another program (that might produce hydrodynamics or runoff waste loads), it can be user-programmed to interface with other computer programs if desired. In the recent New York City 208 Study (13) WASP was modified to accept time-variable non-point runoff flows and waste loads from a landslide rainfall runoff model. Figure 18 shows the general system logic and interfacing of the programs.

PROGRAM OVERVIEW

Before going into a detailed presentation of the WASP program and the theories upon which it is founded and numerically implemented, a brief discussion will be presented of the basic philosophy and assumptions underlying the program.

The key principle upon which the model equations of WASP are founded is the principle of conservation of mass. This principle simply states that the mass of each water quality constituent being investigated must be accounted for in one way or another. WASP, conserving mass both in time and space, accounts for and traces the water quality constituents from their point of spatial and temporal input to their final points of export.

In order to perform the spatial and temporal mass balance computations the user must supply WASP with input data defining the model segmentation, advective and dispersive transport fields, boundary conditions, forcing functions (waste loads), segment parameters, kinetic constants, time variable kinetic functions, and initial conditions for the state variables (water quality constituents). WASP utilizes this input data together with the user supplied kinetic subroutine to construct the mass balance equations, which are then numerically integrated in time. At a user specified time interval (print interval) WASP saves the current values of the state variables, and other user selected variables of interest, and stores them on auxiliary storage disk files for subsequent retrieval by the WASP graphics subroutine and the MVP program. A simplified program flow chart is presented in Figure 19.

MVP may be executed at the user's discretion. If the user should chose to perform MVP analyses he will need to supply field data for comparison to the theoretical computations generated by WASP. MVP uses three statistical tests for scoring the model verification. The scores are determined using a Student's "t" test on a comparison of the means of the theoretical and ob-

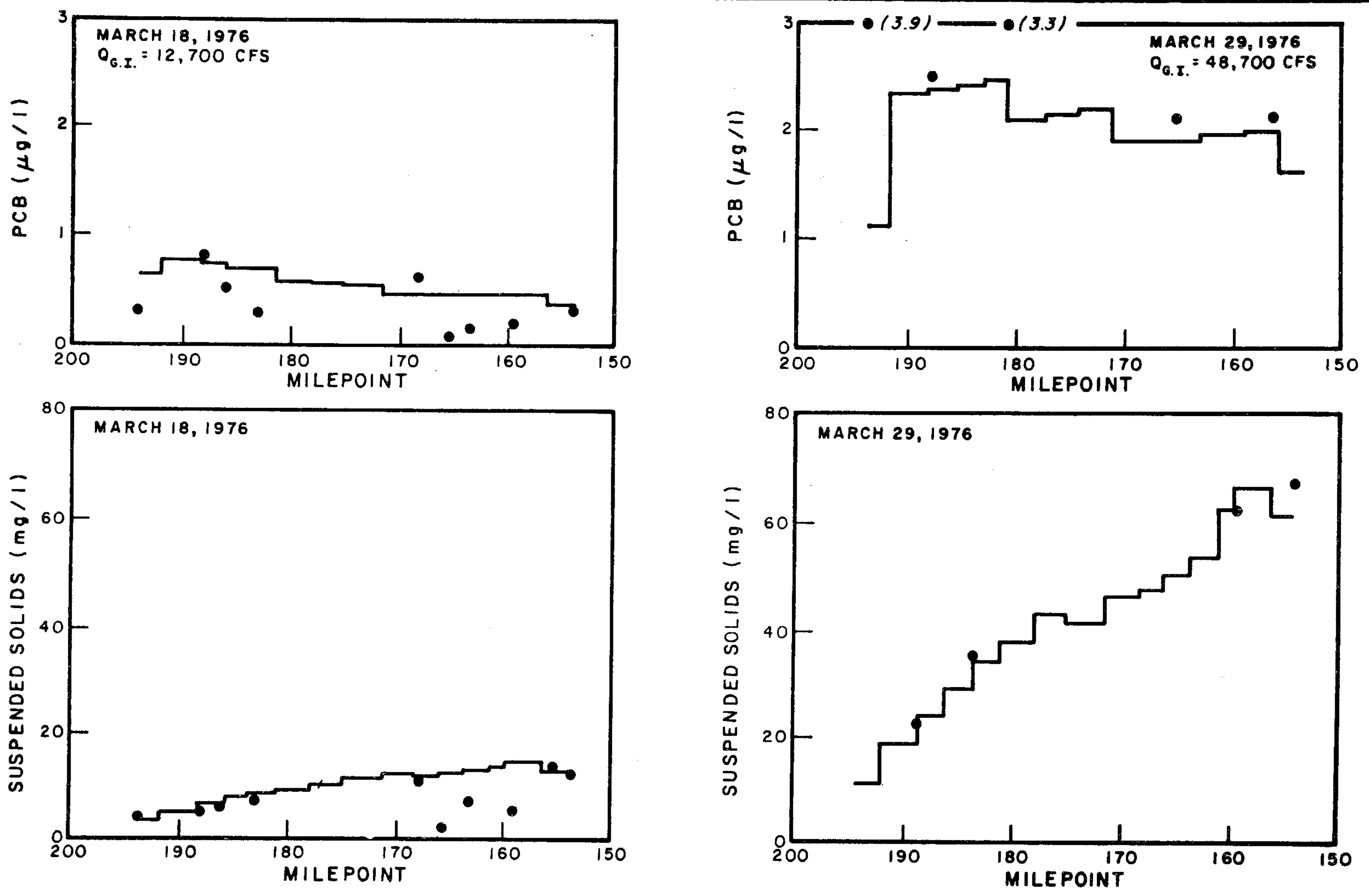


Figure 17. Total PCB and suspended solids calibration profiles (upper Hudson River).

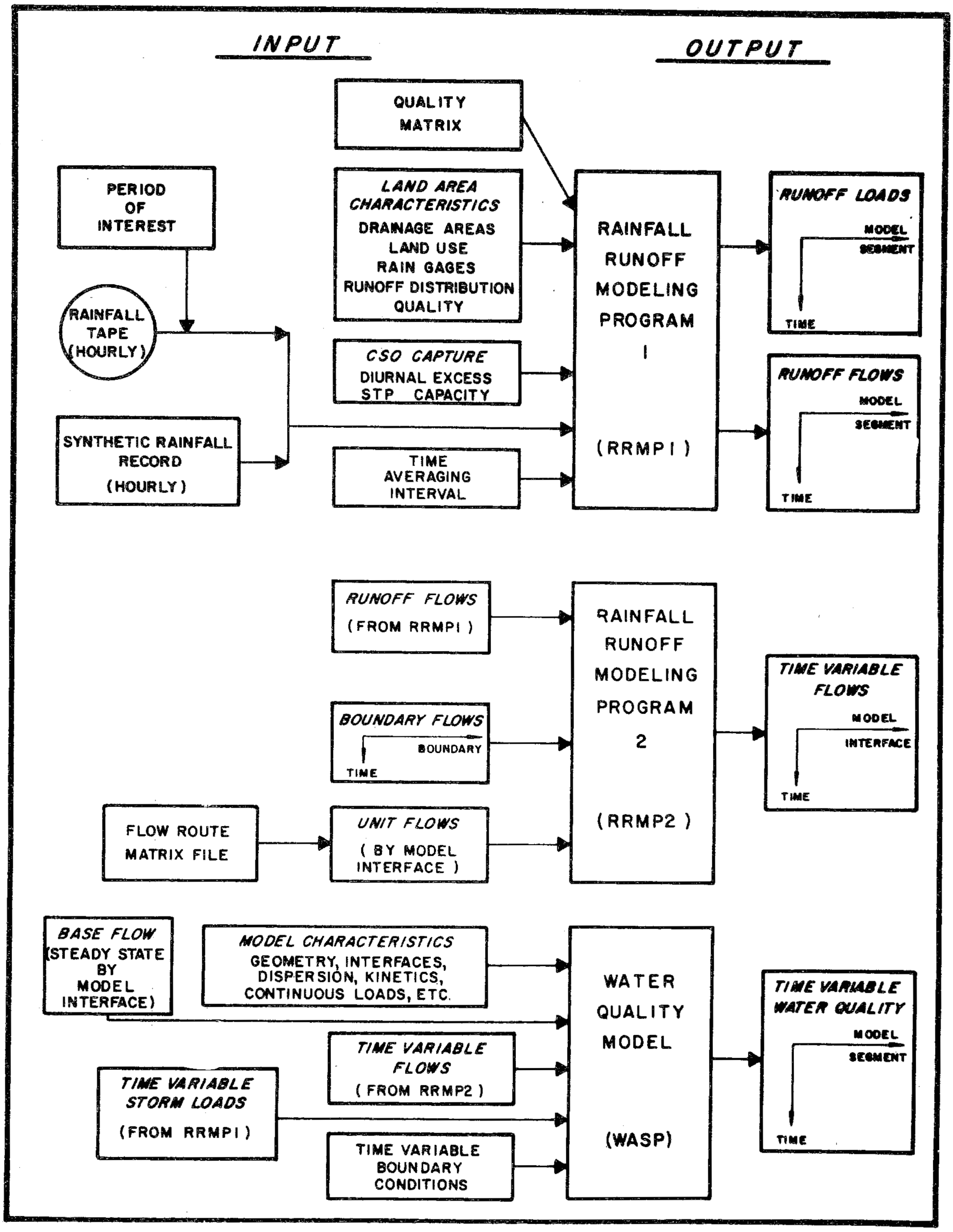

INPUT
OUTPUT
QUALITY MATRIX
PERIOD OF INTEREST
LAND AREA CHARACTERISTICS
DRAINAGE AREAS
LAND USE
RAIN GAGES
RUNOFF DISTRIBUTION
QUALITY
RAINFALL RUNOFF MODELING PROGRAM I
(RRMPI)
RUNOFF LOADS
MODEL SEGMENT
TIME
RAINFALL TAPE (HOURLY)
CSO CAPTURE
DIURNAL EXCESS
STP CAPACITY
RUNOFF FLOWS
MODEL SEGMENT
TIME
SYNTHETIC RAINFALL RECORD (HOURLY)
TIME AVERAGING INTERVAL
RUNOFF FLOWS (FROM RRMPI)
RAINFALL RUNOFF MODELING PROGRAM 2
(RRMP2)
BOUNDARY FLOWS
BOUNDARY
TIME
TIME VARIABLE FLOWS
MODEL INTERFACE
TIME
FLOW ROUTE MATRIX FILE
UNIT FLOWS (BY MODEL INTERFACE)
BASE FLOW (STEADY STATE BY MODEL INTERFACE)
MODEL CHARACTERISTICS
GEOMETRY, INTERFACES, DISPERSION, KINETICS, CONTINUOUS LOADS, ETC.
WATER QUALITY MODEL
(WASP)
TIME VARIABLE WATER QUALITY
MODEL SEGMENT
TIME
TIME VARIABLE FLOWS (FROM RRMP2)
TIME VARIABLE STORM LOADS (FROM RRMPI)
TIME VARIABLE BOUNDARY CONDITIONS

Figure 18. Land side simulator - WASP systems diagram.

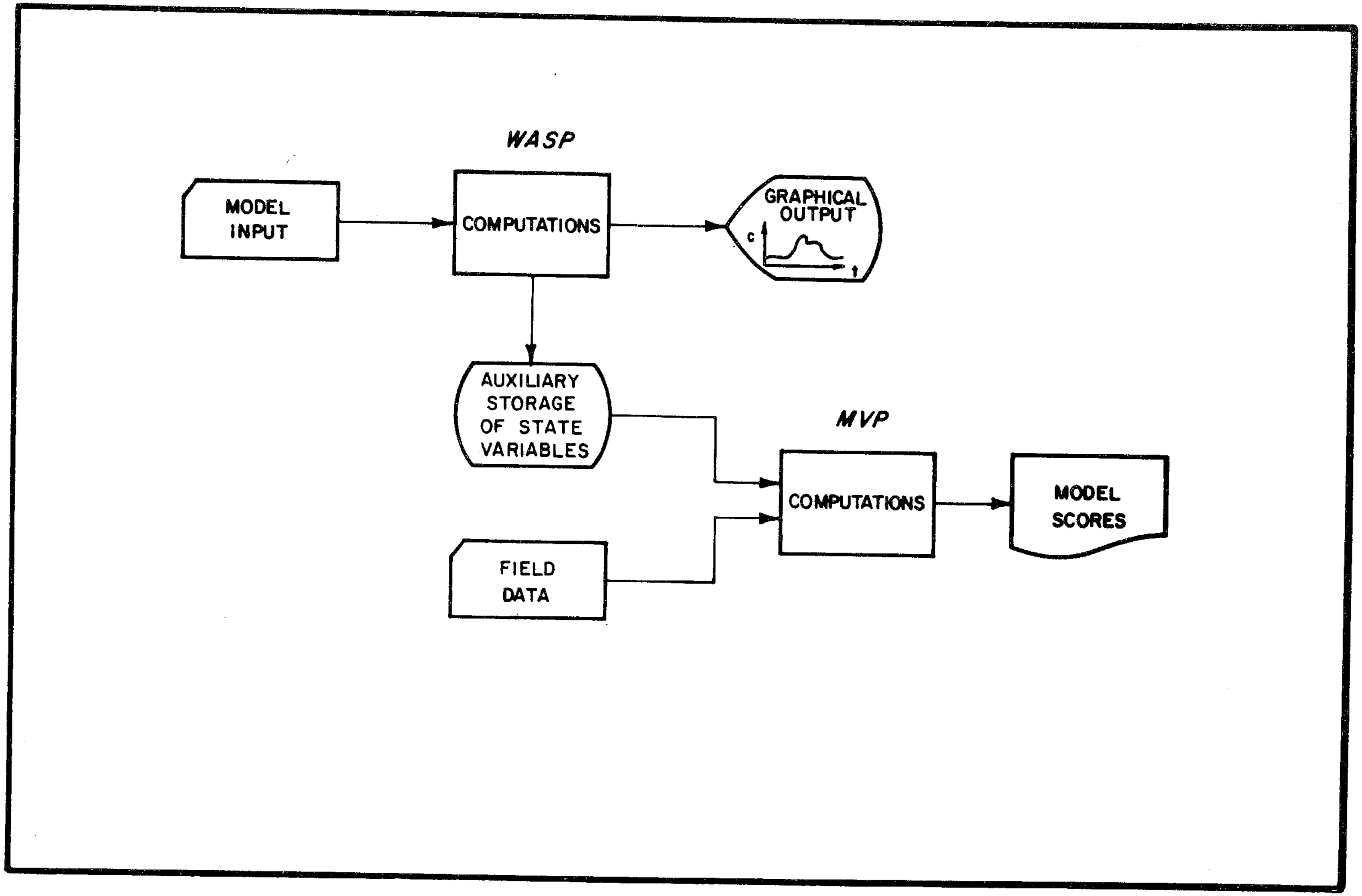


Figure 19. WASP/MVP program overview.

served data, a linear regression of theoretical and observed data, and a comparison of differences between theoretical and observed data. It should be noted, however, that even without extreme field data the MVP may prove useful in aggregating computed output according to any spatial or temporal averaging scheme.

Both WASP and MVP are written in FORTRAN IV, using a modular, subroutine orientated approach. This has permitted both programs to be available for a number of different computers with different core capacities. Table 2 contains a list of the computers on which WASP and MVP have been implemented.

TABLE 2. WASP AVAILABILITY

Computer	Core Capacity Words	Program Systems	Configuration Segments
DEC PDP 11/45 (EPA Grosse Ile)	32K	16	20
DEC PDP 11/70 (Manhattan College)	32K	12	40
DEC PDP 11/70 (EPA-Athens)	32K	16	29
IBM 370/168 (EPA-COMNET)[1]	256K	19	120
CDC 6600 (NYU)[1,2]	100K	23	20

[1] Completely core-resident (all other versions require program overlays).
[2] Modified version of WASP - input structure not compatible with the IBM, DSC, and DEC versions.

REFERENCES

1. Hydroscience. 1972. Mathematical Model of Phytoplankton Dynamics in the Sacramento-San Joaquin Bay Delta, Preliminary report. Prepared for the California Department of Water Resources.

2. Hydroscience. 1974. Western Delta and Suisun Bay Phytoplankton Model - Verification and Projections. Prepared for the California Department of Water Resources.

3. Thomann, R.V., <u>et al</u>. 1975. Mathematical Modeling of Phytoplankton in Lake Ontario, 1. Model Development and Verification. EPA-660/3-75-005, NERC, ORD, EPA, Corvallis, Oregon. 177 pp.

4. Thomann, R.V., <u>et al</u>. 1975. Mathematical Modeling of Phytoplankton in Lake Ontario, 2. Simulations Using LAKE 1 Model. EPA-660/3-75-005, NERC, ORD, EPA, Corvallis, Oregon.

5. DiToro, D.M., <u>et al</u>. Lake Huron report in review.

6. DiToro, D.M., _et al_. Lake Erie report in review.

7. Thomann, R.V., _et al_. 1978. Verification Analysis of Lake Ontario and Rochester Embayment Three-Dimensional Eutrophication Models. In review.

8. Thomann, R.V., _et al_. 1978. A Ten Year Modeling Analysis of Phytoplankton in Lake Ontario. In preparation.

9. Nusser, J.A., _et al_. 1978. Mathematical Modeling for Impact and Control of Hazardous Spills. Proceedings of the 1978 National Conference on Control of Hazardous Material Spills, sponsored by the U.S. EPA and the U.S. Coast Guard, Miami Beach, Florida, April 1978.

10. Hydroscience. 1978. Estimation of PCB Reduction by Remedial Action on the Hudson River. Prepared for the State of New York, Department of Environmental Conservation.

11. Mueller, J.A., _et al_. 1977. Application of Carbonate Equilibrium to High Purity Oxygen and Anaerobic Filter Systems. Presented at the 173rd National Meeting of the American Chemical Society, New Orleans, Louisiana.

12. Mueller, J.A., _et al_. 1978. Nitrification in Rotating Biological Contactors. Presented at the 51st Annual Conference of the Water Pollution Control Federation, Anaheim, California.

13. Hydroscience. 1978. Task Report 334, Time Variable Modeling, submitted to Hazen and Sawyer, Engineers for NYC 208.

SECTION 4

WASP THEORY

MASS BALANCE EQUATIONS

The basic concept in writing a mass balance equation for a body of water is to account for all of the material entering and leaving the water body via direct addition of material (runoff and loads), via advective and dispersive transport mechanisms, and via physical, chemical, and biological transformations. In order to formulate a mass balance equation for a specific water quality variable, consider a coordinate system as shown in Figure 20a, where the x- and y-coordinates are in the horizontal plane and the z-coordinate is in the vertical plane. Several authors (1,2,3) have stated the proper form of the mass balance equation around an infinitesimally small fluid volume (Figure 20b) to be

$$\frac{\partial c}{\partial t} = \frac{\partial}{\partial x} (E_x \frac{\partial c}{\partial x}) + \frac{\partial}{\partial y} (E_y \frac{\partial c}{\partial y}) + \frac{\partial}{\partial z} (E_z \frac{\partial c}{\partial x}) - \frac{\partial}{\partial x} U_x c$$

$$- \frac{\partial}{\partial y} U_y c - \frac{\partial}{\partial z} U_z c \pm S (x,y,z,t) \qquad (2.1)$$

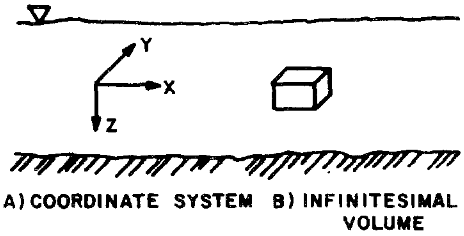


Figure 20. Coordinate systems.

where: c = concentration of the water quality variable $[M/L^3]$

t = time [T]

S = vector of all other sources and sinks of the water quality variable

E = diagonal matrix of dispersive coefficients $[L^2/T]$

U = vector of velocities [L/T]

The dispersion coefficient represents the overall phenomenon of mixing due to the temporal variation of tidal velocity, the lateral and vertical gradients in velocity and the density differences within the water body.

For the sake of clarity and brevity the derivation of finite-difference form of the mass balance equation, to be presented in the following section, will be for a one-dimensional rectangular estuary. Under the assumptions of vertical and lateral homogenuity, and permitting variation, or "a gradient", along only the length of the estuary, the mass balance equation for the one-dimensional case is

$$\frac{\partial c}{\partial t} = \frac{1}{A}\frac{\partial^2(E_x Ac)}{\partial x^2} - \frac{1}{A}\frac{\partial U_x Ac}{\partial x} \pm S\ (x,t) \qquad (2.2)$$

FINITE DIFFERENCES

The fundamental method of solution employed in WASP is the use of finite-difference approximations to the derivatives of Equations (2.1) and (2.2). In order to keep this section from becoming overly long, the development of the numerical procedures will of necessity be abbreviated, with the occasional use of "it can be shown". However, if the reader is interested in a more detailed explanation of finite-difference methods, the author recommends an excellent presentation by Smith (4).

Taylor's series expansion theorem states that for a function u, where its derivatives are single-valued, finite, and continuous functions of x, then

$$u(x+h) = u(x) + hu'(x) + \frac{1}{2}h^2u''(x) + \frac{1}{6}h^3u'''(x) + \ldots \qquad (2.3)$$

$$u(x-h) = u(x) - hu'(x) + \frac{1}{2}h^2u''(x) + \frac{1}{6}h^3u'''(x) + \ldots \qquad (2.4)$$

Assuming that terms containing the third and higher powers of h are negligible in comparison to the lower powers of h, then Equations (2.3) and (2.4) may be subtracted and added together and rearranged to produce Equations (2.5) and (2.6) respectively,

$$u'(x) = (\frac{du}{dx})_{x=x} \simeq \frac{1}{2h} \{u(x+h) - u(x-h)\} \qquad (2.5)$$

$$u''(c) = (\frac{d^2u}{dx^2})_{x=x} \simeq \frac{1}{h^2} \{u(x+h) - 2u(x) + u(x-h)\} \qquad (2.6)$$

with an error terms of order h^2. Referring to Figure 21, Equation (2.5) can be seen as an approximation to the slope of the tangent centered at P formed by chord AB, and is known as the central-difference approximatior

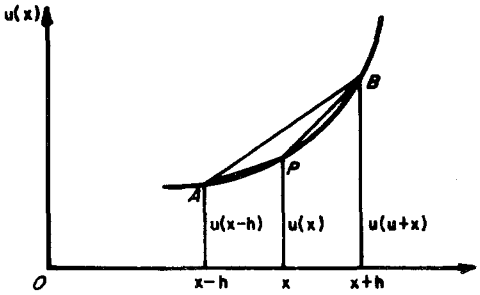


Figure 21. Finite-difference approximations.

The slope of the tangent at P may also be approximated by the slope of the chord PB giving the forward difference formula,

$$u'(x) \simeq \frac{1}{h} \{u(x+h) - u(x)\} \qquad (2.7)$$

or the slope of the chord AP, giving the backward-differences formula,

$$u'(x) \simeq \frac{1}{h} \{u(x) - u(x-h)\} \qquad (2.8)$$

Both Equations (2.7) and (2.8) may be obtained from (2.3) and (2.4) respectively, by assuming the second and higher order powers of h are negligible. The error term for both the forward-difference and the backward-difference is in order h. To ensure positive stable solutions WASP employs a backward-difference approximation in the spatial plane (5). For programming simplicity WASP employs a forward-difference approximation for the temporal plane.

Formulate a grid, as shown in Figure 22, that subdivides the x-t plane into sets of equal rectangles with sides equal to Δx and Δt, and let the coordinates (x,t) of a representative of point P be

$$x = i\Delta x \quad \text{and} \quad t = n\Delta t,$$

where i and n are integers. Use the following notation to denote the value of u at P

$$u_P = u(i\Delta x,\ n\Delta t) = u_{i,n}$$

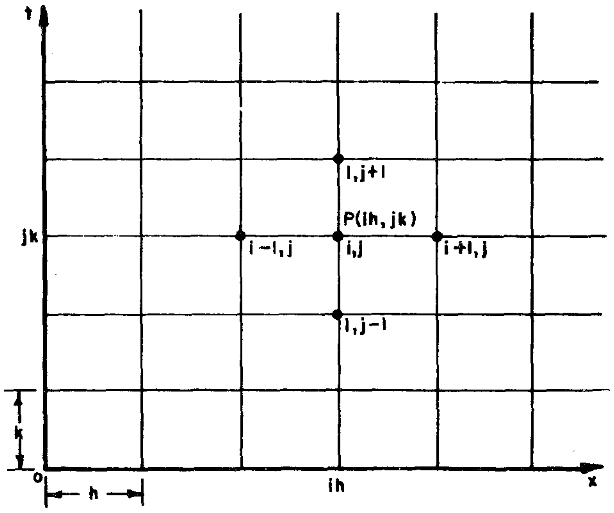


Figure 22. Finite-difference grid.

Then by Equation (2.6)

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_P = \left(\frac{\partial^2 u}{\partial x^2}\right)_{i,n} \simeq \frac{u_{i+1,n}-2u_{i,n}-u_{i-1,n}}{\Delta x^2} \qquad (2.9)$$

Similarly, for the backward-difference approximation Equation (2.8)

$$\left(\frac{\partial u}{\partial x}\right)_P = \left(\frac{\partial u}{\partial x}\right)_{i,n} \simeq \frac{u_{i,n}-u_{i-1,n}}{\Delta x} \qquad (2.10)$$

and for a forward-difference approximation time equation

$$\left(\frac{\partial u}{\partial t}\right)_P = \left(\frac{\partial u}{\partial t}\right)_{i,n} \approx \frac{u_{i,n+1}-u_{i,n}}{\Delta t} \qquad (2.11)$$

Restating Equation (2.2)

$$\frac{\partial c}{\partial t} = \frac{1}{A}\,\frac{\partial^2(E_x Ac)}{\partial x^2} - \frac{1}{A}\,\frac{\partial Q_x c}{\partial x} \pm S \qquad (2.12)$$

where $Q_x = U_x A$, and

A = the cross-sectional area

and using Equations (2.9) and (2.10) as finite-difference approximations for the first and second terms of the right side of the equation respectively, one may develop a finite difference approximation at the mesh point (i,n) as follows:

$$\frac{dc_i}{dt} = -\frac{1}{A}\left[\frac{(Qc)_{i,n}-(Qc)_{i-1,n}}{\Delta x}\right]$$

$$+\frac{1}{A}\left[\frac{(EAc)_{i+1,n}-2(EAc)_{i,n}+(EAc)_{i-1,n}}{\Delta x^2}\right] \pm S \qquad (2.13)$$

Letting V = AΔx and re-arranging terms Equation (2.13) may be expressed

$$V_i\,\frac{dc_i}{dt} = (Qc)_{i-1,n}-(Qc)_{i,n}$$

$$+\frac{(EAc)_{i+1,n}-2(EAc)_{i,n}+(EAc)_{i-1,n}}{\Delta x} \pm S \qquad (2.14)$$

Dividing the water body into completely mixed finite segments as pictured in Figure 23, and recalling that S represents the source-sink terms both for the direct addition of the water quality constituent (point source discharge, distributed runoff, etc.) and addition-removal through reactive processes, one may see that Equation (2.14) represents a mass balance for segment i, and may be restated for a fixed point in time, as follows:

$$V_i\,\frac{dc_i}{dt} = Q_{i-1,i}\,c_{i-1} - Q_{i,i+1}\,c_i + \frac{E_{i-1,i}A_{i-1,i}}{L_{i-1,i}}\,(c_{i-1}-c_i)$$

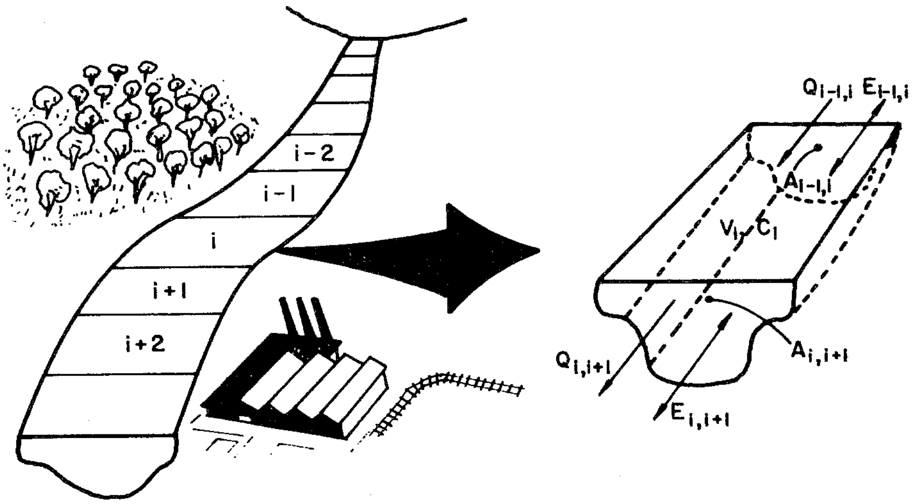


Figure 23. Completely mixed finite segments.

$$+ \frac{E_{i,i+1}A_{i,i+1}}{L_{i,i+1}} (c_{i+1} - c_i) \pm W_i \pm K_i V_i c_i \qquad (2.15)$$

where

c_i = segment concentration, M/L^3

V_i = segment volume, L^3

$Q_{i-1,i}$ = net advective flow, from segment i-1 to segment i, L^3/T

$E_{i-1,i}$ = dispersion coefficient for the i-1, i interface, L^2/T

$A_{i-1,i}$ = cross-sectional area of the i-1,i interface, L^2

$L_{i-1,i}$ = characteristic length or mixing length of segment i-1 and i, L.

$$L_{i-1,i} = \frac{L_{i-1} + L_i}{2}$$

W_i = point, or distributed sources-sinks of the water quality constituent, M/T

K_i = kinetic or reactive process rate, 1/T.

It can be shown (5) that Equation (2.15) can be extended to develop the multi-dimensional, multi-constituent form of the equation as stated by Equation (2.16)

$$V_i \frac{dc_i^m}{dt} = \sum_j Q_{j,i} c_j^m - \sum_k Q_{i,k} c_i^m + \sum_j \frac{E_{i,j}A_{i,j}}{L_{i,j}} (c_j^m - c_i^m)$$

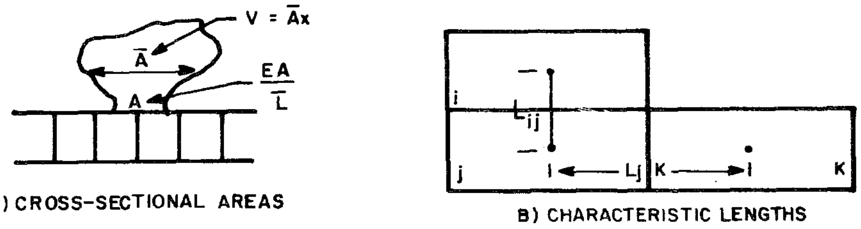
$$\pm W_i^m \pm K_i V_i c_i^m \pm K_i^{lm} V_i c_i^l \qquad (2.16)$$

where

K_i^{lm} = the cross-coupling reaction term

c_i^l = the concentration of state variable l in segment i

It should be noted that for the one-dimensinal and multi-dimensional cases, the advective and dispersive fields are assumed known. It should also be noted that for irregularly shaped water bodies (such as embayments) that $V = \bar{A}\Delta x$, where $\bar{A}$ is some representative cross-sectional area, as shown in Figure 24a. Also note that for multi-dimensional water bodies a segment may have different characteristic lengths for adjoining segments, see Figure 24b.



A) CROSS-SECTIONAL AREAS

B) CHARACTERISTIC LENGTHS

Figure 24. Cross-sectional areas and characteristic lengths.

If we divide Equation (2.16) through by V_i and permit the new right-hand side to equal $\dot{c}_i^n$, and then use the forward-difference approximation for the time derivative, we can form the following expression

$$(c_i^m)^{n+1} = (c_i^m)^n + (\dot{c}_i^m)^n \, \Delta t. \tag{2.17}$$

which states that the concentration at time n+1 is equal to the concentration at time n plus the derivative evaluated in time n times the time step, Δt.

STABILITY AND NUMERICAL ERROR

Classical stability analysis (Smith) requires $\Delta t/\Delta x^2 \leq 1/2$ to guarantee stable solutions for (2.17). However, in the classical analysis Δt and Δx are normalized variables and it is difficult to extend these variables to practicable applications. It has been shown (6) that for computational stability a necessary condition is that

$$\Delta t \leq \text{Min} \left(\frac{V_i}{\sum_j Q_{i,j} + \sum_j R_{i,j} + K_i V_i} \right) \tag{2.18}$$

where

$$R_{i,j} = \text{exchange coefficients} = \frac{E_{ij} \; A_{ij}}{L_{i,j}} \tag{2.19}$$

However, since K_i is, for many applications, non-linear and time-dependent itself, and therefore may be difficult to evaluate, the following criteria may be used for choosing the integration step-size

$$\Delta t \leq \text{Min} \left(\frac{V_i}{\sum_j Q_{ij} + \sum_j R_{ij}}\right) \tag{2.20}$$

providing that

$$\text{Max } (K_i V_i) << \sum_j Q_{ij} + \sum_j R_{ij}.$$

It should be recognized that the use of the completely mixed finite segment approximation in conjunction with the backward difference spatial approximation, introduces a numerical error (sometimes referred to as numerical or pseudo-dispersion in the literature) into the model (7). The extent of this effect is given by

$$E_{num} = \frac{u\Delta x}{2} \tag{2.21}$$

where E_{num} is the numerical error expressed as a pseudo-dispersion coefficient. For some applications, especially intra-tidal models, where u may be large, this error term may lead to highly distorted results (i.e., artificially spread spatial profiles). However, for many applications, especially in well mixed estuaries, where the time scale of importance is on the order of days to seasons rather than hours, and where u is the net advective freshwater velocity (usually small), the effect of the numerical error, E_{num}, is generally not significant. It should be noted that one cannot reduce or eliminate the numerical or pseudo-dispersion coefficient, E_{num}, by adjusting E, the true dispersion coefficient. The only way to reduce the effect of E_{num} in WASP is to reduce the segment length in the direction of u. However, one must also be aware that arbitrary reduction in the segment length effects the integration step size since

$$\Delta t \leq \text{Min} \left(\frac{V}{\Sigma Q + \Sigma R + KV}\right)$$

or

$$\Delta t \leq \text{Min} \left(\frac{A\ x}{\Sigma Q + \Sigma R + KV}\right)$$

and a reduction of Δx requires a like reduction of Δt. Choice of proper spatial and temporal grid sizes is still somewhat of an art and consideration of computer core size and execution speed, the nature of the problem being analyzed, and the degree of accuracy required, all influence the choice.

COMPUTATIONAL PROCEDURE

In order to integrate the aforementioned finite difference equations WASP uses a second order Runge-Kutta or predictor-corrector method (8). The operational procedure for the Runge-Kutta method may be best explained schematically via Figure 25. First, using (2.17), $c_{n+1/2}$ is computed for the half-step inteval $\Delta t/2$ (predictor). Using $c_{n+1/2}$, $t_{n+1/2}$ (i.e., $t_n + \Delta t/2$), the derivative $\dot{c}_{n+1/2}$ is evaluated, and this derivative is used as the average derivative for proceeding over the whole interval Δt (corrector). Experience has shown that for the general class of applications that WASP is used for, the second order Runge-Kutta method yields results comparable to the more commonly used fourth order Runge-Kutta method at an execution time savings of fifty-percent.

WASP uses a slightly modified version of the Runge-Kutta method in that WASP, in normal operations, prohibits any segment concentration from going negative and causing either numerical instability or numerical oscillations of the solution. In some applications, particularly in the nutrient limited growth phase of eutrophication modeling, it is possible for a particular segment derivative-timestep combination to cause a negative concentration. This negative concentration, if permitted to occur, might degenerate the true solution by causing either instability or oscillations. Rather than permit this to happen, WASP, upon detection of a derivative-timestep combination which would cause a negative segment concentration, maintains a positive segment concentration by setting the segment concentration projected for timestep n+1 to half the concentration that was present at timestep n. It should be noted that this procedure does not maintain a proper mass-balance, i.e., mass is not conserved. However, experience has shown this procedure to be acceptable within reason. WASP informs the user if and when a half-concentration procedure was performed, and the user can monitor the frequency of occurrence of the procedure and determine if the simulation must be rerun at a smaller timestep.

REFERENCES

1. Pritchard, D.W. 1958. The Equations of Mass Continuity and Salt Continuity in Estuaries. Journal of Marine Research, Vol. 17 (Nov. 1958), pp. 412-423.

2. Daily, J.W. and D.R.F. Harleman. 1966. Fluid Dynamics. Addison-Wesley Publishing Company, Reading, Mass., Chapter 16.

3. O'Connor, D.J. and Thomann, R.V. 1971. Estuarine Modeling: An Assessment, Chapter III EPA, 16070 DZV 02/71.

4. Smith, G.D. 1965. Numerical Solution of Partial Differential Equations, Oxford University Press, Ely House, London, Great Britian.

5. Thomann, R.V. 1972. Systems Analysis and Water Quality Management, Environmental Science Services Division, Environmental Research and Applications, Inc., New York, New York.

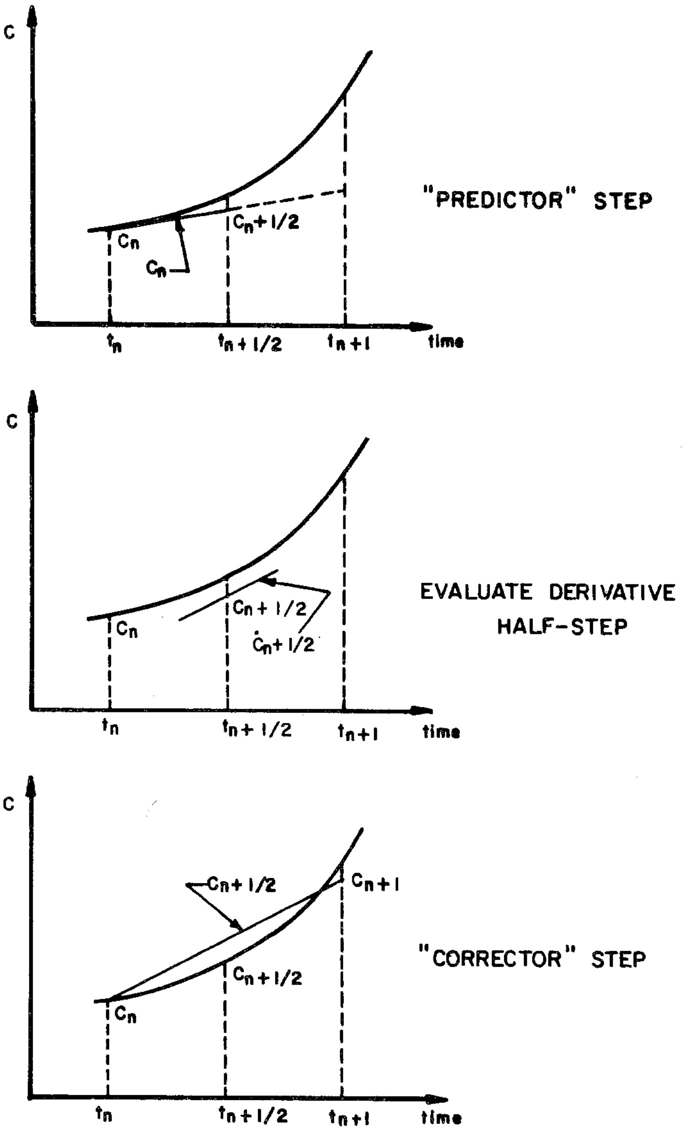


Figure 25. Second-order Runge-Kutta procedure.

6. Kent, R. 1960. Diffusion in a Sectionally Homogenous Estuary. Trans ASCE, Jour. SED, Vol. 86, No SA2, Mar. 1960, pp. 15-47.

7. Bella, D.A. and W.E. Dobbins. 1968. Difference Modeling of Stream Pollution. Proc. ASCE, 94, No. SA5, Oct. 1968, pp. 995-1016.

8. Carnahan, B., et al. 1969. Applied Numerical Methods, John Wiley and Sons, Inc., New York, New York, pp. 361-366.

SECTION 5

WASP PROGRAM LOGIC

INTRODUCTION

WASP was designed and written so as to permit its application to as wide a variety of water quality modeling problems as possible without a user being required to make extensive program revisions for each new application. In order to achieve this goal four objectives were set down and met:

1. FORTRAN IV would be chosen as the programming language due to its universality.

2. WASP would be written using a modular subroutine orientated approach for both program clarity and the ability of most computer operating systems to accomodate subroutine overlay structures where core requirements are restrictive.

3. WASP would permit the user a great deal of flexibility in structuring the physical setup of his model via a number of user selectable input options.

4. WASP would, by splitting the kinetic portion of the mass balance Equation (2.16) away from the remaining terms (the advective and dispersive transport and the source/sink terms), require the user to develop only a FORTRAN subroutine that would describe the kinetic interaction of the state or water quality variables. This would remove from the user the need to know how WASP handles the remaining portion of the mass balance equation on a FORTRAN coding level, although it is important to understand the mass balance equation on a general level.

WASP is comprised of a mainline program and twenty-eight support subroutines (forty subroutines for the DEC PDP 11/45 version, which includes graphics output) and one user written kinetic subroutine. Before going into a detailed explanation of the program logic and the procedure to follow to develop a new kinetic subroutine some background explanations must be given.

TIME VARIABLE FUNCTIONS

WASP permits the user to specify time variable input for any of the following parameters: exchange coefficients, advective flows, boundary conditions, forcing functions (waste loads) and miscellaneous functions which might be required by a user's kinetic framework. The user specifies the time variable input data for any of the previously mentioned parameters as a series of time and value combinations which WASP uses as a piecewise linear function of time. Suppose Figure 26a presents the observed measurements of daily solar radiation incident upon a body of water.

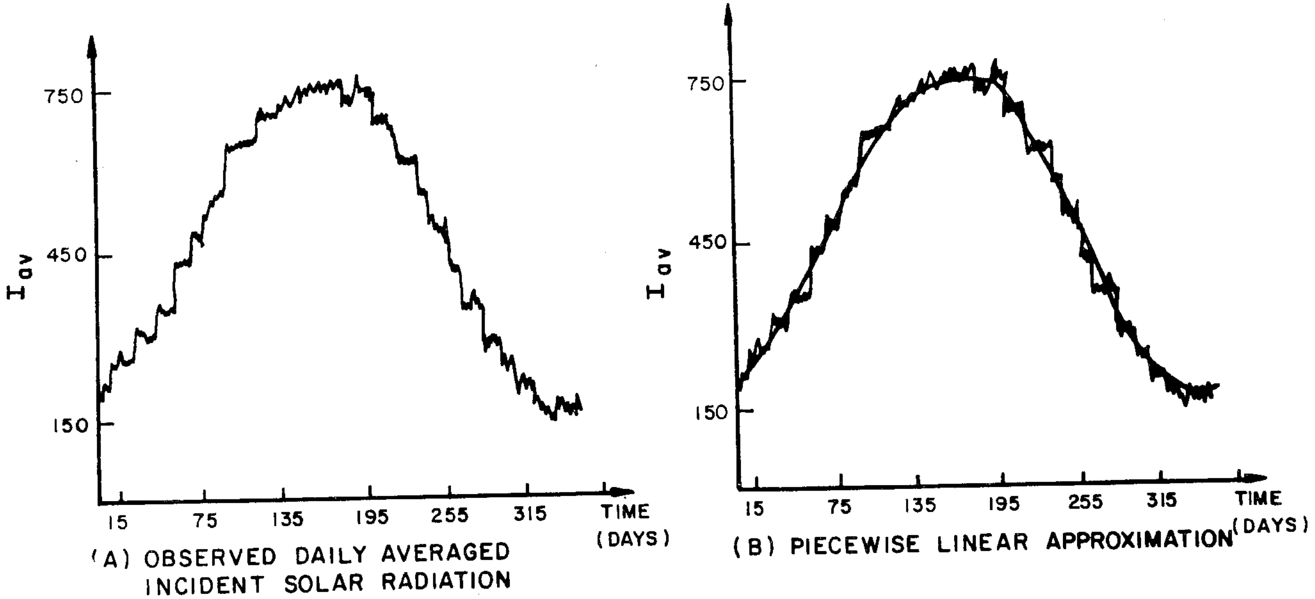
Figure 26b shows how the user might approximate the solar radiation with a piecewise function of time. Table 3 presents the input data the user would supply. Core considerations, especially for the small minicomputer, require WASP to store the entire piecewise linear function on auxiliary disk storage, and maintain core resident only the appropriate information needed to evaluate the function during the current time step. How this is accomplished will be discussed later in the section entitled "Writing a WASPB Subroutine".

TABLE 3. PIECEWISE LINEAR APPROXIMATION FOR FIGURE 26b

I_{av} (ly/day)	Time (days)	I_{av} (ly/day)	Time (days)	I_{av} (ly/day)	Time (days)
187.5	0.	262.5	15.	337.5	45.
487.5	75.	640.0	105.	712.5	135.
750.0	165.	730.0	195.	637.5	225.
487.5	255.	300.9	285.	190.0	315.
150.0	345.	170.0	365.		

KINETIC DATA - CONSTANTS, SEGMENT PARAMETERS AND PIECEWISE LINEAR FUNCTIONS OF TIME

Kinetic data - constants, segment parameters, and miscellaneous functions of time - are read as input data by WASP for use in the user supplied kinetic subroutine, WASPB. The actual choice, of what the constants, segment parameters, and piecewise linear functions of time are to be, is determined by the modeler and the systems analyst as they develop a new kinetic subroutine. The selection and application of kinetic data will be discussed in greater detail in "Writing a WASPB Subroutine", but their basic concept is presented here for background reference. The BOD oxidation rate at 20°C, the saturated growth rate of phytoplankton at 20°C, ammonia to nitrate nitrification rate at 20°C, temperature correction factor for reaeration, and saturated light intensity may all be thought of as constants. Parameters are segment dependent and may include such factors as segment depths, reaeration coefficients, and water temperatures. Kinetic piecewise linear functions might include the daily incident solar radiation for a year, or the fraction of daylight hours over a year. Due to core requirements it was decided not to permit segment dependent piecewise linear functions. How-



(A) OBSERVED DAILY AVERAGED INCIDENT SOLAR RADIATION

(B) PIECEWISE LINEAR APPROXIMATION

Figure 26. Piecewise linear functions.

ever, a user may use segment parameters and piecewise linear functions together to achieve the same effect. For example, a modeler may aggregate observed variations in water temperature over a year for different areas of the water body into one normalized composite piecewise linear function. The modeler would then use the segment parameters for input of the maximum yearly water temperature for the individual segments and a piecewise linear function for input of the normalized aggregate variation during the year.

UNITS

WASP has been programmed using the following internal units conventions. The units of concentration of the state variables (or water quality variables) are assumed to be in mg/l or parts per million parts (ppm). The segment volumes are read in as million cubic feet (MCF). Time is in days. Advective flows are nominally read with units of cubic feet per second (cfs) and internally converted to units of million cubic feet per day (MCF/day) using the following conversion factors:

$$Q \left[\frac{\text{MCF}}{\text{day}}\right] = Q \left[\frac{\text{ft}^3}{\text{sec}}\right] \times \frac{86000 \text{ sec}}{\text{day}} \times \frac{\text{MCF}}{10^6 \text{ft}^3}$$

or

$$Q \left[\frac{\text{MCF}}{\text{day}}\right] = 0.0864 \times Q \text{ cfs}$$

Therefore the term $Q_{ij}c_i$ has units MCF-mg/l/day. The user may read the exchange coefficients as exchange coefficients with units of MCF/day, or as dispersion coefficients, cross-sectional areas, and characteristic lengths and use Equation (2.20) to compute exchange coefficients. If the latter option is chosen the nominal input units for the dispersion coefficients, cross-sectional areas, and characteristic lengths are square miles per day, square feet, and feet, respectively. WASP uses the following factors to convert the dispersion-exchange coefficient to MFC/day:

$$R_{ij} \text{ [MCF/day]} = \frac{E_{ij}[\text{mi}^2/\text{day}] \times A_{ij}[\text{ft}^2]}{L_{ij}[\text{ft}]} \times \frac{5280 \text{ ft}^2}{\text{mi}^2} \times \frac{\text{MCF}}{10^6 \text{ft}^3}$$

or

$$R_{ij} \text{ [MCF/day]} = 27.8764 \times \frac{E_{ij}[\text{mi}^2/\text{day}] \times A_{ij}[\text{ft}^2]}{L_{ij}[\text{ft}]}$$

Similarly the term R_{ij} $(c_j - c_i)$ also has units MCF-mg/l/day. Forcing functions, or waste source loads, are nominally read in units of pounds/day (lb/day), and internally converted using the following convention:

$$W_i \text{ [MCF-mg/l/day]} = W_i \text{ [lb/day]} \times 453.59 \frac{\text{gm}}{\text{lb}}$$

$$\times\ 10^3 \frac{\text{mg}}{\text{gm}} \times \frac{1 \text{ ft}^3}{28.32 \text{ liters}} \times 1 \frac{\text{MCF}}{10^6 \text{ft}^3}$$

or

$$W_i \text{ [MCF-mg/l/day]} = \frac{1}{62.43} \times W_i \text{ [lb/day]}.$$

Looking at the term $V \frac{dc_i}{dt}$, we can readily see that it also has units MCF-mg/l/day. In order to have a consistent set of units for Equation (2.15), the kinetic term must also have units MCF-mg/l/day. Table 4 presents some of the possible forms of the kinetic term which meet this requirement. It should be noted that a user may use units other than those nominally expected by WASP as long as he is careful and consistent in their use. Suppose for example a modeler is investigating coliform bacteria as a state variable. Normally coliform bacteria are measured as most probable number per 100 ml of sample or MPN/100 ml. If the user wishes WASP to interpret the pseudo-concentration of bacteria as MPN/100 ml, he should enter his initial conditions and boundary conditions as MPN/100 ml, and use the following convention to enter waste loads as pseudo "lbs/day"

TABLE 4. KINETIC TERM UNITS

Reaction Order	Kinetic Term	Units
Zero	$K_i V_i$	K_i [=] mg/l/day
First	$K_i C_i V_i$	K_i [=] l/day
Michaelis-Menton	$K_i \frac{C_i}{C_i + K_i} V$	K_i [=] mg/l/day
Coupled (derivative term for state variable 1)	$aK_i C_i^{\,l} C_i^{\,m} V_i$	K_i [=] mg of C^l/mg of C^l a [=] mg of C^l/mg of C^m

$$W_i \text{ ["lbs/day"]} = Q_v \text{ [cfs]} \times 0.0864 \times 62.43 \times C_{coli_i} \text{ [MPN/100 ml]}$$

or

$$W_i \text{ ["lbs/day"]} = Q_v \text{ [cfs]} \times 5.394 \times C_{coli} \text{ [MPN/100 ml]}$$

or

$$W_i \text{ ["lbs/day"]} = Q_v \text{ [MGD]} \times 8.34 \times C_{coli} \text{ [MPN/100 ml]}$$

where

Q_v is the waste discharge volume flow rate

C_{coli} is the coliform bacteria "concentration".

WASP MAINLINE AND SUBROUTINE OVERVIEW

The WASP mainline is really just a program control module. As such it performs no computations but does assign the logical units (disk) for temporary scratch files and the dump-save files generated in the user's kinetic subroutine WASP and does control the calling sequence of the WASP subroutine.

In the following subroutine descriptions the references, within parentheses, to Card Groups are with respect to the card by card description of the WASP input data to be found in Section 6, WASP Input Structure. Also, in the following subroutine description occasional reference to internal WASP program variables will be made. A complete definition of these variables is presented in the section entitled, "WASP Common" immediately following this section.

WASP1

WASP1 reads the model identification and system by-pass options for the user's model (Card Group A). WASP1 also performs some variable initialization. Two variables of interest set by WASP1 are NBCPSY and NWKPSY. NBCPSY and NWKPSY are used to indicate the maximum number of boundary conditions and forcing functions that WASP is dimensioned for, respectively.

WASP2

WASP2, depending upon the input option selected by the user, reads Card Group B, the time variable or constant exchange coefficients (or dispersion coefficients, cross-sectional areas, and characteristic lengths with appropriate conversion to exchange coefficients). WASP2 also reads the exchange by-pass options for each system.

WASP3

WASP3 reads the segment volumes (Card Group C).

WASP4

WASP4, depending upon the input option selected by the user, reads the time variable or constant advective flows (Card Group D). WASP4 converts the flows from cfs to MCF/day. WASP4 also reads the flow by-pass options for each system.

WASP5

WASP5, depending upon the input options selected by the user, reads the time variable or constant boundary conditions for each system in the user's model (Card Group E).

WASP6

WASP6, depending upon the input options selected by the user, reads the time variable or constant forcing functions for each system in the user's model (Card Group F).

WASP7

WASP7 reads the kinetic constants, segment parameters, and kinetic piecewise linear functions of time (Card Group G, H, and I, respectively).

WASP8

WASP8 is used to update the piecewise linear functions of time, if any, for exchange coefficients, advective flows, and the miscellaneous kinetic functions. This means computing new slopes and intercepts, and setting a variable to indicate the next simulation time that the functions are to be updated. The following convention is used for the i^{th} update.

$$\text{slope} = \frac{f(t)_{i+1} - f(t)_i}{t_{i+1} - t_i}$$

$$\text{intercept} = f(t)_{i+1}$$

$$\text{next update time} = t_{i+1}$$

WAS8A

WAS8A is used to update the piecewise linear functions of time, if any, for boundary conditions and forcing functions. This means computing new slopes and intercepts for any system or state variable that requires an update, and setting a variable to indicate the next simulation time that the piecewise linear functions are to be updated. The same conventions used in WASP8 are used in WAS8A for computing slopes and intercepts.

WASP9

WASP9 reads the initial conditions or initial segment concentrations for each system or state variable (Card Group J). WASP9 also reads the stability and accuracy criteria (Card Group K) for each system.

WAS10

WAS10 reads the print control options (Card Group L), consisting of the print interval and up to eight system-segment pairs for intermediate printout during the simulation.

WAS11

WAS11 reads the integration control information (Card Group M). The integration control information includes the integration step-size or sizes to be used, the total simulation time, the starting time for the simulation (if not zero), and whether negative solutions will be permitted.

WAS12 - WA12A

WAS12 and WA12A act together to complete the evaluation of the mass balance Equation (2.16). Upon entry to WAS12, only the kinetic portion of the mass balance equation has been evaluated, which for discussion purposes will be noted as $K_iV_iC_i^m$. WAS12 then goes through the following steps:

a. Using the IQ and JQ vectors as drivers, WAS12 computes

$$(V_i\dot{C}_i^m) = (V_i\dot{C}_i^m) + \Sigma Q_{ji}C_j^m - \Sigma Q_{ik}C_i^m$$

or $$(V_i\dot{C}_i^m) = K_iV_iC_i^m + \Sigma Q_{ji}C_j^m - \Sigma Q_{ik}C_i^m$$

where Q may be computed as a function of time using the MQ (slopes) and BQ (intercepts) vectors or Q may be a constant (using BQ vector).

b. Using the IR and JR vectors as drivers, WAS12 computes

$$(V_i\dot{C}_i^m) = (V_i\dot{C}_i^m) + \Sigma R_{ij}\ (C_j^m - C_i^m)$$

or $$(V_i\dot{C}_i^m) = \{K_iV_iC_i^m + \Sigma Q_{ji}C_j^m - \Sigma Q_{ik}C_i^m\} + \Sigma R_{ij}\ (C_j^m - C_i^m)$$

where R may be computed as a function of time using the MR (slopes) and BR (intercepts) vectors, or R may be a constant (using the BR vector).

c. Using the IWK vector as a driver, WAS12 computes

$$(V_i\dot{C}_i^m) = (V_i\dot{C}_i^m) + W_i^m$$

$$\text{or} \quad (V_i\dot{C}_i^m) = \{K_iV_iC_i^m + \Sigma Q_{ji}C_i^m - \Sigma Q_{ik}C_i^m + \Sigma R_{ij}(C_j^m - C_i^m)\} + W_i^m$$

where W may be computed as a function of time using the MWK (slopes) and BWK (intercepts) vectors or W may be a constant (using the BWK vector).

d. WAS12 completes the evaluation of the derivative, $\dot{C}_i^m$ by dividing through by the volume and multiplying by the time scale factor (SCALT)

$$\dot{C}_i^m = \text{SCALT} \times (V_i\dot{C}_i^m)/V_i$$

Note: $\dot{C}_i^m$ now has units $M/L^3/T$ - nominally mg/l/day.

WAS13

WAS13 is used to print the intermediate system-segment pairs during the simulation.

WAS14

WAS14 is used to adjust the integration stepsize as necessary, during the simulation.

TINIT

TINIT is used to set the initial slopes and intercepts for any piecewise linear functions of time if the user starts his simulation at any time other than time equal to zero. The following real case history demonstrates the use of TINIT. For the analysis performed in the Western Delta - Suisun Bay area of San Francisco Bay, time zero of the simulation runs was January 1 of the particular year being studied. However, due to lowered water temperatures, low incident solar radiation, and high extinction coefficients, the phytoplankton growth rate was almost zero for the first two months of the simulation, therefore there was little or no algal growth. In addition this first two month period was also the period of high flows (snowmelt and rains) which, due to the stability criteria presented in Equation (2.21), required the smallest integration step-size. Since the initial conditions did not dramatically change over the first sixty days, it was decided to start the simulation at day sixty for model calibration runs saving some 20 to 25 percent of the running time. Of course, for final verification and projection runs, the simulation was correctly started at time equal to zero.

WAS15

WAS15 is the heart of the integration procedure. It determines the calling sequence for TINIT, WASPB (the user's kinetic subroutine), WAS12, WA12A, WAS13, and WAS14, and performs the second order Runge-Kutta integration. A brief flowchart is presented in Figure 27.

WAS16

WAS16, dependent upon the user's input data, retrieves and prints the state variables (or water quality concentrations) and any other variables of interest (that the user computed in WASPB) from the auxiliary storage (disk files) that were generated in WASPB during the simulation.

WAS17

WAS17, dependent upon the user's input data, retrieves and provides printer plots for the state variables (or water quality concentrations) and any other variables of interest (that the user computed in WASPB) from the auxiliary storage (disk files) that were generated in WASPB during the simulation.

WAS18

WAS18, written especially for the DEC PDP 11/45 at Grosse Ile, provides off-line digital pen plotting capabilities. The pen plots generated are similar to the printer plots available through WAS17, but in addition permit the user to overplot his observed field data for model-data comparison.

WAS19

WAS19, dependent upon the user's input data, retrieves and provides printer plots of the spatial variation at selected times for the state variables (or water quality concentrations) and any other variables of interest (that the user computed in WASPB) from the auxiliary storage (disk files) that was generated in WASPB during the simulation.

Auxiliary plot subrountines called by WAS19 include STR, PLOT, BLKPLN, and ENCOD (for IBM version only).

Miscellaneous Subroutines

WASP also includes a number of subroutines which perform rather trivial operations which the user need not concern himself with. These subroutines include SCALP, WMESS, WERR, SETIA, SETRA and FMTER.

DEC PDP Subroutines

Two special subroutines were written for the DEC computer system. These subroutines FILEOP and FILEOC were necessitated due to the way the DEC operating system handles disk output, i.e., requiring separate core buffers for each disk file. FILEOP and FILEOC permit the disk files to share a

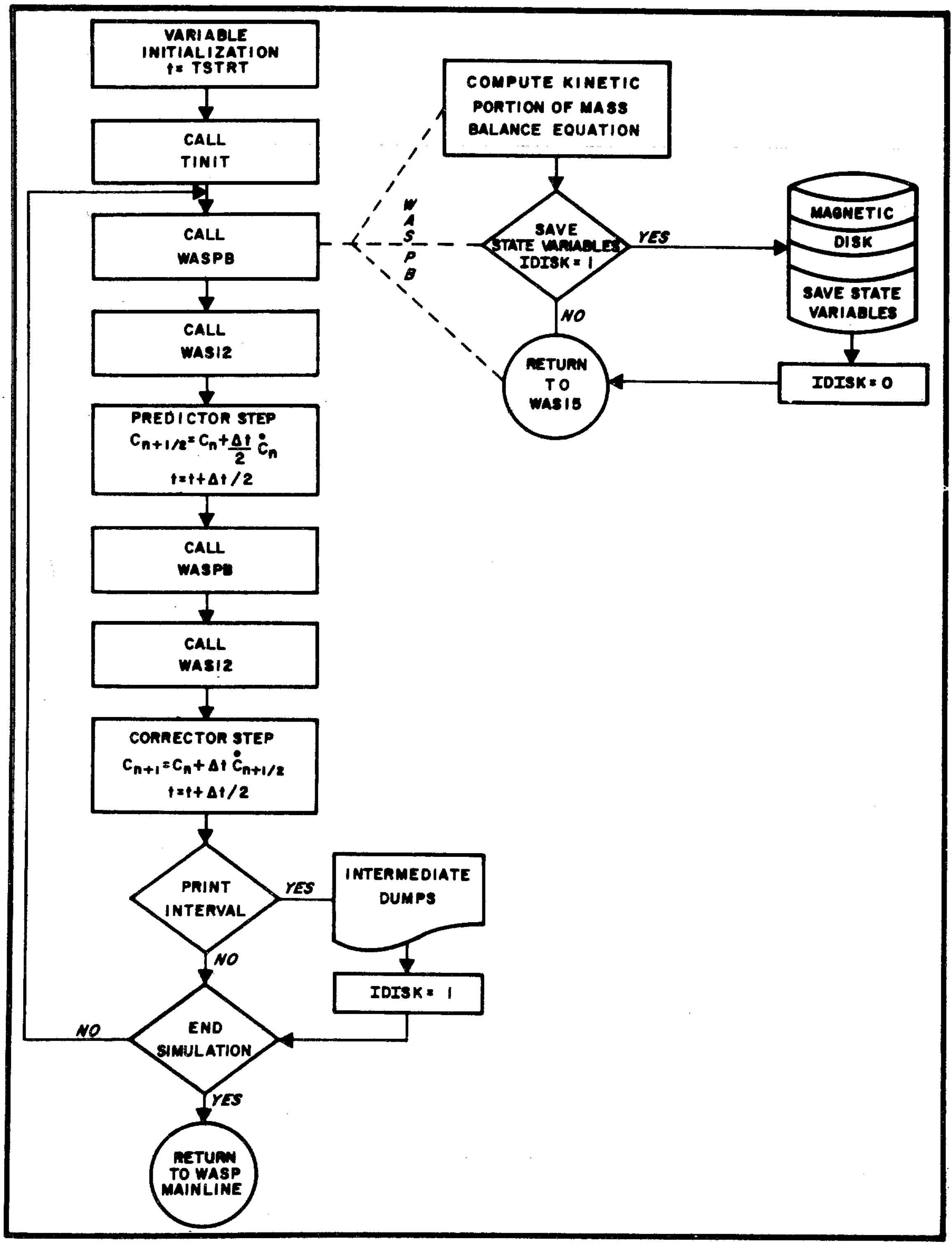


Figure 27. Simplified WAS15 - WASPB flow charts.

common disk buffer, reducing the excessive core requirements required for separate buffers, at little cost to execution time.

WASP COMMON

The following list defines the variables contained in blank COMMON. Blank COMMON is used by WASP as the vehicle to pass information from subroutine to subroutine within the program. The R, I, and * contained within parentheses after the variable name indicate, respectively, whether the variable is a REAL (floating point), or INTEGER (fixed point), and whether the variable is read as input data.

Variable Name	Definition
IN(I)	Device number for reading input data.
OUT(I)	Device number for printer output.
NOSYS(I*)	Number of systems or water quality constituents in the user's model.
ISYS(I)	System currently having its derivatives evaluated.
ISEG(I)	Segment currently having its derivatives evaluated.
ISIM(I*)	Simulation type - currently only time variable is permitted.
LISTG(I*)	User selected option to print exchange coefficient, segment volume, advective flow and boundary condition input data.
LISTC(I*)	User selected option to print forcing function (waste load), kinetic constants, segment parameters, and miscellaneous kinetic time functions, and initial condition input data.
INITB(I)	Internal program indicator which permits the user to perform initialization or to execute special code upon initial entry to the WASPB kinetic subroutine. Initially equal to zero, INITB must be reset by the user in WASPB.
IPRNT(I)	Not currently used.
IDUMP(I*)	System - segment combinations to be printed out during the integration procedure.
IDISK(I)	When checked by the user in the kinetic subroutine, WASPB, IDISK acts as internal program indicator which informs the user when a print interval has

been reached, permitting the user to write the current state variables or segment concentrations to auxiliary storage (disk). Normally IDISK equals zero, but at a print interval it is externally set to one; must be reset by the user before exiting from WASPB.

IREC(I)
Internal counter used to keep track of the number of print intervals generated during the course of the simulation.

MXDMP(I)
Blocking factor or the maximum number of variables saved per segment at each print interval.

IDFRC(I)
Used only in the DEC-PDP version as the record address pointers for the direct access dump files. Not needed for the IBM 370 version since sequential files are used.

NBCPSY(I)
Maximum number of boundary conditions permitted per system; set for a particular WASP configuration in subroutine WASP1.

NWKPSY(I)
Maximum number of forcing functions (waste loads) permitted per system; set for a particular WASP configuration in subroutine WASP1.

SYSBY(I*)
User selected system by-pass indicators. If a user wishes he may choose to by-pass computations for a particular system (or systems), ISYS, for a simulation run by setting SYSBY (ISYS) appropriately.

RBY(I*)
User selected exchange transport by-pass indicators. If a user wishes he may by-pass exchange transport for a particular system, ISYS, by setting RBY(ISYS) appropriately. (Example: if a user had incorporated rooted aquatic plants in his model he would not wish to have them "disperse").

QBY(I*)
User selected advective transport indicators. If a user wishes he may by-pass advective transport for a particular system, ISYS, by setting QBY(ISYS) appropriately. (Example: if a user had incorporated rooted aquatic plants in his model he would not wish to have them transported via flow).

NEGSLN(I*)
Indicates whether the user has chosen to permit WASP to compute negative water quality concentrations (Example: permit negative D.O. deficit, i.e., supersaturation). NEGSLN normally equals zero, but will equal one, if user choses to permit negative solutions.

TIME(R) Current simulation time.

DT(R) Current integration time step.

TZERO(R*) User selectable time for start of simulation. If for example a user's input data for a model was set up such that time zero was January 1, a user may skip computations for January and February and start March 1 by setting TZERO (on input) to 59.

SCALT(R*) Time scale factor. The nominal unit for time in WASP in days. A user may choose to run his simulation in hours by inputting SCALT to be 0.041667 (or 1/24). WASP will then interpret any time specifications (such as the print interval, integration step sizes and total simulation time and the time breaks for any piecewise linear functions) to be hours rather than days.

TEND(R) Ending time for use of the current integration step size. For single integration stepsize input this will be the total simulation time. For multiple integration stepsize histories, when TIME equals TEND, a new integration step size will be chosen and TEND reset.

PRNT(R*) Print interval.

OMEGA(R) Not used in current version of WASP.

ITCHCK(R) Not used in current version of WASP.

MXITER(R) Not used in current version of WASP.

C(R) State variable or water quality concentration array.

CD(R) Derivative array.

CMAX(R*) Stability criteria vector. The vector contains the maximum allowable segment concentration for each system. If any segment exceeds the stability criteria for any system (usually because the integration stepsize is too large) the simulation is terminated.

CMIN(R*) Not used in current version of WASP (although user must include in his input data check).

PARAM(R*) Segment parameters for use in the WASPB kinetic subroutine.

CONST(R*) Constants for use in the WASPB kinetic subroutine.

EVOL(R*) — Segment volumes (nominal input units MCF, nominal internal units MFC).

MVOL(R) — Not used in current version of WASP.

The value for any time variable exchange coefficient, advective flow, boundary condition, forcing function or time-variable function utilized by the WASPB kinetic subroutine is computed using a form of the following equation

VAL = M*TIME + B

where

VAL is the desired value at time = TIME
M is the slope of the piecewise linear function used to approximate the exchange coefficient, flow, etc.
B is the intercept of the piecewise linear function.

BR(R) — Exchange coefficient intercepts.

BQ(R) — Advective flow intercepts.

BBC(R) — Boundary condition intercepts.

BWK(R) — Forcing function intercepts.

BFUNC(R) — Intercepts for the time variable functions required for the WASPS kinetic subroutine.

MR(R) — Exchange coefficient slopes.

MQ(R) — Advective flow slopes.

MBC(R) — Boundary condition slopes.

MWK(R) — Forcing function slopes.

MFUNC(R) — Slopes for the time-variable functions required for the WASPB kinetic subroutine.

Note: If any of the above are time invariant (i.e., constant in time), then the "B" vector (array) will contain the time invariant value of exchange, flow, etc., and the "M" vector (array) will contain zero slope.

IR(I*), JR(I*) — Contain the segment numbers between which exchange is to take place.

IQ(I*), JQ(I*) — Contain the segment numbers between which advective flow is to take place. If the advective flow is positive then JQ will contain the upstream segment number (from which flow is leaving) and IQ will contain the downstream segment number (to which flow will go). If, however, the advective flow is negative then JQ will be considered the downstream segment (flow to) and IQ will be considered the upstream segment (flow from).

IBC(I*) Contains the segment numbers for which boundary conditions have been specified.

IWK(I*) Contains the segment numbers for which forcing functions have been specified (i.e., receiving water segments for waste loads).

IVOPT(I*) User selected volume input option. Currently WASP only permits time-invariant or constant volumes (IVOPT=1).

NOV(I*) Number of volumes (normally NOV equals NOSEG).

IROPT(I*) User selected exchange coefficient input option. IROPT flags the exchange coefficients as constant in time (IROPT=1) or time-variable (IROPT=2,3).

NOR(I*) Number of exchange coefficients read.

IQOPT(I*) User selected advective flow input option. IQOPT flags the flows as constant in time (IQOPT=1) or time-variable (IQOPT=2,3).

NOQ(I*) Number of advective flows read.

IBCOP(I*) User selected boundary condition input options for each system. IBCOP(ISYS) flags the boundary conditions for system ISYS as being constant in time (IBCOP(ISYS)=1) or time-variable (IBCOP(ISYS)=2,3).

NOBC(I*) Number of boundary conditions read for each system.

IWKOP(I*) User selected forcing functions input option for each system. IWKOP(ISYS) flags the forcing functions for system ISYS as being constant in time (IWKOP(ISYS)=1) or time-variable (IWKOP(ISYS)=2,3).

NOWK(I*) Number of forcing functions read for each system.

NOPAM(I*) Number of segment parameters (for use in the WASPB kinetic subroutine) read.

NCONS(I*) Number of constants (for use in the WASPB kinetic subroutine) read.

NFUNC(I*) Number of time variable functions (for use in the WASPB kinetic subroutine) read.

NVOLT(R) Not used in the current version of WASP.

NRT(R) Used if the exchange coefficients are time-variable (approximated by a piecewise linear functions of time). NRT will contain the time at which the next break in the piecewise linear functions will occur, at which point it will be necessary to obtain new slopes (MR) and intercepts (BR).

NQT(R) Used if the advective flows are time-variable (approximated by a piecewise linear functions of time). NQT will contain the time at which the next break in the piecewise linear functions will occur, at which point it will be necessary to obtain new slopes (MQ) and intercepts (BQ).

NBCT(R) Used if the boundary conditions for a system are time-variable (approximated by a piecewise linear functions of time). NBCT(ISYS) will contain the time at which the next break in the piecewise linear functions for system ISYS, will occur, at which point it will be necessary to obtain new slopes (MBC) and intercepts (BBC) for system ISYS.

NWKT(R) Used if the forcing functions for a system are time-variable (approximated by a piecewise linear function of time). NWKT(ISYS) will contain the time at which the next break in the piecewise linear functions, for system ISYS, will occur, at which point it will be necessary to obtain new slopes (MWK) and intercepts (BWK) for system ISYS.

NFUNT(R) Used if time variable functions (approximated as piecewise linear functions of time) have been read for use in the WASPB kinetic subroutine. NFUNT will contain the time at which the next break in the piecewise linear functions will occur, at which point it will be necessary to obtain new slopes (MFUNC) and intercepts (BFUNC).

ITIMV(I) Not used in current version of WASP.

ITIMR(I) Used as a breakpoint counter for obtaining correct slope and intercept values for the time-variable exchange coefficients.

ITIMQ(I) Used as a breakpoint counter for obtaining correct slope and intercept values for time-variable advective flows.

ITIMP(I)	Used as a breakpoint counter for obtaining correct slope and intercept values for the time-variable functions required by the WASPB kinetic subroutine.
ITIMB(I)	Used as a breakpoint counter for obtaining correct slope and intercept values for time-variable boundary conditions.
ITIMW(I)	Used as a breakpoint counter for obtaining correct slope and intercept values for time-variable forcing functions.

WRITING (OR REVISING) A WASPB KINETIC SUBROUTINE

It is the intent and purpose of this section to define a procedure for the modeler/systems analyst to follow in order to develop, program, and debug a new WASPB kinetic subroutine or to revise an already existent kinetic subroutine. The three steps - model development, programming and debugging - are of equal importance in developing a new kinetic subroutine. It is the responsibility of the modeler to develop the modeling or kinetic structure. He must understand in sufficient detail the physical, chemical, and biological principles that form the model structure, so as to be able to write the basic mass balance equation, including kinetics, for each state variable or water quality variable in the model. In addition he must, with the possible assistance of the system analyst, determine what the kinetic constants, segment parameters, and kinetic piecewise linear functions of time are to be. It is the systems analyst's (or programmer's) responsibility to program and debug the WASPB kinetic subroutine. He takes the modeler's mass-balance equations and programs them within the WASP framework. Finally, it is the joint responsibility of the modeler and the systems analyst to "debug" and checkout the newly written (or revised) subroutine. This can be done by setting the print interval equal to the integration interval (stepsize) and taking a few integration steps. Then after getting the state variables printed, the user can check the WASP results by hand computations for a few system-segment-timestep combinations. A kinetic subroutine should never be used unless this checkout procedure is performed.

It is recommended that the following procedure be used in developing anc coding a WASPB kinetic subroutine:

Subroutine Design

1. Review and understand the kinetic structure proposed by the modeler - check for consistent units.

2. Assign WASP system numbers to the state variables.

3. Assign the order that the kinetic constants, segment parameters, and piecewise linear functions are to be read in the user's input deck (the internal assignments are facilitated by use of the FORTRAN

EQUIVALENCE statement with the CONST, PARAM, MFUNC and BFUNC arrays).

Subroutine Coding

4. Determine if any variable and/or program initialization need be performed upon first entry into WASPB. If so, code the variable and/or program logic, and use the variable INITB to determine primary entry.

5. If any piecewise linear functions are required in the user's kinetic subroutine, develop the code necessary to update the slopes and intercepts (via a CALL to WASP8) and to compute the function values using the appropriate slopes and intercepts.

6. Code the segment loop, which evaluates the kinetic portion of the derivative.

7. Code the logic necessary for writing the state variables and other variables of interest, that are computed in WASPB, to auxiliary storage files. Using the variable, IDISK, as an indicator that a print interval has been reached, the user may code the WRITE statement in one of the following two ways:

   a. for large kinetic subroutines where core requirements may be restrictive, it is recommended that the user code the WRITE statements in the segment loop.

   b. for smaller kinetic subroutines or where core requirements are not of concern, the user should code the WRITE statements outside the segment loop (after the end of the segment loop), in order to cut down in I/O time.

EXAMPLE WASPB KINETIC SUBROUTINE

As an example of how this procedure might be followed consider the modeling structure for a simple BOD-DO model. This model will assume first order linear kinetics for BOD removal, DO utilization and reaeration. The BOD removal coefficient, K_r, the deoxygenation coefficient, K_d, and the atmospheric reaeration coefficient, K_a, all at 20°C, will be assigned as kinetic constants. However, these coefficients together with the DO saturation will be temperature corrected using a segment parameter for inputting the segment dependent temperatures. Following the previously defined procedure:

Subroutine Design

1. Review and understand the kinetic structure. The rate equations are given by Equations (3.1) and (3.2)

$$\frac{dBOD_i}{dt} = - K_r(T_i) * BOD_i \tag{3.1}$$

$$\frac{dBOD_i}{dt} = - K_d(T_i) * BOD_i + K_a(T_i) * (DO_{sat}(T_i) - DO_i) \tag{3.2}$$

The formula for temperature correcting reaction coefficient is

$$K(T) = K(20°C)\theta^{T-20} \tag{3.3}$$

and the following empirical non-linear equation will be used to compute DO saturation (1)

$$\begin{aligned} DO_{sat} &= 14.652 - 0.41022T + 0.007991T^2 \\ &\qquad - 0.000077774T^3 \end{aligned} \tag{3.4}$$

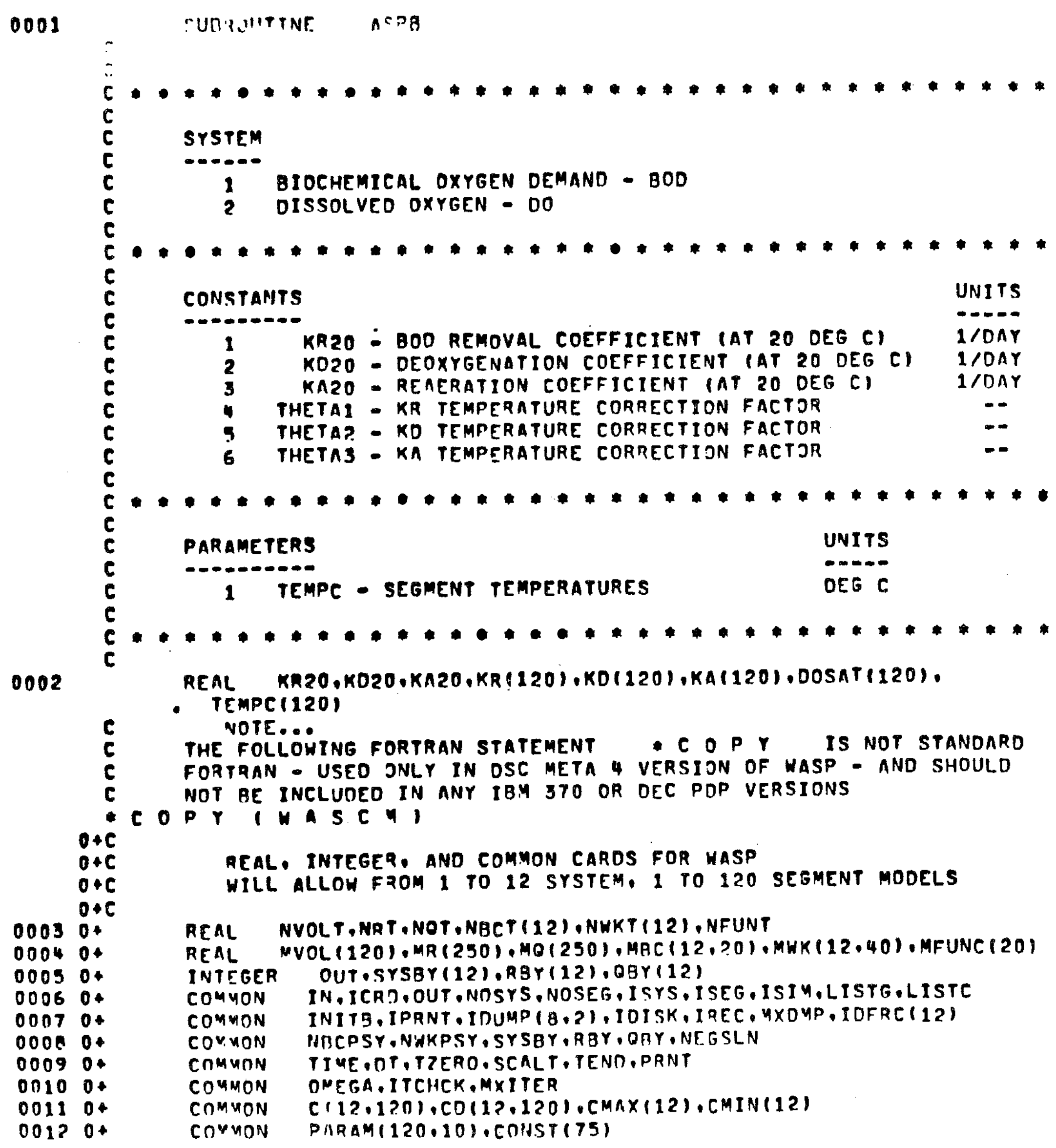
Equation (3.1) states that BOD removal is a first order linear function of the removal coefficient and the segment concentration (Note: there are no "kinetic" sources of BOD). Equation (3.2) contains both a sink (deoxygenation) and source (reaeration) term for DO. K_r need not equal K_d since K_r reflects BOD removal both by biochemical oxidation and by settling, while K_d is a function only of biochemical oxidation.

2. Assign WASP system numbers for the state variables. BOD will be assigned as system 1 and DO will be assigned as system 2.

3. Assign the order for reading kinetic constants, segment parameters and piecewise linear functions of time K_r, K_d, and K_a (all at 20°C) will be assigned as kinetic constants one through three respectively, while the temperature correction factors (θ) will be assigned as four through six. The segment temperatures will be assigned as parameter number one (the only segment parameter). There are no piecewise linear functions to be assigned.

Subroutine Coding - refer to Figure 28 as requested

4. Variable or program initialization. Since temperature is time-invariant in this model framework, the reaction coefficients and DO saturation will also be time-variant, and therefore need only be computed once, at the beginning of the WASPB subroutine. Refering to Figure 28 this task is accomplished using the following FORTRAN statements (line numbers refer to the statement line numbers under the heading C-ERRS)

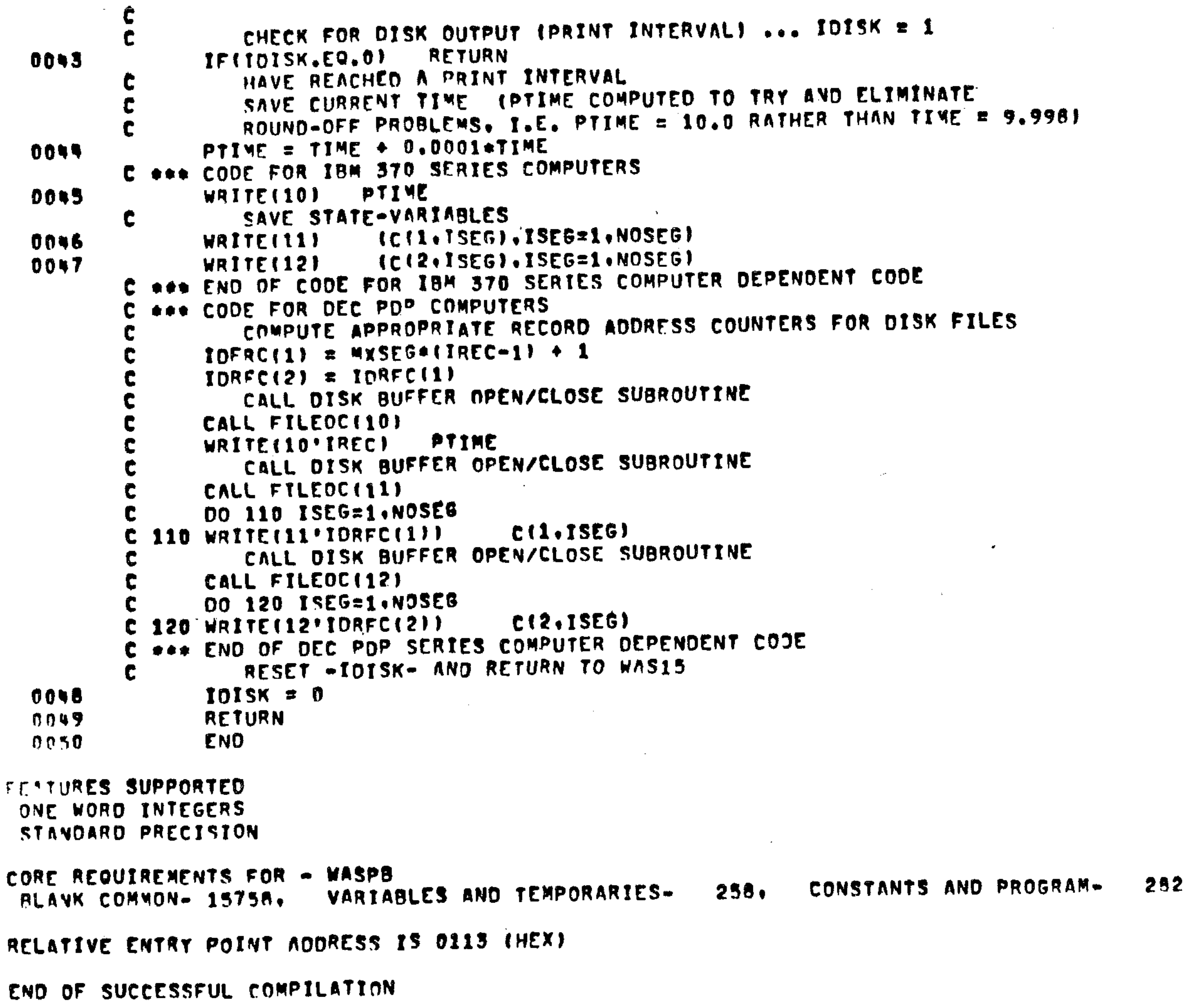


```
0001         SUBROUTINE WASPB
          C
          C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
          C
          C     SYSTEM
          C     ------
          C        1   BIOCHEMICAL OXYGEN DEMAND - BOD
          C        2   DISSOLVED OXYGEN - DO
          C
          C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
          C
          C     CONSTANTS                                                   UNITS
          C     ---------                                                   -----
          C        1      KR20 - BOD REMOVAL COEFFICIENT (AT 20 DEG C)      1/DAY
          C        2      KD20 - DEOXYGENATION COEFFICIENT (AT 20 DEG C)    1/DAY
          C        3      KA20 - REAERATION COEFFICIENT (AT 20 DEG C)       1/DAY
          C        4    THETA1 - KR TEMPERATURE CORRECTION FACTOR             --
          C        5    THETA2 - KD TEMPERATURE CORRECTION FACTOR             --
          C        6    THETA3 - KA TEMPERATURE CORRECTION FACTOR             --
          C
          C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
          C
          C     PARAMETERS                                       UNITS
          C     ----------                                       -----
          C        1   TEMPC - SEGMENT TEMPERATURES              DEG C
          C
          C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
          C
0002            REAL   KR20,KD20,KA20,KR(120),KD(120),KA(120),DOSAT(120),
               .  TEMPC(120)
          C        NOTE...
          C     THE FOLLOWING FORTRAN STATEMENT    * C O P Y    IS NOT STANDARD
          C     FORTRAN - USED ONLY IN DSC META 4 VERSION OF WASP - AND SHOULD
          C     NOT BE INCLUDED IN ANY IBM 370 OR DEC PDP VERSIONS
          * C O P Y  ( W A S C M )
        0+C
        0+C        REAL, INTEGER, AND COMMON CARDS FOR WASP
        0+C        WILL ALLOW FROM 1 TO 12 SYSTEM, 1 TO 120 SEGMENT MODELS
        0+C
0003    0+      REAL     NVOLT,NRT,NOT,NBCT(12),NWKT(12),NFUNT
0004    0+      REAL     MVOL(120),MR(250),MQ(250),MBC(12,20),MWK(12,40),MFUNC(20)
0005    0+      INTEGER    OUT,SYSBY(12),RBY(12),QBY(12)
0006    0+      COMMON   IN,ICRD,OUT,NOSYS,NOSEG,ISYS,ISEG,ISIM,LISTG,LISTC
0007    0+      COMMON   INITB,IPRNT,IDUMP(8,2),IDISK,IREC,MXDMP,IDFRC(12)
0008    0+      COMMON   NBCPSY,NWKPSY,SYSBY,RBY,QBY,NEGSLN
0009    0+      COMMON   TIME,DT,TZERO,SCALT,TEND,PRNT
0010    0+      COMMON   OMEGA,ITCHCK,MXITER
0011    0+      COMMON   C(12,120),CD(12,120),CMAX(12),CMIN(12)
0012    0+      COMMON   PARAM(120,10),CONST(75)
```

Figure 28. Simplified BOD-DO WASPB kinetic subroutine (Version 1).

```
0013 0+      COMMON     BV(120),BR(250),BQ(250),BBC(12,20),BWK(12,40),BFUNC(20)
0014 0+      COMMON     MV,ML,MR,MQ,MBC,MWK,MFUNC
0015 0+      COMMON     IR(250),JR(250),IQ(250),JQ(250),IBC(12,20),IWK(12,40)
0016 0+      COMMON     IVOPT,NOV,IROPT,NOR,IQOPT,NOQ,IBCOP(12),NOBC(12)
0017 0+      COMMON     IWKOP(12),NOWK(12),NOPAM,NCONS,NFUNC
0018 0+      COMMON     NVOLT,NRT,NQT,NBCT,NWKT,NFUNT
0019 0+      COMMON     ITIMV,ITIMR,ITIMQ,ITIMF,ITIMB(12),ITIMW(12)
     0+C
     0+C        THE FOLLOWING LABELED COMMON BLOCK IS NEEDED ONLY
     0+C        FOR THE DEC PDP SERIES COMPUTERS
0020 0+      COMMON    /PDP/  MXSYS,MXSEG
     0+C
       C        PICK-UP KINETIC CONSTANTS THROUGH -CONST- VECTOR
0021         EQUIVALENCE    (CONST(1),KR20) , (CONST(2),KD20) , (CONST(3),KA20)
            .   , (CONST(4),THETA1) , (CONST(5),THETA2) , (CONST(6),THETA3)
       C        PICK-UP SEGMENT PARAMTERS THROUGH -PARAM- ARRAY
0022         EQUIVALENCE    (PARAM(1,1),TEMPC(1))
       C        SAVE CORE STORAGE BY EQUIVALENCE-ING REACTION VECTORS TO UNUSED
       C        PORTIONS OF THE -PARAM- ARRAY
0023         EQUIVALENCE    (PARAM(1,8),KR(1)) , (PARAM(1,9),KD(1))
            .                        (PARAM(1,10),KA(1))
       C
       C        SET -MXDMP-
0024         MXDMP = 1
       C        COMPUTE TEMPERATURE CORRECTED REACTION COEFFICIENTS AND DO SAT-
       C        URATION UPON INITIAL ENTRY INTO THE  WASPB  SUBROUTINE
0025         IF(INITB.NE.0)   GO TO 50
       C        COMPUTE COEFFICIENTS
0026         DO 30 ISEG=1,NOSEG
0027         TEMP = TEMPC(ISEG) - 20.
0028         KR(ISEG) = KR20*THETA1**TEMP
0029         KD(ISEG) = KD20*THETA2**TEMP
0030         KA(ISEG) = KA20*THETA3**TEMP
0031         TEMP = TEMPC(ISEG)
0032         DOSAT(ISEG) = 14.652 - 0.41022*TEMP + 0.007991*TEMP*TEMP
            .         - 0.00077774*TEMP*TEMP*TEMP
0033      30 CONTINUE
       C        RESET -INITB-
0034         INITB = 1
0035      50 CONTINUE
       C
       C               S E G M E N T   L O O P
       C
0036         DO 100 ISEG=1,NOSEG
0037         BOD = C(1,ISEG)
0038         DO = C(2,ISEG)
0039         VOL = BVOL(ISEG)
       C        BOD DERIVATIVE
0040         CD(1,ISEG) = -KR(ISEG)*BOD*VOL
       C        DO DERIVATIVE
0041         CD(2,ISEG) = (-KD(ISEG)*BOD + KA(ISEG)*(DOSAT(ISEG)-DO)) * VOL
0042     100 CONTINUE
```

Figure 28 (Cont.)



```
             C
             C          CHECK FOR DISK OUTPUT (PRINT INTERVAL) ... IDISK = 1
0043                 IF(IDISK.EQ.0)    RETURN
             C          HAVE REACHED A PRINT INTERVAL
             C          SAVE CURRENT TIME  (PTIME COMPUTED TO TRY AND ELIMINATE
             C          ROUND-OFF PROBLEMS, I.E. PTIME = 10.0 RATHER THAN TIME = 9.998)
0044                 PTIME = TIME + 0.0001*TIME
             C ***   CODE FOR IBM 370 SERIES COMPUTERS
0045                 WRITE(10)    PTIME
             C          SAVE STATE-VARIABLES
0046                 WRITE(11)     (C(1,ISEG),ISEG=1,NOSEG)
0047                 WRITE(12)     (C(2,ISEG),ISEG=1,NOSEG)
             C ***   END OF CODE FOR IBM 370 SERIES COMPUTER DEPENDENT CODE
             C ***   CODE FOR DEC PDP COMPUTERS
             C          COMPUTE APPROPRIATE RECORD ADDRESS COUNTERS FOR DISK FILES
             C       IDFRC(1) = MXSEG*(IREC-1) + 1
             C       IDRFC(2) = IDRFC(1)
             C          CALL DISK BUFFER OPEN/CLOSE SUBROUTINE
             C       CALL FILEOC(10)
             C       WRITE(10'IREC)    PTIME
             C          CALL DISK BUFFER OPEN/CLOSE SUBROUTINE
             C       CALL FILEOC(11)
             C       DO 110 ISEG=1,NOSEG
             C 110   WRITE(11'IDRFC(1))      C(1,ISEG)
             C          CALL DISK BUFFER OPEN/CLOSE SUBROUTINE
             C       CALL FILEOC(12)
             C       DO 120 ISEG=1,NOSEG
             C 120   WRITE(12'IDRFC(2))      C(2,ISEG)
             C ***   END OF DEC PDP SERIES COMPUTER DEPENDENT CODE
             C          RESET -IDISK- AND RETURN TO WAS15
0048                 IDISK = 0
0049                 RETURN
0050                 END

FEATURES SUPPORTED
 ONE WORD INTEGERS
 STANDARD PRECISION

CORE REQUIREMENTS FOR - WASPB
 BLANK COMMON- 15758,   VARIABLES AND TEMPORARIES-   258,   CONSTANTS AND PROGRAM-   282

RELATIVE ENTRY POINT ADDRESS IS 0113 (HEX)

END OF SUCCESSFUL COMPILATION
```

Figure 28 (Cont.)

Line number 0002: dimensions or allocates storage for the temperature corrected rate coefficients and DO saturation values.

Line number 0023: conserves storage by equivalencing the dimensional vectors to the unused portions of the segment parameter array (PARAM).

Lines 0025-0034: computes the temperature corrected reaction coefficients and DO saturation. This will be performed only once (saving some computer time) by checking the status of INITB (line no. 0025). Note the user must reset INITB himself (line no. 0034).

5. Piecewise linear functions. Since there are no piecewise linear functions this procedure was not coded.

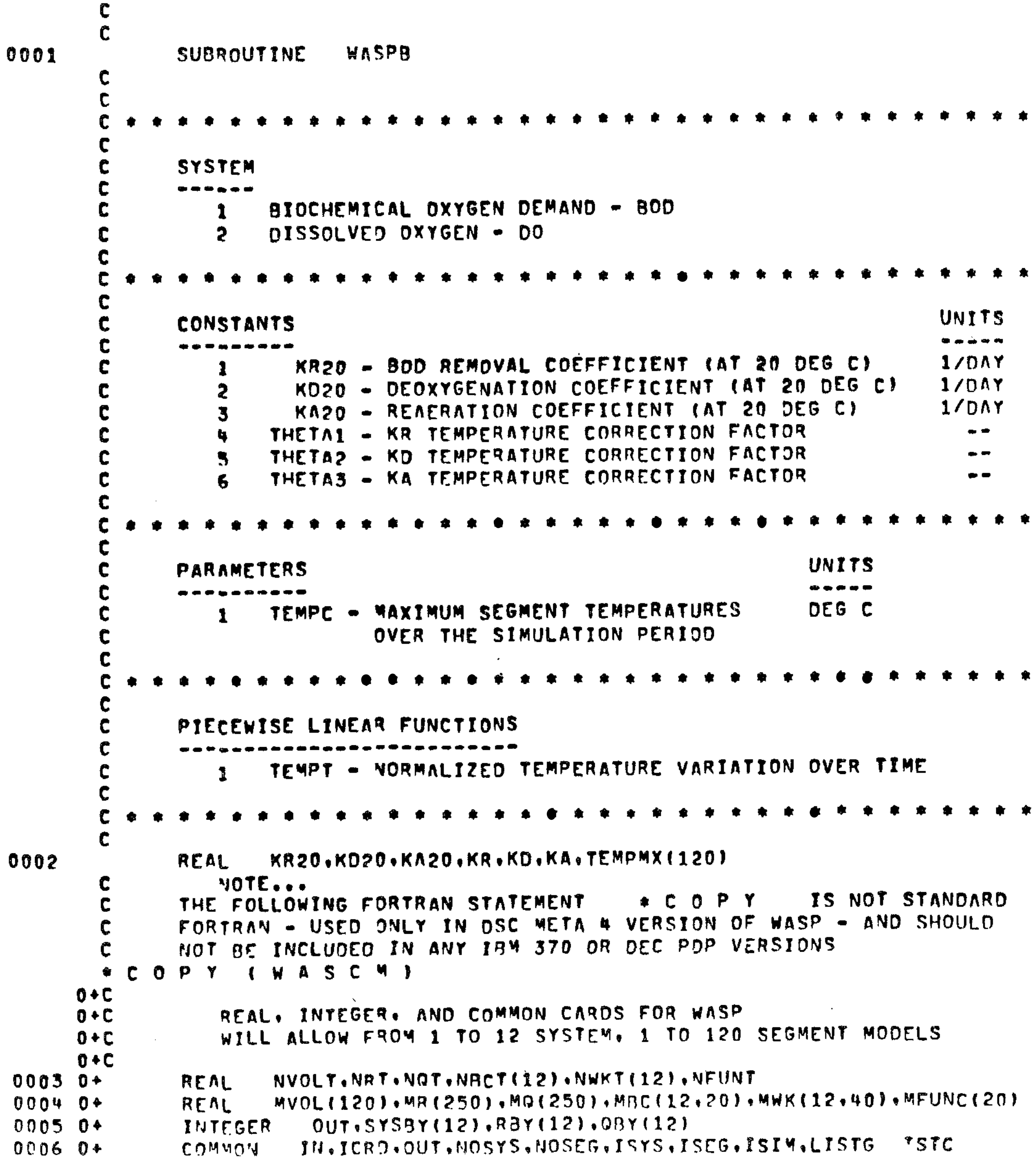
6. Segment loop. Line numbers 0036 through 0042 contain the body of the kinetic subroutine. Note the inclusion of the segment volumes since at this point we are computing V_i dC_i/dt.

7. Saving the state-variables. Line numbers 0043 through 0048 provide the instructions necessary to store the state variables on disk files for the IBM 370 series computer, while the Comment statements immediately following show the code to be utilized on the DEC PDP series computers. Note the user must set MXDMP, as shown on line number 0024.

Figure 29 shows a second version of the subroutine which has temperature as a piecewise linear function of time. Since temperature is time variable the reaction coefficients and DO saturation will also be time variable, necessitating their evaluation at every time step (thereby eliminating the need for step 4). Lines 0025 through 0027 show the FORTRAN code necessary to update the piecewise linear functions of time, while line number 0028 shows the computation of the piecewise linear function at time = TIME, using the MFUNC and BFUNC vectors. As in the previous example there is sufficient storage available (unused portion of the PARAM array) to compute and store the temperature corrected reaction coefficients and the DO saturation values and output these numbers, together with the state variables, outside the segment loop. However, for demonstration purposes the code was included inside the segment loop. Note: the variable MXDMP is now set equal to four to reflect the fact that three additional variables per system are being saved; and the WRITE for system 2 also contains a dummy variable, DUMMY, to "pad out" the WRITE to MXDMP equal 4.

REFERENCES

1. Solubility of Atmospheric Oxygen in Water. 1960. Twenty-ninth progress report of the Committee on San. Engr. Res. of San. Engr. Div., ASCE, Jour. San. Engr. Div., Vol. 86, No. SA4, July 1960, pp. 41-53.



```
           C
           C
0001             SUBROUTINE   WASPB
           C
           C
           C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
           C
           C     SYSTEM
           C     ------
           C        1   BIOCHEMICAL OXYGEN DEMAND - BOD
           C        2   DISSOLVED OXYGEN - DO
           C
           C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
           C
           C     CONSTANTS                                                    UNITS
           C     ---------                                                    -----
           C        1      KR20 - BOD REMOVAL COEFFICIENT (AT 20 DEG C)       1/DAY
           C        2      KD20 - DEOXYGENATION COEFFICIENT (AT 20 DEG C)     1/DAY
           C        3      KA20 - REAERATION COEFFICIENT (AT 20 DEG C)        1/DAY
           C        4    THETA1 - KR TEMPERATURE CORRECTION FACTOR              --
           C        5    THETA2 - KD TEMPERATURE CORRECTION FACTOR              --
           C        6    THETA3 - KA TEMPERATURE CORRECTION FACTOR              --
           C
           C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
           C
           C     PARAMETERS                                        UNITS
           C     ----------                                        -----
           C        1   TEMPC - MAXIMUM SEGMENT TEMPERATURES       DEG C
           C                    OVER THE SIMULATION PERIOD
           C
           C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
           C
           C     PIECEWISE LINEAR FUNCTIONS
           C     --------------------------
           C        1   TEMPT - NORMALIZED TEMPERATURE VARIATION OVER TIME
           C
           C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
           C
0002             REAL   KR20,KD20,KA20,KR,KD,KA,TEMPMX(120)
           C        NOTE...
           C     THE FOLLOWING FORTRAN STATEMENT    * C O P Y    IS NOT STANDARD
           C     FORTRAN - USED ONLY IN DSC META 4 VERSION OF WASP - AND SHOULD
           C     NOT BE INCLUDED IN ANY IBM 370 OR DEC PDP VERSIONS
           * C O P Y  ( W A S C M )
         0+C
         0+C        REAL, INTEGER, AND COMMON CARDS FOR WASP
         0+C        WILL ALLOW FROM 1 TO 12 SYSTEM, 1 TO 120 SEGMENT MODELS
         0+C
0003     0+      REAL   NVOLT,NRT,NQT,NRCT(12),NWKT(12),NFUNT
0004     0+      REAL   MVOL(120),MR(250),MQ(250),MBC(12,20),MWK(12,40),MFUNC(20)
0005     0+      INTEGER   OUT,SYSBY(12),RBY(12),QBY(12)
0006     0+      COMMON   IN,ICRD,OUT,NOSYS,NOSEG,ISYS,ISEG,ISIM,LISTG  *STC
```

Figure 29. Simplified BOD-DO WASPB kinetic subroutine (Version 2).

```
0007 0+       COMMON    INMTB,IPRNT,IDUMP(8,2),IDISK,IREC,MXDMP,IDFRC(12)
0008 0+       COMMON    NBCPSY,NWKPSY,SYSBY,RBY,QBY,NEGSLN
0009 0+       COMMON    TIME,DT,TZERO,SCALT,TEND,PRNT
0010 0+       COMMON    OMEGA,ITCHCK,MXITER
0011 0+       COMMON    C(12,120),CD(12,120),CMAX(12),CMIN(12)
0012 0+       COMMON    PARAM(120,10),CONST(75)
0013 0+       COMMON    BVOL(120),BR(250),BQ(250),BBC(12,20),BWK(12,40),BFUNC(20)
0014 0+       COMMON    MVOL,MR,MQ,MBC,MWK,MFUNC
0015 0+       COMMON    IR(250),JR(250),IQ(250),JQ(250),IBC(12,20),IWK(12,40)
0016 0+       COMMON    IVOPT,NOV,IROPT,NOR,IQOPT,NOQ,IBCOP(12),NOBC(12)
0017 0+       COMMON    IWKOP(12),NOWK(12),NOPAM,NCONS,NFUVC
0018 0+       COMMON    NVOLT,NRT,NQT,NBCT,NWKT,NFUNT
0019 0+       COMMON    ITIMV,ITIMR,ITIMQ,ITIMF,ITIMB(12),ITIMW(12)
     0+C
     0+C         THE FOLLOWING LABELED COMMON BLOCK IS NEEDED ONLY
     0+C         FOR THE DEC PDP SERIES COMPUTERS
0020 0+       COMMON   /PDP/  MXSYS,MXSEG
     0+C
       C         PICK-UP KINETIC CONSTANTS THROUGH -CONST- VECTOR
0021          EQUIVALENCE   (CONST(1),KR20) , (CONST(2),KD20) , (CONST(3),KA20)
             .   , (CONST(4),THETA1) , (CONST(5),THETA2) , (CONST(6),THETA3)
       C         PICK-UP SEGMENT PARAMTERS THROUGH -PARAM- ARRAY
0022          EQUIVALENCE   (PARAM(1,1),TEMPMX(1))
0023          DATA   DUMMY/0.0/
       C
       C         SET -MXDMP-
0024          MXDMP = 4
       C         EVALUATE PIECEWISE LINEAR FUNCTIONS OF TIME
       C      ...BUT FIRST SEE IF THEY NEED TO BE UPDATED
0025          IF(NFUNT.GT.TIME)   GO TO 25
       C         MUST UPDATE SLOPES(MFUNC) AND INTERCEPTS(BFUNC)
0026          CALL WASP8(MFUNC,BFUNC,NFUNC,3,ITIMF,NFUNT,73)
       C         INCREASE BREAK COUNTER  -ITIMF-
0027          ITIMF = ITIMF + 1
       C         NOW GET F(TIME)
0028       25 TEMPN = MFUNC(1)*(TIME-NFUNT) + BFUNC(1)
       C
       C         CHECK FOR DISK OUTPUT (PRINT INTERVAL) ... IDISK = 1
0029          IF(IDISK.EQ.0)   GO TO 50
       C         HAVE REACHED A PRINT INTERVAL
       C         SAVE CURRENT TIME  (PTIME COMPUTED TO TRY AND ELIMINATE
       C         ROUND-OFF PROBLEMS, I.E. PTIME = 10.0 RATHER THAN TIME = 9.998)
0030          PTIME = TIME + 0.0001*TIME
       C         NOTE.....
       C      CODE FOR IBM 370 COMPUTERS WILL BE DESIGNATED WITH  IBM 370  SEQN
       C      CODE FOR DEC PDP COMPUTERS WILL BE DESIGNATED WITH  DEC PDP  SEQN
0031          WRITE(10)   PTIME                                                 IBM 370
       C         COMPUTE APPROPRIATE RECORD ADDRESS COUNTERS FOR DISK FILES     DEC PDP
       C      IDFRC(1) = MXSEG*(IREC-1) + 1                                     DEC PDP
       C      IDRFC(2) = IDRFC(1)                                               DEC PDP
       C         CALL DISK BUFFER OPEN/CLOSE SUBROUTINE                         DEC PDP
       C      CALL FILEOC(10)                                                   DEC PDP
```

Figure 29 (Cont.)

```
                         SEGMENT   LOOP
0032        50 DO 100 ISEG=1,NOSEG
               BOD = C(1,ISEG)
0033           DO = C(2,ISEG)
0034           VOL = BVOL(ISEG)
0035     C        COMPUTE SEGMENT TEMPERATURE
0036           TEMPSG = TEMPN*TEMPMX(ISEG)
         C        TEMPERATURE CORRECT REACTION COEFFICIENTS
0037           TEMP = TEMPSG - 20.
0038           KR = KR20*THETA1**TEMP
0039           KD = KD20*THETA2**TEMP
0040           KA = KA20*THETA3**TEMP
         C        COMPUTE SATURATION
0041           DOSAT = 14.652 - 0.41022*TEMPSG + 0.007991*TEMPSG*TEMPSG
              *                  -0.00077774*TEMPSG*TEMPSG*TEMPSG
         C        COMPUTE DO DEFICIT
0042           DODEF = DOSAT - DO
         C        BOD DERIVATIVE
0043           CD(1,ISEG) = -KR*BOD*VOL
         C        SAVE STATE-VARIABLES
0044           IF(IDISK.EQ.1)        WRITE(11)    BOD,KR,KD,KA                    IBM 370
         C        CALL DISK BUFFER OPEN/CLOSE SUBROUTINE                          DEC PDP
         C     IF(IDISK.EQ.1)   CALL FILEOC(11)                                   DEC PDP
         C     IF(IDISK.EQ.1)   WRITE(11'IDFRC(1))     C(1,ISEG),KR,KD,KA         DEC PDP
         C        DO DERIVATIVE
0045           CD(2,ISEG) = (-KD*BOD + KA*(DOSAT-DO)) * VOL
         C        SAVE STATE-VARIABLES
0046           IF(IDISK.EQ.1)   WRITE(12)   DO,DOSAT,DODEF,DUMMY                  IBM 370
         C        CALL DISK BUFFER OPEN/CLOSE SUBROUTINE                          DEC PDP
         C     IF(IDISK.EQ.1)     CALL FILEOC(12)                                 DEC PDP
         C     IF(IDISK.EQ.1)   WRITE(12'IDFRC(2))   DO,DOSAT,DODEF,DUMMY         DEC PDP
0047       100 CONTINUE
         C        RESET -IDISK- AND RETURN TO WAS15
0048           IDISK = 0
0049           RETURN
0050           END

FEATURES SUPPORTED
 ONE WORD INTEGERS
 STANDARD PRECISION

CORE REQUIREMENTS FOR - WASPB
 BLANK COMMON- 15758,    VARIABLES AND TEMPORARIES-    34,   CONSTANTS AND PROGRAM-   278

RELATIVE ENTRY POINT ADDRESS IS 0036 (HEX)

END OF SUCCESSFUL COMPILATION
```

Figure 29 (Cont.)

SECTION 6

WASP INPUT STRUCTURE

INTRODUCTION

This section will describe, in detail, the input required to run a user's WASP model. No attempt has been made in this section to detail the job control language (JCL) commands required to execute the WASP program since JCL is not only machine dependent but often is site-specific even for the same series of computer.

To arrange the input data into a logical format, the data cards required are divided into fourteen card groups, A through N. The card groups are briefly summarized in Table 5. For each card group, a brief description of each card is given to define the variables which appear within the group, and any options which may be available. Depending upon the structure of the user's model, a certain card group, or cards within a group, may not need to be inputted. Where it is appropriate, the manual informs the user how to avoid inputting unnecessary information.

Provisions for handling a wide range of time and space scales are contained in the data input structure via scale factors. These scale factors facilitate the conversion of the user's time and space units to those consistent with the WASP program. The standard units for the WASP input data are detailed in this manual. Departure from these units necessitates the use of appropriate scale factors. Scale factors may also be used to scale input data in sensitivity analysis. For example, a user may wish to test the effect of increased dispersion upon the model. Rather than altering all the interfacial dispersion coefficients he may simply change the dispersion coefficient scale factor to reflect the appropriate increase in dispersion levels.

INPUT DATA

Card Group A
Model Identification and System Bypass Options

The variables which appear on each card are as follows:

1. Model Identification Numbers

TABLE 5. SUMMARY OF CARD GROUPS

Card Group	
A.	Model Identification and System Bypass Options 1. Model Identification Numbers 2. Title Card 3. Simulation Option 4. System Bypass Option
B.	Exchange Coefficients 1. Number of Exchange Coefficients, Input Option Number 2. Scale Factor 3. Exchange Coefficients 4. Exchange Bypass Options
C.	Segment Volumes 1. Number of Volumes, Input Option Number 2. Scale Factor 3. Volumes
D.	Flow 1. Number of Flows, Input Option Number 2. Scale Factor 3. Flows 4. Flow Bypass Option
E.	Boundary Conditions 1. Number of Boundary Conditions, Input Option Number 2. Scale Factor 3. Boundary Conditions Cards 1-3 are inputted for each system of the model
F.	Forcing Functions 1. Number of Forcing Functions, Input Option Number 2. Scale Factor 3. Forcing Functions Cards 1-3 are inputted for each system of the model
G.	Parameters 1. Number of Parameters 2. Scale Factors 3. Parameters
H.	Constants 1. Number of Constants 2. Constants

TABLE 5. (CONT.)

Card Group	
I.	Miscellaneous Time Functions 1. Number of Time Functions 2. Functions Name, Number of Breaks in Function 3. Time Function Cards 2 and 3 are inputted for each time function required by the model
J.	Initial Conditions 1. Initial Conditions for each system of the model
K.	Stability and Accuracy Criteria 1. Stability Criteria 2. Accuracy Criteria
L.	Intermediate Print Control 1. Print Interval 2. Display Compartments
M.	Integration Control Information 1. Integration Option 2. Time Warp Scale Factor 3. Integration Interval and Total Time
N.	Display Parameters 1. Variable Names 2. Dump Parameters 3. Printer Plot Parameters (Time History) Cards 1 and 2 are read for each system; etc. 4. Printer Plot Parameters (Spatial Profile) 5. Pen Plot Parameters

Card Group A
Model Identification and System Bypass Options

```
    5    10    15    20    25    30    35
MODEL  ISER  IRUN NOSEG NOSYS LISTG LISTC
FORMAT (7I5)
```

MODEL = model designation.

ISER = series designation.

IRUN = run number.

NOSEG = number of model segments.

NOSYS = number of systems.

LISTG = 0, print input data for exchange coefficients, volumes, flows, and boundary conditions on the principal output device.

= 1, suppress printing of input data for exchange coefficients, volumes, flows, and boundary conditions.

LISTC = 0, print input data for forcing functions, segment parameters, constants, miscellaneous time functions, and initial conditions on the principal output device.

= 1, suppress printing of input for forcing functions, segment parameters, constants, miscellaneous time functions, and initial conditions.

MODEL, ISER, IRUN, although not actually used by the WASP program, can assist the user in maintaining a log of computer simulations.

2. Title

```
1                                          80
VERIFICATION OF AUGUST 1973 RIVER SURVEY, REACH 1
FORMAT (20A4)
```

Card column 1-80 contain any information the user feels would be helpful in describing the run and identifying the output for later reference.

3. Simulation Option

Presently WASP permits only time variable simulations, therefore include the following card:

```
1                        24
TIME VARIABLE SIMULATION
FORMAT (6A4)
```

4. Systems Bypass Options

```
          2            4
SYSBY(1) SYSBY(2) ... SYSBY(NOSYS)
FORMAT (19I2)
```

SYSBY(K) = 0, perform the kinetic and transport phenomena associated with system K (numerically integrate the differential equations).

= 1, bypass all kinetic and transport phenomena associated with system K (concentrations read as initial conditions for system K apply throughout simulation period).

Card Group B
Exchange Coefficients

Exchange coefficients may be inputted in one of two formats, actual exchange coefficients or, they may be calculated from inputted dispersion coefficients and accompanying cross-sectional areas and characteristic lengths as per Equation (2.19).

1. Data Input Option Number; Number of Exchange Coefficients

```
    5    10
IROPT   NOR
FORMAT(2I5)
```

Data input options:

IROPT = 1, constant exchange coefficients.

= 2, all exchange coefficients proportional to one piecewise linear approximation.

= 3, each exchange coefficient represented by its own piecewise linear approximation.

= 4, constant exchange coefficients calculated from the dispersion coefficient, cross-sectional area, and characteristic lengths specified for each interface.

= 5, all exchange coefficients proportional to one piecewise linear approximation, calculated from a piecewise linear dispersion coefficient approximation and respective cross-sectional areas, and characteristic lengths.

= 6, each exchange coefficient proportional to its own piecewise linear approximation, calculated from a piecewise linear approximation for the dispersion coefficients, cross-sectional area, and characteristic length specified for each interface.

NOR = number of exchange coefficients.

If no exchange coefficients are to be read, set NOR equal to zero, and continue with Card Group C.

2. Scale Factor for Exchange Coefficients

10
SCALR

FORMAT (E10.3)

SCALR = scale factor for exchange coefficients. Exchange coefficients are normally expressed as million cubic feet per day under options 1, 2, and 3. Options 4, 5, and 6 normally require the following units:

Dispersion Coefficient - Square miles per day

Area - Square feet

Length - Feet

The conversion of sq. mi. - feet/day to MCF/day for options 4, 5, and 6 is handled internally in WASP. If units other than the normal units are necessitated by alteration of the space and time scales, SCALR should be set such that the product of SCALR and the exchange coefficient (or the equivalent computed from the dispersion coefficient) yields MCF/day.

3. Exchange Coefficients

The data input format is determined by the option selected.

Option 1

Each card in this package contains the exchange coefficient information for four interfaces. The number of exchange coefficients read is equal to NOR. The information on each card is described below:

10	15	20	30	35	40
BR(K)	IR(K)	JR(K)	BR(K+1)	IR(K+1)	JR(K+1)

50	55	60	70	75	80
BR(K+2)	IR(K+2)	JR(K+2)	BR(K+3)	IR(K+3)	JR(K+3)

FORMAT (4(F10.0, 2I5)

BR(K) = exchange coefficient between segments IR(K) and JR(K) in million cubic feet per day.

IR(K),JR(K) = segments between which exchange takes place -- the order of the segments is not important; if a segment exchanges with a boundary, the boundary is specified as zero.

Option 2

The card package consists of two sub-packages. Sub-package I contains the exchange coefficient data, while sub-package II contains a detailed specification of the piecewise linear approximation to which all the exchange coefficients are proportional.

Sub-Package I - Exchange Coefficients

Each card in this sub-package contains the exchange coefficient information for four interfaces. The number of exchange coefficients read is equal to NOR. The information on each card is described below:

10	15	20	30	35	40
BR(K)	IR(K)	JR(K)	BR(K+1)	IR(K+1)	JR(K+1)

50	55	60	70	75	80
BR(K+2)	IR(K+2)	JR(K+2)	BR(K+3)	IR(K+3)	JR(K+3)

FORMAT (4(F.10.0, 2I5))

BR(K) = ratio of the exchange coefficient between segments IR(K) and JR(K) to the piecewise linear approximation.

IR(K),JR(K) = segments between which exchange takes place. NOTE: the order of the segments is not important; if a segment exchanges with a boundary, the boundary is specified as zero.

Sub-Package II - Piecewise Linear Approximation

The number of breaks required to describe the piecewise linear approximation is followed by a time series describing the piecewise linear approximation. Each time series element consists of two parts; an exchange value, and a time (normal time scale is days). The input is as follows:

5
NOBRK

FORMAT (I5)

NOBRK = number of values and times used to describe the piecewise linear approximation.

10	20	30	40	50	60	70	80
RT(K)	T(K)	RT(K+1)	T(K+1)	RT(K+2)	T(K+2)	RT(K+3)	T(K+3)

FORMAT (8F10.0)

RT(K) = value of the approximation at time T(K), in million cubic feet per day.

T(K) = time in days; if the length of the simulation exceeds T(NOBRK), the piecewise linear approximation will repeat itself, starting at time T(1); i.e., the approximation is assumed to be periodic with period equal to T(NOBRK), this holds true for all piecewise linear functions time.

Option 3

Each exchange coefficient is defined by a package of cards consisting of two sub-packages. The first sub-package identifies the two segments between which the exchange will take place, and the number of values comprising the piecewise linear approximation. The second sub-package defines the piecewise linear approximation which describes the exchange coefficient. The input is as follows:

Sub-Package I

5	10	15
IR(K)	JR(K)	NOBRK(K)

FORMAT (3I5)

IR(K),JR(K) = segments between which exchange takes place. NOTE: for exchange only, order of segments is not important. If a segment exchanges with a boundary, the boundary is specified as zero.

NOBRK = number of values and times used to describe the piecewise linear approximation. All exchanges must have the same number of breaks, and all breaks must occur at the same time relative to each other.

Sub-Package II - Piecewise Linear Approximation

This consists of a time series describing the piecewise linear approximation. Each time series element consists of two parts; an exchange value, and a time (normal time units - days). The input is as follows:

10	20	30	40	50	60	70	80
RT(K)	T(K)	RT(K+1)	T(K+1)	RT(K+2)	T(K+2)	RT(K+3)	T(K+3)

FORMAT (8F10.0)

RT(K) = value of the piecewise linear approximation at time T(K) in million cubic feet per day.

T(K) = time in days. All break times must agree for all segments, i.e., T(1) must be the same for all exchanges, T(2) must be the same for all exchanges, etc.

Option 4

Each card in this package contains the information to calculate the exchange coefficients for two interfaces. The number of dispersion coefficients is equal to NOR. The information on each card is described below:

10	20	25	30	35	40
E(K)	A(K)	IL(K)	JL(K)	IR(K)	JR(K)

50	60	65	70	75	80
E(K+1)	A(K+1)	IL(K+1)	JL(K+1)	IR(K+1)	JR(K+1)

FORMAT (2(2F10.0, 2F5.0, 2I5))

E(K) = dispersion coefficient for the interface between segment IR(K) and JR(K) in square miles/day.

A(K) = the interfacial cross-sectional area between segments IR(K) and JR(K), in square feet.

IL(K) = the length of segment IR(K), with respect to the IL(K)-JL(K) interface, in feet.

JL(K) = the length of segment JR(K) in relation to the IR(K)-JR(K) interface, in feet. If a segment exchanges with a boundary, the characteristic length of the boundary should be set equal to the length of the segment with which it is exchanging.

IR(K),JR(K) = segments between which exchange takes place. NOTE: for exchange only, order is not important -- if a segment exchanges with a boundary, the boundary is specified as zero.

Option 5

The card package consists of two sub-packages. Sub-package I contains the information necessary to calculate the exchange coefficients, while sub-package II contains a detailed specification of the piecewise linear approximation to which the dispersion coefficients contained in sub-package I are proportional.

Sub-Package I

Each card in this sub-package contains the information necessary to calculate the exchange coefficient for two interfaces. The number of dispersion coefficients is equal to NOR. The information on each card is described below.

10	20	25	30	35	40
E(K)	A(K)	IL(K)	JL(K)	IR(K)	JR(K)

50	60	65	70	75	80
E(K+1)	A(K+1)	IL(K+1)	JL(K+1)	IR(K+1)	JR(K+1)

FORMAT (2(2F10.0, 2F5.0, 2I5)

E(K) = the ratio of the dispersion coefficient between segment IR(K) and JR(K) to the piecewise linear approximation.

A(K) = the interfacial cross-sectional area between segments IR(K) and JR(K), in square feet.

IL(K) = the length of segment IR(K) in relation to the IR(K)-JR(K), in square feet.

JL(K) = the length of segment JR(K) in relation to the IR(K)-JR(K) interface, in feet. If a segment exchanges with a boundary, the characteristic length of the boundary should be set equal to the length of the segment with which it is exchanging.

IR(K),JK(K) = segments between which exchange takes place. NOTE: for exchange only, order is not important.

Sub-Package II - Piecewise Linear Approximation

The number of breaks required to describe the piecewise linear approximation is followed by a time series describing the piecewise linear approximation. Each time series element consists of two parts; a dispersion coefficient and q time (normal units are days). The input is as follows:

10	20	30	40	50	60	70	80
RT(K)	T(K)	RT(K+1)	T(K+1)	RT(K+2)	T(K+2)	RT(K+3)	T(K+3)

FORMAT (8F10.0)

RT(K) = value of the piecewise linear approximation at time T(K), in square miles/day.

T(K) = time in days.

Option 6

Each exchange coefficient is defined by a package of cards consisting of three sub-packages. The first sub-package identifies the two segments between which the exchange will take place, and defines the number of values comprising the piecewise linear approximation. The second sub-package defines the piecewise linear approximation which describes the dispersion coefficient. The third sub-package defines the interfacial cross-sectional area, and the characteristic lengths of the two segments involved. The input is as follows:

Sub-Package I

5	10	15
IR(K)	JR(K)	NOBRK(K)

FORMAT (3I5)

IR(K),JR(K) = segments between which exchange takes place. NOTE: for exchange only, order is not important.

NOBRK = number of values and times used to describe the piecewise linear approximation. All exchanges must have the same number of breaks, and all breaks must occur at the same time relative to one another.

Sub-Package II - Piecewise Linear Approximation

This consists of a time series describing the piecewise linear approximation. Each time series element consists of two parts; a dispersion coefficient, and a time (consistent with the normal time scale of the model). The input is as follows:

10	20	30	40	50	60	70	80
RT(K)	T(K)	RT(K+1)	T(K+1)	RT(K+2)	T(K+2)	RT(K+3)	T(K+3)

FORMAT (8F10.0)

RT(K) = value of the piecewise linear approximation at time T(K), in square miles/day.

T(K) = time in days; all break times must agree for all segments, i.e., T(1) must be the same for all exchanges, T(2) must be the same for all exchanges, etc.

Sub-Package III

This card defines the interfacial cross-sectional area and the characteristic lengths of the segments involved.

10	20	30
A(K)	IL(K)	JL(K)

FORMAT (3F10.0)

A(K) = the interfacial cross-sectional area between segment IR(K) and JR(K) in square feet.

IL(K) = the length of segment IR(K) in relation to the IR(K)-JR(K) interface, in feet.

JL(K) = the length of segment JR(K) in relation to the IR(K)-JR(K) interface in feet.

If a segment exchanges with a boundary, the characteristic length of the boundary should be set equal to the length of the segment with which it is exchanging.

4. Exchange Bypass Options

```
         2        4
RBY(1)   RBY(2)    ....RBY(NOSYS)
FORMAT (19I2)
```

RBY(K) = 0, exchange phenomena occurs in system K.

= 1, bypass exchange phenomena for system (K) (effectively set for all exchange coefficients equal to zero for system K).

Card Group C
Volumes

1. Data Input Option Number; Number of Volumes

```
    5       10
IVOPT      NOV
FORMAT (2I5)
```

Data input options:

IVOPT = 1, constant volumes. Currently WASP only permits constant volumes.

NOV = number of volumes; normally NOV is equal to NOSEG, the number of segments, but for some special input structures, NOV need not equal NOSEG.

2. Scale Factor for Volumes

```
     10
SCALV
FORMAT (E10.3)
```

SCALV = scale factor for volumes; volumes are normally expressed in units of million cubic feet. If other units are necessitated by alteration in the space scale, SCALV should contain the appropriate conversion factor; if normal units are employed, SCALV = 1.0.

3. Volumes

The data input format is determined by the option selected.

Option 1

Each card in this package contains the volume information for eight segments. The number of volumes is equal to NOV. The information on each card is described below.

```
      10          20          30                     70          80
VOL(K)   VOL(K+1)   VOL(K+2)   ....   VOL(K+6)   VOL(K+7)
FORMAT (8F10.0)
```

VOL(K) = volumes of segment K, in million cubic feet. The volumes are to be input consecutively, starting with segment 1, and ending with segment NOV.

Card Group D
Flows

1. Data Input Option Number; Number of Flows

```
    5    10
IQOPT   NOQ
FORMAT (2I5)
```

Data Input Options:

IQOPT = 1, constant flows.

= 2, all flows proportional to one piecewise linear approximation.

= 3, each flow is represented by its own piecewise linear approximation,

NOQ = number of flows.

If no flows are to be inputted, set NOQ to zero, and go to Card Group E.

2. Scale Factor for Flows

```
     10
SCALQ
FORMAT (E10.3)
```

SCALQ = scale factor for flows, flows are normally read in cubic feet per second (cfs).

3. Flows

The data input format is determined by the option selected.

Option 1

Each card in this package contains the flow information for four interfaces, the number of flow specifications is equal to NOQ. The information on each card is described below.

10	15	20	30	35	40
BQ(K)	IQ(K)	JQ(K)	BQ(K+1)	IQ(K+1)	JQ(K+1)

50	55	60	70	75	80
BQ(K+2)	IQ(K+2)	JQ(K+2)	BQ(K+3)	IQ(K+3)	JQ(K+3)

FORMAT (4(F10.0, 2I5)

BQ(K) = flow between segment IQ(K) and JQ(K) in cfs. AESOP convention is: if the flow value is positive, then flow is from segment IQ(K) to JQ(K).

IQ(K) = upstream segment.

JQ(K) = downstream segment.

If flow is from a segment to a boundary, then JQ(K) is set equal to zero; if a flow is from a boundary to a segment, then IQ(K) is set equal to zero.

Option 2

The card package consists of two sub-packages. Sub-package I contains the flow routing while sub-package II contains a detailed specification of the piecewise linear approximation to which all the flows are proportional.

Sub-Package I - Flows

Each card in this sub-package contains the flow information for four interfaces. The number of flow specifications is equal to NOQ. The information on each card is described below:

10	15	20	30	35	40
BQ(K)	IQ(K)	JQ(K)	BQ(K+1)	IQ(K+1)	JQ(K+1)

50	55	60	70	75	80
BQ(K+2)	IQ(K+2)	JQ(K+2)	BQ(K+3)	IQ(K+3)	JQ(K+3)

FORMAT (4(F10.0, 2I5))

BQ(K) = ratio of the flow between segments IQ(K) and JQ(K) to the piecewise linear flow approximation.

IQ(K) = upstream segment.

JQ(K) = downstream segment.

If flow is from a segment to a boundary, then JQ(K) is set equal to zero; if a flow is from a boundary to a segment, then IQ(K) is set equal to zero.

Sub-Package II - Piecewise Linear Flow

The number of breaks required to describe the piecewise linear approximation is followed by a time series describing the piecewise linear flow approximation. Each time series element consists of two parts; a flow and a time. The input is as follows:

10
NOBRK

FORMAT (I5)

NOBRK = number of values and times used to describe the piecewise linear approximation.

10	20	30	40	50	60	70	80
QT(K)	T(K)	QT(K+1)	T(K+1)	QT(K+2)	T(K+2)	QT(K+3)	T(K+3)

FORMAT (8F10.0)

QT(K) = value of the piecewise linear approximation at time T(K), in cubic feet per second.

T(K) = time in days, if the length of the simulation exceeds T(NOBRK), the broken line function will repeat itself, starting at time T(1), i.e., the approximation is assumed to be periodic, with period equal to T(NOBRK).

Option 3

Each flow is defined by a package of cards consisting of two sub-packages. The first sub-package identifies the two segments between which the flow occurs, and the number of values comprising the piecewise linear flow approximation. The second sub-package defines the piecewise linear approximation which describes the flow. The input is as follows:

Sub-Package I

5	10	15
IQ(K)	JQ(K)	NOBRK

FORMAT (3I5)

IQ(K) = upstream segment, flow from segment IQ(K) to JQ(K), assuming positive flow.

JQ(K) = downstream segment flow from segment JQ(K), assuming positive flow.

NOBRK = number of values and times used to describe the broken line approximation. All flows must have the same number of breaks, and all breaks must occur at the same time relative to one another.

Sub-Package II

Sub-package II is a time series describing the piecewise linear approximation. Each time series element consists of two parts: a flow and a time. The input is as follows:

```
    10      20        30       40        50        60        70        80
QT(K)   T(K)   QT(K+1)  T(K+1)  QT(K+2)  T(K+2)  QT(K+3)  T(K+3)
FORMAT (8F10.0)
```

QT(K) = value of the piecewise linear flow approximation at time T(K) in cfs.

T(K) = time in days, if the length of the simulation exceeds T(NOBRK) the broken line function will repeat itself, starting at time, T(1). All break times must agree for all flows, T(2) must be the same for all flows, etc.

4. Flow Bypass Options

The flow bypass options permit the flow transport to be set equal to zero in one or more systems, while maintaining the flow regime (as defined by one of the above options) in the remaining systems.

```
      2        4
QBY(1)   QBY(2)    ....    QBY(ISYS)
FORMAT (19I2)
```

QBY(K) = 0, flow transport occurs in system K.

= 1, bypass the flow transport for system K (effectively set all flows equal to zero in system K).

Card Group E
Boundary Condition

All input is read NOSYS times; once for each system of the model.

1. Data Input Option Number; Number of Boundary Conditions

```
         5           10
IBCOP(K)     NOBC(K)
FORMAT (2I5)
```

Data Input Options:

IBCOP(K) = 1, constant boundary conditions.

= 2, all boundary conditions proportioned to one piecewise linear approximation.

= 3, each boundary condition represented by its own piecewise linear approximation.

NOBC(K) = number of boundary conditions used for system (K).

If no boundary conditions are to be inputted, set NOBC(K) equal to zero, and continue with the next system, or go to the next card group.

2. Scale Factor for Boundary Conditions

10
SCALB

FORMAT (E10.3)

SCALB = scale factor for boundary conditions. Boundary conditions are normally expressed as milligrams per liter (mg/l), or parts per million parts (ppm).

3. Boundary Conditions

The data input format is determined by the option selected.

Option 1

10	15	25	30	40
BBC(K)	IBC(K)	BBC(K+1)	IBC(K+1)	BBC(K+2)

45	55	60	70	75
IBC(K+2)	BBC(K+3)	IBC(K+3)	BBC(K+4)	IBC(K+4)

FORMAT (5(F10.0, I5))

BBC(K) = boundary condition of segment IBC(K) in mg/l.

IBC(K) = segment number to which boundary condition BBC(K) is to be applied.

Option 2

The card package consists of two sub-packages. Sub-package I contains the boundary condition data, while sub-package II contains a detailed specification of the piecewise linear approximation to which all the boundary conditions are to be proportional.

Sub-Package I

Each card in this sub-package contains the boundary condition information for five segments. The number of boundary condition specifications is equal to NOBC. The information on each card is described below.

10	15	25	30	40
BBC(K)	IBC(K)	BBC(K+1)	IBC(K+1)	BBC(K+2)

45	55	60	70	75
IBC(K+2)	BBC(K+3)	IBC(K+3)	BBC(K+4)	IBC(K+4)

FORMAT (5(F10.0, I5))

BBC(K) = ratio of the boundary condition for segment IBC(K) to the piecewise linear approximation.

IBC(K) = segment number.

Sub-Package II - Piecewise Linear Boundary Condition Approximation·

The number of breaks required to describe the piecewise linear boundary condition approximation is followed by a time series describing the boundary approximation. Each time series element consists of two parts; boundary concentration, and a time. The input is as follows:

5
NOBRK

FORMAT (I5)

NOBRK = number of values and times used to describe the piecewise linear approximation.

10	20	30	40	50	60	70	80
BCT(K)	T(K)	BCT(K+1)	T(K+1)	BCT(K+2)	T(K+2)	BCT(K+3)	T(K+3)

FORMAT (8F10.0)

BCT(K) = value of the broken line approximation at time T(K) in mg/l.

T(K) = time at breaks in broken line approximation, in days.

If the length of the simulation exceeds T(NOBRK), the piecewise linear approximation is repeated, starting at T(1), i.e., the approximation is assumed to be period equal to T(NOBRK).

Option 3

Each boundary condition is defined by a package of cards consisting of two sub-packages. The first sub-package identifies the segment associated with the boundary condition and the number of values comprising the piecewise linear approximation. The second sub-package defines the piecewise linear approximation which describes the boundary conditions. All boundary conditions within a system must have the same number of breaks. The input is as follows:

Sub-Package I

5	10
IBC(K)	NOBRK(K)

FORMAT (2I5)

IBC(K) = boundary segment number.

NOBRK(K) = number of values and times used to describe the broken line approximation. The number of breaks must be equal for all boundary conditions within a system.

Sub-Package II - Piecewise Linear Boundary Condition Approximation

The segment number and the number of breaks required to describe the broken line approximation is followed by a time series describing the broken line approximation. Each time series element consists of two parts: a boundary concentration, and a time (consistent with the normal time scale of the model). The number of breaks must be the same for all boundary approximations. The input is as follows:

10	20	30	40	50	60	70	80
BCT(K)	T(K)	BCT(K+1)	T(K+1)	BCT(K+2)	T(K+2)	BCT(K+3)	T(K+3)

FORMAT (8F10.0)

BCT(K) = value of the boundary approximation at time T(K) in mg/l.

T(K) = time in days if the length of the simulation exceeds T(NOBRK), the broken line approximation is repeated, starting at T(1), i.e., the approximation is assumed to be periodic, with period equation to T(NOBRK). All break times must agree for all segment, i.e., T(1) must be the same for all exchanges, T(2) must be the same for all exchanges, etc.

Card Group F
Forcing Functions

All input is read NOSYS times, once for each system of the model.

1. Data Input Option Number; Number of Forcing Functions

5	10
IWKOP(ISYS)	NOWK(ISYS)

FORMAT (2I5)

Data Input Options:

IWKOP(ISYS) = 1, constant forcing functions.

= 2, all forcing functions are proportioned to one piecewise linear approximation.

= 3, each forcing function represented by its own piecewise linear approximation.

NOWK(ISYS) = number of forcing functions used for system ISYS. NOTE: forcing functions may also be considered as sources (loads) or sinks of a water quality constituent. If no forcing functions are to be inputted, set NOWK(ISYS) to zero, and continue with next system or go to next card group.

2. Scale Factor for Forcing Functions

10
SCALW

FORMAT (E10.3)

SCALW = scale factor for forcing functions. Forcing functions are normally read as pounds per day.

3. Forcing Functions

The data input format is determined by the option selected.

Option 1

10	15	25	30	40
BWK(K)	IWK(K)	BWK(K+1)	IWK(K+1)	BWK(K+2)

45	55	60	70	75
IWK(K+2)	BWK(K+3)	IWK(K+3)	BWK(K+4)	IWK(K+4)

FORMAT (5(F10.0, I5))

BWK(K) = forcing function of segment IWK(K), in pounds/day.

IWK(K) = segment number to which forcing function BWK(K) is to be applied.

Option 2

The card package consists of two sub-packages. Sub-package I contains the forcing function data, while sub-package II contains a detailed specification of the piecewise linear approximation to which all the forcing functions are proportional.

Sub-Package I

Each card in this sub-package contains the forcing function information for five segments. The number of specifications is equal to NOWK. The information on each card is described below:

10	15	25	30	40
BWK(K)	IWK(K)	BWK(K+1)	IWK(K+1)	BWK(K+2)

45	55	60	70	75
IWK(K+2)	BWK(K+3)	IWK(K+3)	BWK(K+4)	IWK(K+4)

FORMAT (5(F10.0, I5))

BWK(K) = ratio of the forcing function for segment IWK(K) to the piecewise linear approximation.

IWK(K) = segment number to which forcing function BWK(K) is to be applied.

Sub-Package II - Piecewise Linear Forcing Function Approximation

The number of breaks required to describe the piecewise linear forcing function approximation is followed by a time series describing the forcing function. Each time series element consists of two parts; a function value and a time. The input is as follows:

5
NOBRK

FORMAT (I5)

NOBRK = number of values and times used to describe the piecewise linear approximation.

10	20	30	40	50	60	70	80
WKT(K)	T(K)	WKT(K+1)	T(K+1)	WKT(K+2)	T(K+2)	WKT(K+3)	T(K+3)

FORMAT (8F10.0)

WKT(K) = value of the forcing function at time T(K), in pounds/day.

T(K) = time in days, if the length of the simulation exceeds T(NOBRK), the forcing function approximation is repeated, starting at T(1), i.e., the approximation is assumed to be periodic, with period equal to T(NOBRK).

Option 3

Each forcing function is defined by a package of cards consisting of two sub-packages. The first sub-package identifies the segment associated with the forcing function and the number of values comprising the piecewise linear approximation. The second sub-package defines approximation which describes the forcing function. The input is as follows:

5	10
IWK(K)	NOBRK(K)

FORMAT (2I5)

IWK(K) = segment number which has forcing function BWK(K).

NOBRK(K) = number of breaks used to describe the forcing function approximation. The number of breaks must be equal for all forcing functions within a system.

Sub-Package II - Piecewise Linear Forcing Function Approximation

The segment number and the number of breaks required to describe the piecewise linear forcing function approximation is followed by a time series, describing the forcing function. Each time series element consists of two parts: a function value and a time. The input is as follows:

```
      10   20       30   40       50     60        70     80
WKT(K) T(K) WKT(K+1) T(K) WKT(K+2) T(K+2) WKT(K+3) T(K+3)
FORMAT (8F10.0)
```

WKT(K) = value of the forcing function at time T(K), in pounds/day.

T(K) = time in days, if the length of the simulation exceeds T(NOBRK), the approximation is repeated, starting at T(1), i.e., the approximation is assumed to be periodic with period equal to T(NOBRK). All break times must agree for all segments; i.e, T(1) must be the same for all boundary conditions, T(2) must be the same for all boundary conditions, etc.

Card Group G
Parameters

The definition of the parameters will vary, depending upon the structure and kinetics of the systems comprising each model. The input format however is constant and is detailed below.

1. Number of Parameters

```
    5
NOPAM
FORMAT (I5)
```

NOPAM = number of parameters required by the model. If no parameters are to be inputted, set NOPAM to zero and go to card group H.

2. Scale Factors for Parameters

```
        10          20          30
SCALP(1)   SCALP(2)   SCALP(3)    ...    SCALP(NOPAM)
FORMAT (8E10.3)
```

SCALP(K) = scale factor for parameter group K.

3. Segment Parameters

```
       5                        15                     20
ANAME(K)             PARAM(ISEG,K)             ANAME(K+1)

              30                 35                     45
PARAM(ISEG,K+1)          ANAME(K+2)         PARAM(ISEG,K+2)

         50                60        65                75
ANAME(K+3)  PARAM(ISEG,K+3)  ANAME(K+4)   PARAM(ISEG,K+4)
FORMAT (5(A5, F10.0))
```

ANAME(K) = an optional one to five alpha-numeric character descriptive name for parameter PARAM(ISEG,K).

PARAM(ISEG,K) = the value of parameter ANAME(K) in segment ISEG.

Card Group H
Constants

The definition of the constants will vary, depending upon the structure and kinetics of the systems comprising each model.

1. Number of Constants

5
NCONS

FORMAT (I5)

NCONS = number of constants required by the model.

If no constants are to be inputted, set NCONS equal to zero and continue with the Card Group I.

2. Constants

5	15	20	30	35
ANAME(K)	CONST(K)	ANAME(K+1)	CONST(K+1)	ANAME(K+2)

45	50	60	65	75
CONST(K+2)	ANAME(K+3)	CONST(K+3)	ANAME(K+4)	CONST(K+4)

FORMAT (5(A5, F10.0))

ANAME(K) = an optional one to five alpha-numeric character descriptive name for constant CONST(K).

CONST(K) = the value of constant ANAME(K).

Card Group I
Miscellaneous Time Functions

The definition of the miscellaneous piecewise linear time functions will vary depending upon the structure and the kinetics of the systems comprising each model. The input format however is constant and is detailed below.

1. Number of Time Functions

5
NFUNC

FORMAT (I5)

NFUNC = number of time functions required by the model.

If no time functions are to be inputted, set NFUNC equal to zero, and go to Card Group K.

2. Time Functions

The following package of cards is required for each time function. The first sub-package defines the function name and the number of breaks in the time function. The second sub-package contains a detailed specification of the piecewise linear time function.

Sub-Package I

5	10
ANAME(K)	NOBRK(K)

FORMAT (A5, I5)

ANAME(K) = an optional one to five alpha-numeric character descriptive name for the time function K.

NOBRK(K) = number of breaks used to describe the time function K.

Sub-Package II

10	20	30	40
VALT(K)	T(K)	VALT(K+1)	T(K+1)

50	60	70	80
VALT(K+2)	T(K+2)	VALT(K+3)	T(K+3)

FORMAT (8F10.0)

VALT(K) = value of the function at time T(K).

T(K) = time in days. If the length of the simulation exceeds T(NOBRK), the time function will repeat itself, starting at T(1), i.e., the approximation is assumed to be periodic, with period equal to T(NOBRK). All time functions must have the same number of breaks and all break times must agree for all functions, i.e., T(1) must be the same for all functions, T(2) must be the same for all functions, etc.

Card Group J
Initial Conditions

The initial conditions are the segment concentrations for the state variables at time zero (or the start of the simulation).

1. Initial Conditions

5	15	20	30
ANAME(K)	C(ISYS,K)	ANAME(K+1)	C(ISYS,K+1)

35	45	50	60
ANAME(K+2)	C(ISYS,K+2)	ANAME(K+3)	C(ISYS,K+3)

FORMAT (4(A5, F10.0))

ANAME(K) = an optional one to five alpha-numeric character descriptive name for the initial condition in segment K of system ISYS.

C(ISYS,K) = initial condition in segment K of system ISYS in the appropriate units (normally mg/l or ppm).

The user will be required to input initial conditions for each system even if the system is bypassed or if the initial conditions are zero. The initial conditions are read in one system at a time (from system 1 through system NOSYS), with the concentrations being read from segment 1 through NOSEG within a system packet.

Card Group K
Stability and Accuracy Criteria

1. Stability Criteria

10	20	30		80
CMAX(K)	CMAX(K+1)	CMAX(K+2)	....	CMAX(NOSYS)

FORMAT (8F10.0)

CMAX(K) = stability criteria for system K, i.e., the maximum concentration (normal units mg/l or ppm) for system K which if exceeded by any segments in system K indicates that the numerical integration procedure has become unstable. If unstability occurs an appropriate message is printed and the integration procedure is terminated and a call is made to the display subroutines.

2. Accuracy Criteria

10	20	30		80
CMIN(K)	CMIN(K+1)	CMIN(K+2)	....	CMIN(NOSYS)

FORMAT (8F10.0)

CMIN(K) = originally WASP read the accuracy criteria for system K; i.e., for time variable simulations the minimum concentration (normal units mg/l or ppm) that governs integration step-size control, if the user chooses option 1 (described later). Since it is not recommended that the user utilize this option due to numerical difficulties, just set CMIN = 0.0 for each system.

Card Group L
Intermediate Print Control

1. Print Interval

```
    10
  PRINT
  FORMAT (F10.0)
```

PRINT = print interval in days for time-variable applications. NOTE: The maximum number of print outs = total prototype time/print interval + 1 (for time zero) must be equal to or less than 41.

2. Compartments (System - Segment) to be Displayed

```
  ISYS(1)  ISEG(1)  ISYS(2)  ISEG(2)  ...  ISYS(8)  ISEG(8)
  FORMAT (16I3)
```

ISYS(K), ISEG(K) = system, segment combinations user wishes to have displayed during simulation - user may select a maximum of 8. NOTE: All system-segment concentrations as well as other miscellaneous calculations may be displayed at the end of the simulation, see Card Group N.

Card Group M
Integration Control Information

1. Integration Option - Negative Solution Option

```
     2        4
  INTYP   NEGSLN
  FORMAT (2I2)
```

INTYP = 1, user wishes the WASP program to determine the integration step size (based upon its own accuracy criteria). It is recommended that the user not use this option.

= 2, the user will supply the integration step sizes to be used by WASP. It is recommended that the user utilize this option.

NEGSLN = 0, a user wishes to restrict integration to the positive plane only - this is the normal option selected.

= 1, user will permit the integration procedure to go negative - used for special applications (ex., DO deficit, pH - alkalinity).

2. Time Warp Scale Factor - Starting Simulation Time (TZERO)

10	20
SCALT	TZERO

FORMAT (2E10.4)

SCALT = time warp scale factor - allows the user to completely change the time scale of his simulation via just one card. This scale factor changes all times employed in piecewise linear functions or piecewise linear approximations for volumes, exchanges, flows, etc., as well as print out time.

TZERO = prototype time for start of simulation, usually equal to zero, but user may start at time other than zero (used to initialize any of the piecewise linear time functions).

3. Number of Integration Step Sizes

5
NOSTEP

FORMAT (I5)

NOSTEP = number of integration step sizes to be used in the simulation.

4. Integration Step Size History

10	20	30	40	
DT(K)	TIME(K)	DT(K+1)	TIME(K+1)	...

FORMAT (8F10.0)

DT(K) = integration step size (normal units-days).

TIME(K) = time until which step size DT(K) will be used, then switching to DT(K+1) until TIME(K+1).

Card Group N
Display Parameters

The cards presented for this group are those required by the normal WASP display package. A special subroutine that permits off-line digital pen plotting capability is available for the DEC PDP system. The input data required is described here.

The following two sub-groups are read for each system, starting with system 1 and running through system NOSYS. For each variable-segment combination chosen a time history of the segment will be displayed (dumped).

1. Variable Names

```
        8         16                    80
ANAME(1)   ANAME(2)    ....     (ANAME(10)
FORMAT (10A8)
```

ANAME(K) = a one to eight alpha-numeric character descriptive name for display variable K. The order of these names is determined via the appropriate disk file WRITE in the users kinetic subroutine. NOTE: As presently written WASP permits a maximum of 10 variables per segment to be saved in each system for the IBM 1130, DSC META4, IBM 370 versions and a maximum of 8 variables for the DEC PDP versions.

2. Variable Number, Segment Numbers

```
    3        6          9                   27
VARNO   SEG(K)   SEG(K+1)    ....    SEG(K+7)
FORMAT (9I3)
```

VARNO = the position of the desired variables, to be displayed, in the WRITE file statement in the kinetic subroutine (see previous note).

SEG(K) = segment number to be displayed. NOTE: Order of display unimportant, i.e., need not be sequential.

A blank card terminates display for system, ISYS. Then another Vairable Name card, followed by Variable Number, Segment Number card(s) is read until system NOSYS has been read, then the plot cards will be read.

3. Printer Plot Display Cards

The following cards are read for each system, starting with system 1, and running through NOSYS. The printer plot display sub-group requires 3 cards per plot, and since two plots are formulated per page of printout, the user should input an even number of plots.

3a. Number of Segments and Variable Number for This Plot

```
    2        4
NSPLT    VARNO
FORMAT (2I2)
```

NSPLT = number of segments to be plotted (maximum of five).

VARNO = the position of the desired variable to be plotted, in the WRITE file statement in the kinetic subroutine.

3b. Plotting Scales

10	20
PMIN	PMAX

FORMAT (2F10.0)

PMIN, PMAX = minimum and maximum values, respectively, to be used for this plot.

3c. Segment to be Plotted

3	6		9
SEG(1)	SEG(2)	....	SEG(INSPLT)

FORMAT (5I3)

SEG(K) = segment numbers to be plotted (a maximum of 5 segments/-plot allowed).

A blank card terminates the plotting for system, ISYS.

At this point, unless the user has written additional display routines (such as the PDP graphics), WASP input is finished, and the user should end his deck with the appropriate end of data set indicator.

4. Plot Display Cards (Spatial Profile)

Cards 4a and 4b are read in once, and apply to all spatial plots. There are two types of spatial plots. The first type plots predicted variables, and is controlled by Cards 4c and 4d. The second type plots observed data, and is controlled by Cards 4e, 4f, and 4g. Any number of plots desired can be produced by repeating card combinations 4c-4d and/or 4e-4g.

4a. Spatial Scale

5	10
RM1	RM2

FORMAT (2F5.0)

RM1, RM2 = minimum and maximum river mile values, respectively, to be used for all spatial plots.

4b. Segment River Miles to be Plotted

5	10	15	20	25	30	35
SEG(K)	RM(K)	SEG(K+1)	RM(K+1)	SEG(K+2)	RM(K+2)	SEG(K+3)

40	45	50	55	60	65
RM(K+3)	SEG(K+4)	RM(K+4)	SEG(K+5)	RM(K+5)	SEG(K+6)

70	75	80
RM(K+6)	SEG(K+7)	RM(K+7)

FORMAT (8(I5, F5.0))

SEG(K) = segment number to be plotted.

RM(K) = river mile value for SEG(K).

A maximum of "NOSEG" combinations of SEG-RM pairs are allowed.

For each spatial plot, Cards 4c-4d, or 4e-4g are read.

4c. Predicted Variable Plot Control Information

5	10	15	20	25	30		70
MXTIM	IVAR	YSTR	YSTP	SYSOPT	OVRLAY	...	TITL1

FORMAT (2I5, 2F5.0, 2I5, 40A1)

MXTIM = number of time selections to be included on this plot (maximum of 5).

IVAR = the position of the desired variable to be plotted in the WRITE file statement in the kinetic subroutine.

YSTR,YSTP = minimum and maximum values, respectively, to be used for the Y-axis of this plot.

SYSOPT = system number of the desired variable to be plotted.

OVRLAY = flag to cause this plot to be overlaid with the following plots:

= 0, causes this plot to be printed alone (or with preceeding plot, if OVRLAY on the preceeding plot cards is set to 1).

= 1, causes this plot to be overlaid on the following plot. (Note, although any number of plots can be overlaid, we suggest a maximum of three; YSTR and YSTP values should be compatible for overlaid plots.)

TITL1 = title for plot; when overlaying plots, the first two titles and the last title will be printed.

4d. Time Selections and Characters for Predicted Variable Plot

5	10	15	20	25	26
TIM(1)	TIM(2)	TIM(3)	TIM(4)	TIM(5)	SYMTAB(1)

27	28	29	30
SYMTAB(2)	SYMTAB(3)	SYMTAB(4)	SYMTAB(5)

FORMAT (5F5.0, 5A1)

TIM(K) = time selections for this plot (1-MXTIM).

SYMTAB(K) = plot symbol associated with time TIM(K).

4e. Observed Data Plot Control Information

5	10	15	20	25	30	70	71
FLAG	IUNIT	YSTR	YSTP	NOOBS	OVRLAY	TITL1	OBSSYM

FORMAT (2I5, 2F5.0, 2I5, 40A1, 1A1)

FLAG = flag to indicate observed data.

= 99999, indicates observed data are to be plotted.

IUNIT = unit device number where observed data are to be found (default = 5; optional unit numbers are 82-89).

YSTR,YSTP = minimum and maximum values, respectively, to be used for the Y-axis of this plot.

NOOBS = number of observed data points for this plot.

OVRLAY = flag to cause this plot to be overlaid with the following plot:

= 0, causes this plot to be printed alone (or with preceeding plot, if OVRLAY on the preceeding plot cards is 1).

= 1, causes this plot to be overlaid on the following plot.

TITL1 = title for this plot.

OBSSYM = plot symbol associated with observed data for this plot.

If "IUNIT" on Card 4e equals zero or five, Card 4f is read and Card 4g is skipped. If "IUNIT" equals 82-89, then Card 4f is skipped and Card 4g is read.

4f. River Mile - Observed Data Values

10	20	30	40
RIVMIL(K)	VALUE(K)	RIVMIL(K+1)	VALUE(K+1)

50	60	70	80
RIVMIL(K+2)	VALUE(K+2)	RIVMIL(K+3)	VALUE(K+3)

FORMAT (8F10.0)

RIVMIL(K) = river mile location for observed data point "K".

VALUE(K) = observed value of variable at RIVMIL(K).

If "IUNIT" on Card 4e equals zero or five, Card 4g is skipped.

4g. Format Specification for Data on Auxiliary Input File "IUNIT"

1	80
FMT	

FORMAT (20A4)

FMT = format specification for observed river mile - observed data values on auxiliary input file "IUNIT" (specified on Card 4e). Must begin and end with parentheses, and contain valid formats, such as (2F5.0), (16F5.0), or (F5.0/F5.0)

5. Pen Plot Display Cards

The following cards permit the user to obtain plots of WASP results on a digital pen plotter, should he have access to one on his computer system. The pen plot software has been written such that six "variable vs. time" are generated per 11 inch x 11 inch frame. Therefore, the user must supply the following input data cards in multiples of six. The user should note that unlike the Dump and Printer Plot Display Cards which are organized by system, the Pen Plot Display Cards permit the user to mix the various system outputs on the same frame.

5a. System, Segment, Variable, and Units for This Plot

5	10	15	16	29
SYS	SEG	IVAR		UNITS

FORMAT (3I5, A14)

SYS = system number of the desired variable to be plotted.

SEG = segment number of the desired variable to be plotted.

IVAR = the position of the desired variable, to be plotted, in the WRITE file statement in the kinetic subroutine.

UNITS = an alphanumeric descriptor, which describes the units of the variable to be plotted.

5b. Plotting Scales

10	20
YMIN	YMAX

FORMAT (2F10.0)

YMIN, YMAX = minimum and maximum values, respectively, to be used for this plot.

5c. Field Data

The pen plot subroutine, WAS18, has been written so as to permit the user to overplot theory with observed field data. Basically the input data to be supplied for overplotting field data consists of four pieces of information: a survey number, the time (relative to time zero of the WASP simulation) at which the data was collected, the value (mean value if several samples were collected and are to be aggregated together) of the particular water quality parameter to be plotted, and the standard deviation of the data if applicable. The input data is read as follows:

```
     5         10       20      30                 60      70      80
SURVEY  SURVYT(1)  Y(1)  SD(1)  ....  SURVYT(3)  Y(3)  SD(3)
FORMAT (I5, 3(F5.0, 2F10.0))
```

SURVEY = survey number that the data was collected for. Must be a number from 1 to 5. In actuality SURVEY selects the appropriate plotting symbol for plotting the field data.

SURVYT(I) = time, relative to time zero of the WASP simulation, that the data was collected. Nominal units are days.

Y(I) = the value for the water quality parameter to be plotted. If a number of samples were collected and are aggregated together than Y(I) is the aggregated sample mean.

SD(I) = the standard deviation of the aggregated sample mean (if applicable).

The following description, used together with Table 6, should demonstrate to the user how to prepare data input for the plots. The user will be plotting phytoplankton and secchi depth results for segment 1 through 3 of his model. The kinetic subroutine was written such that the phytoplankton were written the first system file as the first variable (SYS = 1, IVAR = 1) and, the secchi depths were written to the second system file as the fourth variable (SYS = 2, IVAR = 4). Cards 1, 7, 16, 19, 22, and 27 specify that phytoplankton and secchi depth are to be plotted for segments 1, 2, and 3 respectively. Cards 2, 8, 17, 20, 23 and 28 select the plotting scales to be used for the appropriate variables. Cards 3 through 5 contain the field data (times, means, and standard deviations) to be overplotted with the theory for segment 1. Card 6 indicates that no more survey data was collected for segment 1. Cards 9 through 14 contain the field data, from three different surveys, to be overplotted with segment 2 theory. Note that the data for the third survey is just grab samples (no associated standard deviations). Card 15 indicates the end of survey data for segment 2. Card 17 contains grab sample data collected on the third survey, while Card 18 indicates no more survey data for segment 3.

No secchi depth field data was collected for segments 1 and 3, therefore the cards following cards 20 and 28 are blank. Cards 24 and 25 contain single point measurements of secchi depth collected during survey 2. Card 26 indicates the end of survey data. Card 30 indicates that no more plots are required.

TABLE 6. PEN PLOT EXAMPLE

Card Number											
1	1	1	1	UGMS/L							
2		0.0		5.0							
3	1	15.		1.2	0.7	43.	1.0	0.4	90.	1.75	0.6
4	1	180.		3.2	1.1	233.	2.1	0.76	271.	1.43	0.6
5	1	311.		0.9	0.41						
6											
7	1	2	1	UGMS/L							
8		0.0		5.0							
9	1	15.		1.1	0.8	43.	0.89	0.5	90.	1.93	0.64
10	1	180.		2.97	0.93	233.	2.61	0.4	271.	1.14	0.5
11	1	311.		0.76	0.46						
12	2	210.		2.54	0.86	245.	1.76	0.7	285.	1.0	0.41
13	2	320.		0.76	0.33	360.	0.89	0.7			
14	3	60.		1.4		182.	2.6		305.	1.04	
15											
16	1	3	1	UGMS/L							
17	3	60.		1.23		182.	1.97		305.	0.89	
18											
19	2	1	4	/METER							
20		0.0		10.0							
21											
22	2	2	4	/METER							
23		0.0		10.0							
24	2	21.0		3.2		245.	4.6		285.	5.15	
25	2	320.		3.7		360.	3.65				
26											
27	2	3	4	/METER							
28		0.0		10.0							
29											
30											

SECTION 7

MVP THEORY

INTRODUCTION

The task of a modeler in calibrating and verifying a model is complicated by a number of factors. Often two of these factors are the size (number of compartments) and the quantity of data (either lack of or wealth of). The eutrophication analysis performed on the Western Delta-Suisun Bay was performed using an 11 system-39 segment model. The 429 linear and nonlinear equations were numerically integrated for a 12 month period. Using a print interval of 10 days, some 15,000 numbers were generated for each simulation run. The output generated from these simulations was no small task for the modeler to assimilate. The limitations caused by a lack of data in a model calibration are obvious, but what a wealth of data could mean. For example, consider that as a result of the International Field Year on the Great Lakes (IFYGL) some 200,000 observations encompassing some 75 water quality parameters were made on Lake Ontario alone. The modeler attempting to understand the behavior of such large scale models would find an almost impossible task. Even with the most sophisticated computer graphics package the user can only view certain portions of the model (either in variable, temporal or spatial planes). Furthermore, since portions of the model are so interactive, "adjustments" to improve one portion of the model often adversely effect other portions of the model.

An important consideration in a model calibration and verification analysis is the degree to which a model computes a "reasonable" representation of the real world. It is at this point that a considerable degree of confusion and difference exists both in the realm of the model builder and in the mind of the decision maker. What is "reasonable"? Is it sufficient for a model to generate profiles which "look" like what is being observed? Is it sufficient, for example, for a phytoplankton model to simply generate a spring pulse which has been observed or is it necessary to ensure that the magnitude of the pulse is correct using some quantitative measure?

The Model Verification Program provides a statistical framework (1) in which to answer some of these questions in a quantitative way and to enable the user to determine a model's relative behavior and its "reasonableness" in representing the real world. Further, MVP, as will be discussed in Section 8, provides a means for determining the temporal and spatial scale over which a model may be applicable. That is, does the modeling framework chosen by the modeler apply equally well over all temporal scales from hour-to-hour, to day-to-day, to year-to-year, and/or all spatial scales from local near-shore or tidal flats up to open-lake or estuary scales?

STATISTICAL THEORY

Assuming that the user has a sufficient water quality data base and can generate mean and variance statistics over segments and over time (daily, weekly, monthly, or yearly, depending upon the time scale of the problem) for the variables of interest, MVP can perform three simple statistical tests to compare model output to observed field data. These tests are:

1. Test of the difference of means, using a Student's "t" test.

2. A linear regression analysis between observed data and model output.

3. An evaluation of the model using relative error to provide a gross measure of model adequacy.

1. Student's "t"

Let $\bar{x}_{ijk}$ = the observed mean for variable i, segment j, time span or time average (day, week, etc.) k, and $\bar{c}_{ijk}$ = the comparable model computed mean. Permit $\bar{d} = \bar{c}-\bar{x}$, the difference of the calculated and observed means where the triple subscript has been dropped, to be distributed as a Student's "t" probability density function. The test statistic is given by (2)

$$t = \frac{\bar{d} - \delta}{s_{\bar{d}}} \qquad (5.1)$$

where

δ is the difference between the model and the data,
$s_{\bar{d}}$ is the standard deviation given by the pooled variance or

$$s_{\bar{d}}^2 = \frac{2s_x^2}{N} \qquad (5.2)$$

for s_x^2 as the data variance for the specific variable, segment, and time averaged period.

Under the null hypothesis: $\delta = 0$, there is a critical $\bar{d}$ which delineates the region of rejection of the hypothesis and is given by

$$\bar{d}_c = \pm\, t s_{\bar{d}} \qquad (5.3)$$

which for a 95% confidence region (5% change of making a Type 1 error) is given by (5.4)

$$\bar{d}_c = \pm 2s_{\bar{d}} = \frac{2.83}{\sqrt{N}} s_x \qquad (5.4)$$

The distribution of $\bar{d}$ and the critical regions are shown in Figure 30. As indicated if $(-\bar{d}_c<\bar{d}<\bar{d}_c)_{ijk}$ the model is considered verified for variable i, segment j, and time span k, and a score of V=0 is given. For $\bar{d}$ values falling outside the confidence interval a positive value of V, given by (5.5), indicates an overestimate of the mean, while a negative value of V, given by (5.6) indicates an underestimate of the mean.

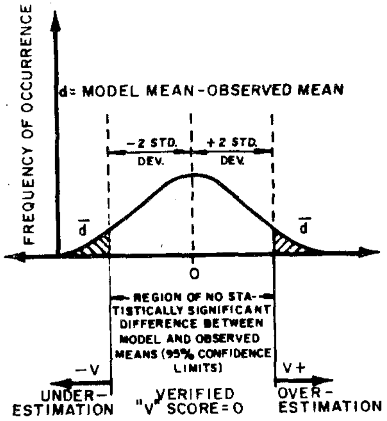

FREQUENCY OF OCCURRENCE
d= MODEL MEAN-OBSERVED MEAN
-2 STD. DEV.
+2 STD. DEV.
$\bar{d}$
$\bar{d}$
O
REGION OF NO STATISTICALLY SIGNIFICANT DIFFERENCE BETWEEN MODEL AND OBSERVED MEANS (95% CONFIDENCE LIMITS)
-V
V+
UNDER-ESTIMATION
VERIFIED "V" SCORE=O
OVER-ESTIMATION

Figure 30. Determination of verification score V.

$$V_{ijk} = \bar{d}_{ijk} - \bar{d}_c \text{ for } \bar{d} > \bar{d}_c \qquad (5.5)$$

$$V_{ijk} = \bar{d}_c - \bar{d}_{ijk} \text{ for } \bar{d} < \bar{d}_c \qquad (5.6)$$

The V score is therefore a measure of the degree to which the model deviates from the observed data, given the spatial and temporal variability within a segment and timespan. More precisely, V is the extent to which the analysis has penetrated into the region of rejection of the null hypothesis. Of course, all the caveats of the application of such statistics apply, especially the change (unknown) of making Type 2 error (that null hypothesis is not rejected when it should be).

Another simple measure that may be used is the number of segments in a given time span that have a V score of zero. Therefore let

$$K_{ijk} = 1 \text{ for } V_{ijk} = 0$$

A score, defined as the S score, for variable i and time span k is given by

$$S_{ik} = \sum_{j=1}^{n} K_{ijk}/n$$

where n is the total number of segments where a V score can be computed, either for the entire water body or just for regions of the water body (as for example tidal flats vs. deep water channels for an estuary or near shore vs. open water for a lake). The score then simply represents the fraction (or percent) of the segments that "passed" the verification test. Since a number of variables may be scored using this verification analysis, an overall aggregated score may be of interest and can also be computed. However, equal verification of all variable may not be of concern. For example, in an eutrophication model one may be willing to accept a lack of verification of ammonia nitrogen in the water body, but be particularly concerned about say, total phosphorus and chlorophyll. Therefore, a series of weights, w_i, can be assigned to each variable i, representing the relative importance of each variable within the analysis. The aggregated S score for a time span k is then given by

$$S_k = \sum_{i}^{r} \sum_{j}^{n} w_i K_{ijk}/(n \sum_{i}^{r} w_i) \qquad (5.7)$$

where r is the number of variables in the aggregated score. S_k therefore represents the weighted fraction of the total number of segment variables that "passed" a Student's "t" test (V=0) for the time span k. It should be noted that not all segments and variables can be tested every time span, so that r and n are functions of the data availability for time span (daily, weekly, etc.) k.

As an example of the utility of this scoring framework consider Figure 12. A comparison of verification scores is made for two different eutrophication kinetic structures that were used in the Lake Ontario analyses (3). The three pairs of bar-histograms reflect the verification scores as a function of the standard deviation or standard error of the mean. The smaller the estimated standard error of the mean the more stringent are the requirements for "passing" the "t" test. It should be noted that the minimum improvement in verification scores, irregardless of the estimated standard error, was 40%, due solely to a change in the kinetic structure. For the readers information the principal kinetic changes between the LAKE1 and LAKE1A models were as follows:

a. The addition of silica as a state variable and the splitting of chlorophyll into diatoms and non-diatoms.

b. Use of threshold nutrient limitation in contrast to product expressions. The growth rate is limited by

$$\text{Min }\left(\frac{[PO_4\text{-}P]}{K_{sp} + [PO_4\text{-}P]}, \frac{[N]}{K_{sN} + [N]}, \frac{[Si]}{K_{sSi} + [Si]}\right.$$

rather then

$$\frac{[PO_4\text{-}P]}{K_{sP} + [PO_4 - P]} \times \frac{[N]}{K_{sN} + [N]} \times \frac{[Si]}{K_{sSi} + [Si]}$$

where K_{sP}, K_{sN}, K_{sSi} are the half-saturation constants for phosphorus $[PO_4\text{-}P]$, nitrogen [N], and silica [Si], respectively.

c. The rate of mineralization of unavailable to available forms of the nutrients, $[PO_4\text{-}P]$, [N], [Si], for uptake by the phytoplankton is dependent, through a Michaelis Menton recycle expression, upon chlorophyll. Therefore, the general expression for conversion of unavailable forms is

$$[\text{Unavailable}] \xrightarrow{R} [\text{Available}]$$

$$\text{for } R = K\theta^{T-20} \frac{[chl\text{-}a]}{[chl\text{-}a] + K_{schl\text{-}a}}$$

where

K = mineralization rate @ 20%.
K = half-saturation constant for chlorophyll [chl-a].

2. Linear Regression Analyses

An alternate perspective on the adequacy of a model can be obtained by regressing the calculated values with observed values. Therefore, let the testing equation be

$$\overline{x} = \alpha + \beta\overline{c} + \varepsilon \qquad (5.8)$$

where α and β are the true intercept and slope respectively between the calculated and observed values and ε is the error in $\overline{x}$. The model Equation (5.8) assumes, of course, that $\overline{c}$, the calculated value from the user's model is known with certainty which is not the actual case. With Equation (5.8), standard linear regression statistics can be computed, including

a. The square of the correlation coefficient, r^2, (the % variance accounted for) between calculated and observed.

b. Standard error of estimate, representing the residual error between model and data.

c. Slope estimate, b of β and intercept estimate, a of α.

d. Tests of significance on the slope and intercept.

In this work, the null hypothesis on the slope and intercept is given by

$\beta = 1$ and $\alpha = 0$.

Therefore, the test statistics

$$\frac{b-1}{s_b} \text{ and } \alpha/s_a$$

are distributed as student's "t" with n-2 degrees of freedom (4). The variance of the slope and intercept, s_b^2 and s_a^2 are computed according to standard formulae. A two-tailed "t" test is conducted on b and a, separately, with a 5% probability in each tail, i.e., a critical value of t of about 2 provides the rejection limit of the null hypothesis.

Regressing the calculated and observed values can result in several situations. Figure 31 (b) and (c) shows that very good correlation may be obtained but a constant fractional bias may exist ($b<1$, $b>1$); also Figure 31 (d) indicates the case of good correlation but for $a > 0$ a constant bias may exist. Evaluation of r^2, b and a together with the residual standard error of estimate can provide an additional level of insight into the comparison between model and data.

3. Relative Error

An additional simple statistical comparison is given by the relative error defined as

$$e = \frac{|\overline{x} - \overline{c}|}{\overline{x}} \tag{5.9}$$

for each variable, segment or time span. Various aggregations of this error across regions can also be calculated and the cumulative frequency of error over time spans or segments can also be computed. The difficulties with this statistic are its relatively poor behavior at low values of $\overline{x}$ and the fact that it does not recognize the variability in the data. In addition, the statistic is poor when $\overline{x} > \overline{c}$ since under that condition the maximum relative error is 100%. As a result, the distribution of this error statistic is most poorly behaved at the upper tail. Nevertheless, if the median error is considered, this statistic is the most easily understood comparison and provides a gross measure of model adequacy. It can also be especially useful in comparisons between models.

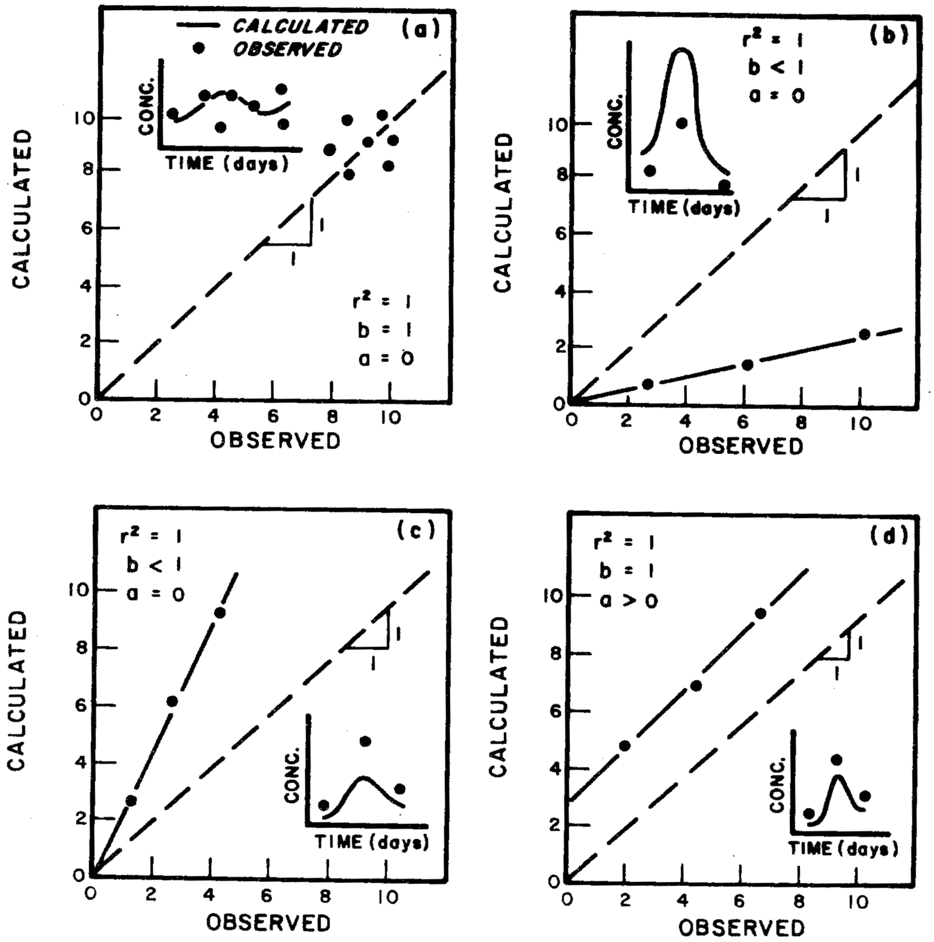


Figure 31. Possible cases in regression between calculated and observed values.

Figure 11 presents a comparison of the relative error analyses for the primary state variables incorporated in the LAKE1 and LAKE1A modeling frameworks. Again, this type of analyses demonstrates the improvement in model verification using a revised kinetic structure.

REFERENCES

1. Thomann, R.V. and R.P. Winfield. 1976. On the Verification of a Three-Dimensional Phytoplankton Model of Lake Ontario. Environmental Modeling and Simulation, ORD and OPM, U.S. EPA.

2. Wine, R.L. 1964. Statistics for Scientists and Engineers. Prentice-Hill, Inc., Englewood Cliffs, N.J. 671 pp.

3. Thomann, R.V., et al. 1978. Verification Analysis of Lake Ontario and Rochester Embayment Three Dimensional Eutrophication Model, in manuscript.

4. Carnahan, B., et al. 1969. Applied Numerical Methods. John Wiley and Sons, Inc., New York, New York. pp. 571-573.

SECTION 8

MVP PROGRAM LOGIC

INTRODUCTION

MVP was designed and written to be a direct compliment to the WASP program, so that a user would have a statistical framework for judging the adequacy of a model developed using the WASP program. As such the same design considerations that were incorporated into the WASP program, were included in MVP. A very important design consideration, carried over from WASP, was that the user not have to make any programming changes to MVP no matter how he formulated his model within the overall WASP framework.

However, the most important design philosophy included in MVP was to provide the user with the greatest flexibility in analyzing the model verification over different temporal and spatial scales. Assuming a sufficient water quality data base is available for the analyses, a user may look at a model's verification or ability to reproduce observed data from week to week up to seasonal time scale or from a segment by segment near field scale up to a whole lake, river, or estuary scale within the same MVP run. This provides both the model builder and the decision maker with useful information. For the model builder it will quickly inform him of areas or time periods where additional modeling effort and/or data collection is required. For the decision maker, the information provided might enable him to formulate waste load allocation strategies that reflect the spatial and temporal scales verified by the model. That is, waste load allocations might be formulated on a regional or seasonal basis, if the model could only be verified over large spatial scales or long term time scales.

The purpose of this section is to review the program logic of MVP. However, unlike the WASP program, which requires the model builder (if not the user as well) to understand the program structure as well as the theory upon which WASP is based, it is sufficient for the model builder and user to understand only the statistical theory which MVP utilizes. In addition, sufficient flexibility has been incorporated into the program so as to make program modification virtually unnecessary. For these reasons the level to which the program logic is detailed will be limited to a rather simple overview. Instead greater emphasis will be placed on providing an understanding of the types of temporal, spatial, and variable aggregation schemes available through user selectable options within the program.

MVP MAINLINE AND SUBROUTINE OVERVIEW

The MVP mainline is really just a program control module and as such it performs no computations but just consists of a calling sequence to various subroutines in the MVP package.

In the following subroutine descriptions the references, within parentheses, to Card Groups are with respect to the card by card description of the MVP input data to be found in Section 9, MVP Input Structure.

MVP01

MVP01 performs various initialization procedures and reads the model configuration and user selectable options for all of Card Group A.

MVP2A

MVP2A is the heart of the model verification analysis procedure. MVP2A controls the calling sequence for the various subroutines which read the field data and the model theory, as computed by WASP, and perform the various statistical analyses on both. A brief flow chart is presented in Figure 32.

MVP02

MVP02 reads the variable name card, score aggregate weighting card(s) and field data to be used in the model verification statistical procedures (Card Groups BB through BD). As the field data is read it is transferred to a workfile for later retrieval when needed by the statistical analyses subroutines MVP03, MVP04, and MVP05. When necessary field data are aggregated over time, if two or more data entries would fall within the same time interval.

MVP03-RESCOR

The linear regression analyses on field observations and the WASP model's theoretical results, are performed by subroutines MVP03 and RSCOR. MVP03 permits the user to select a number of ways to aggregate field observation and model theory over space and time before performing each linear regression analysis. An option to provide graphical display of the regression analysis and the confidence intervals around the regression line is provided. All options for MVP03 are selected by the appropriate cards read from Card Group BF.

MVP04

MVP04 performs the relative error analyses. The percent relative error between the WASP model theory and the field observations is computed and if desired the user may obtain distributive and cumulative relative error histogram plots. The user may, via the input options available through Card Group BG, aggregate field observations and model theory over space and time.

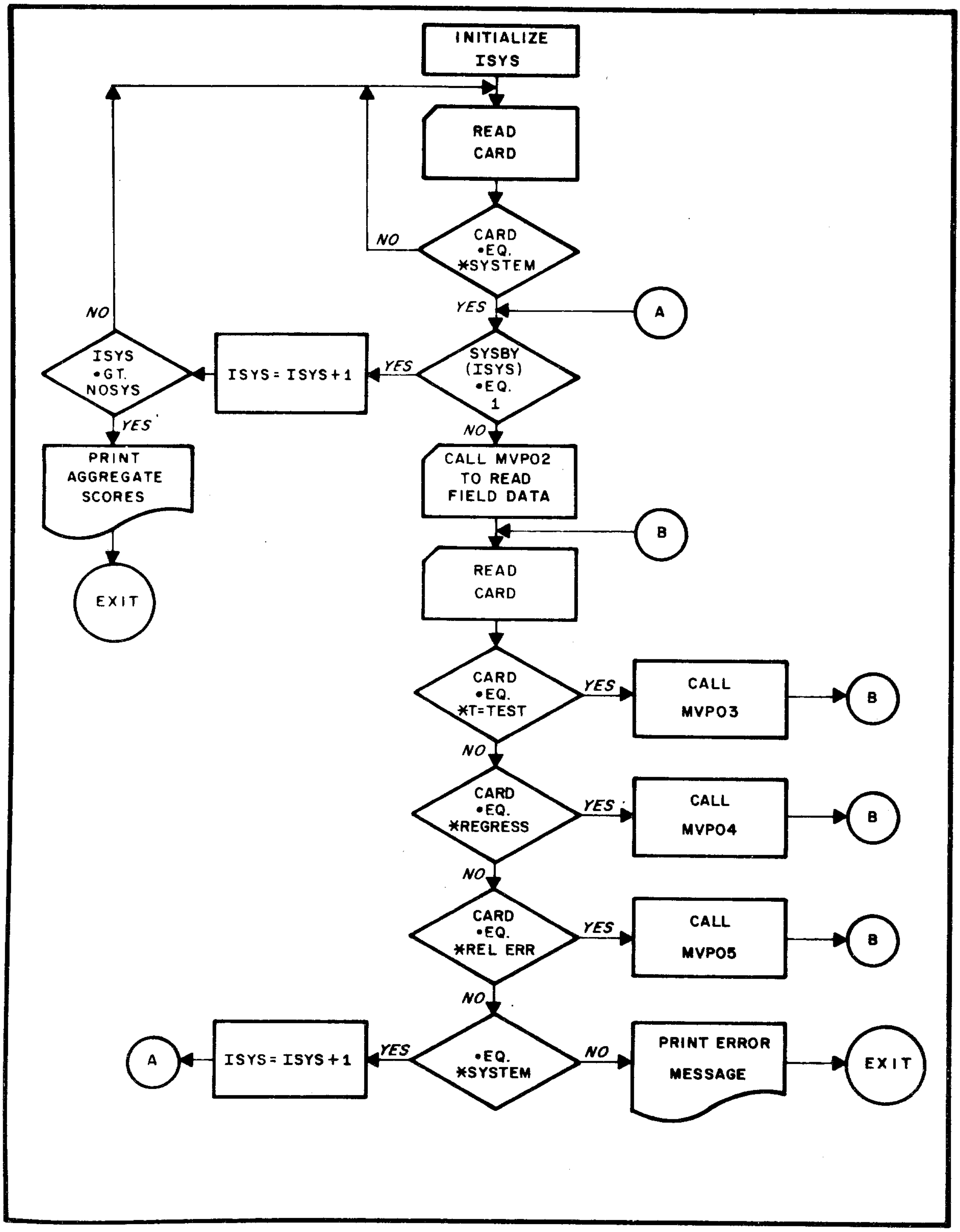


Figure 32. Simplified MVP2A flow chart.

MVP05-TSCOR

MVP05 and its associated subroutine RSCOR compute the Student's "t" test scores on the differences in means between field data and theory results. As with the linear regression analyses and relative error analyses, the user is permitted (via Card Group BE) to aggregate field data and WASP theory over space and time before performing the Student's "t" test analyses.

LOADR

LOADR performs the task of loading the WASP theory results into core memory from the WASP disk storage files. Theory for the current system variable combination is loaded into main memory for all segments and for all times (every print interval specified in the WASP input) so as to minimize disk storage retrievals. LOADR is the one subroutine in the MVP package which is the most sensitive to disk input/output formatting of different computer software operating systems (as is the WASPB kinetic subroutine disk output structure), and so the system analyst must ensure compatibility between LOADR and WASPB.

Miscellaneous Subroutines

The remaining subroutines in the MVP package (MVPHR, COAD2, SETA, and SETRA) perform various heading, printer plot, and variable array initialization procedures.

MVP COMMON

The following list defines the relevant variables of importance contained in blank COMMON of the MVP package. Blank COMMON is used by MVP as the vehicle to pass information from subroutine to subroutine within the program. The R, I, and * contained within parentheses after the variable name indicate, respectively, whether the variable is a REAL (floating point), or INTEGER (fixed point), and whether the variable is read as input data.

Variable Name	Definition
SCORE(R)*	Used to compute the aggregate "t" test scores across systems. SCORE (J,1) contains the number of "t" tests passed, and SCORE (J,2) contains the number of "t" tests performed. The final aggregate score is defined as FSCORE(J) = 100. * SCORE(J,1)/SCORE(J,2).
LSGLG(I*)	Used to indicate which segment or segments are to be included in which aggregate score. Rather than use an entire word to indicate whether or not segment ISEG is to be included in aggregate score J, a single bit (bit J of LSGLG(ISEG) is set to 1 if segment ISEG is to be included, and set to 0 if segment ISEG is not to be included.

INTMT(I*) Used to indicate if the aggregated "t" test score is to be computed over all time intervals (INTMT = 0) or if the "t" test score is to be computed for a specific time interval.

VARWT(R*) VARWT(J,K) are the weights to be assigned each variable (K) for each scoring aggregate (J). This permits the user to give more weight (or importance) to the "t" test score of particular water quality variable than another, when computing an aggregate score across state variables.

TMTRX(R) Used to hold in core the WASP model theory results for a particular water quality variable for all segments over all time intervals. TMTRX is loaded with the model results by subroutine LOADR.

TIMEX(R) Contains the times at which the WASP model theory results were saved on the auxiliary disk files.

IBEG(I*), IEND(I*) Contain the beginning and ending time intervals over which the verification analysis is to be performed.

TMULT(I*) Specifies the number of WASP print intervals that are to be aggregated together to form one WASP theory data point. For example if the user had chosen a print interval of 10 days for his WASP simulation and wishes to perform the MVP analysis using a monthly averaging interval TMULT would be read as 3.

DT(R) Contains the print interval used in the WASP simulation.

NOINT(I) Contans the number of aggregated WASP time intervals that the MVP analysis will be performed on.

HEADR(R*) Contains the user selected heading or title for the MVP run.

NOSYS(I*) The number of systems or state-variables in the user's WASP model.

NOSEG(I*) The number of segments in the user's WASP model.

MXTIM(R*) Is the ending time of the WASP simulation.

MXDMP(I*) Is the blocking factor or the maximum number of variables at each print interval. MXDMP is determined from the user's WASPB kinetic subroutine.

NOAGG(I*) The number of segment aggregations that the user wishes to compute.

NOWGT(I*) The number of segment weighting vectors specified by the user for spatial aggregation.

NOVAR(I*) The current variable number in system ISYS that the user is performing an analysis on.

IN(I) Device number for reading input data.

OUT(I) Device number for printer output.

SOPT(I*) Contains the variance option for the WASP theory as chosen by the user.

IOPT(I*) Contains the degree of freedom option for the Student's T and Snecedor's F distribution parameters as chosen by the user.

SC(I*) Used in conjunction with the SOPT variance option.

ISYS(I) System currently undergoing verification analysis.

LSTVR(I) Used as an internal flag to control the calling of subroutine LOADR. If NOVAR is equal to LSTVR then it is not necessary to call LOADR since the necessary WASP theory is already in core. If NOVAR is not equal to LSTVR then a call to LOADR is made, to obtain the WASP results for variable NOVAR and then LSTVR is set equal to NOVAR.

IDF(I) Used in the DEC-PDP version of MVP as the record address pointers for the direct access WASP dump files. Not utilized in the IBM 370 version since sequential files are used.

SYSBY(I*) User selected system bypass indicators. If a user wishes he may choose to bypass the verification analysis for a particular system (or systems) ISYS, by setting SYSBY(ISYS) appropriately.

IRCRD Not currently used.

IDF40(I),
IDF41(I) Internal record address pointers for the direct access files utilized by MVP.

LO(R), HI(R) Used to store the confidence intervals of the linear regression analysis.

IPTR(I) Used as an internal record address pointer for locating field data for each segment (similar in concept to an ISAM table).

JPTR(I) Used as an interval record address pointer for retrieving WASP theory results.

WGT(R*) Stores the weighting factors to be used in aggregating across state variables for the "t" test scores.

MVP ANALYSIS OVERVIEW AGGREGATION

As stated earlier, MVP permits the user to analyze his model's verification under different temporal and spatial scales, as well as across state variables. It is the purpose of this section to outline the various reasons why a modeler may choose to perform these aggregations, and the various ways that these aggregations may be formulated using MVP. This second point is especially important since, due to the flexibility built into MVP, there are a number of options and aggregation strategies which the user may choose from and unless the user understands what is available to him he may be confused by the MVP input structure.

Segment Weights

If the user wishes to formulate a spatial aggregation, i.e., compute a mean for a state variable or water quality variable that is representative of a number of segments, he must decide how to weight the individual segment concentrations in forming the overall area average. To do this MVP permits a maximum of three segment weighting vectors (Card Group AE). Normally a volume weighted average is what is desired, but other schemes may be included by the user. For example, in a multi-layer model the user may wish to obtain the average light extinction coefficient. In this case the user may choose to use a weighting scheme which is a function of segment depth, rather than volume average.

State Variable Aggregating

A user may determine how well his model verified against all state variables and parameters of interest using the Student's "t" test on the difference between theory and field means as the scoring criteria. The user, via Card Group AG, specifies the segment and time aggregations for which he will compute "t" scores accross variables. Then utilizing Card Group BC (read for each non-by-passed system), he determines what state-variable and/or parameters are to be included in the aggregate score, and what weight each is to be given in forming the overall score.

Spatial Aggregation

Perhaps the best way to demonstrate the flexibility in spatial aggregation is to refer back to Figure 1, the segmentation map for the Western Delta-Suisun Bay eutrophication study. Segments 1,2,3,4,5,6,14,24,25,26, 27,28, and 36 comprise the main stem of the Sacramento River (channelized for shipping), while 7,8,9, and 11 are the main stem segments of the San Joaquin River. The remaining segments are either tidal flats (regions of high algal productivity) or turbid, low flow sloughs. The user may perform his verification analysis on a near field or segment by segment basis for those segments for which is available. Going up the spatial scale he might analyze: the upper Sacramento River by aggregating segments 1 through 6, the lower Sacramento River by aggregating segments 14,24,25,26,27,28, and 36, and finally the entire Sacramento River by including both of the aforementioned groupings. A similar procedure could be followed for the tidal flats. A comparison of verification scores for the main stems or deep channel segments vs. tidal flats could be computed by appropriate segment groupings. Finally, an overall score is computed by aggregating all segments together.

Control of such groupings is provided through Card Groups BE, BF, and BG for the "t" test, linear regression, and relative error statistical analyses respectively.

Temporal Aggregation

With the exception of the linear regression analysis which has a little more flexibility, the user has a relatively small degree of control over temporal aggregation, in that he may aggregate over sequential time periods only. For example, assuming that a user has performed a WASP simulation for a prototype time of one year and used a print interval of ten days, he could perform the verification analysis on a monthly basis by specifying TMULT (Card Group AC) to be equal to 3. A user however (with the exception of linear regression analyses) cannot aggregate over non-sequential time periods, eg., days 0 to 30, with days 330 to 360.

SECTION 9

MVP INPUT STRUCTURE

INTRODUCTION

The Model Verification Program, MVP, was developed and written for use with WASP. It provides for the modeler a statistical framework for judging model verification improvements (or lack of) resulting from adjustments to model parameters or changes in the overall kinetic structure. Since MVP is to be used in conjunction with WASP, many of the concepts and variable definitions used in this section are direct carry overs from the section discussions concerning WASP. Therefore it is expected that the reader have some familiarity with these sections before attempting to use MVP.

The input data required by the MVP program is divided into two major card groups with each major card group having seven minor card groups. These card groups are briefly summarized in Table 7.

TABLE 7. SUMMARY OF MVP CARD GROUPS

A.	MVP Model Configuration Cards
AA	Title Card
AB	System Identification Card
AC	Print Interal Aggregate Card
AD	System Bypass Option Card
AE	Segment Weighting Vector(s) Card
AF	Variance Option Card
AG	Score Aggregate(s) Card
B.	MVP System Information Cards
BA	System Card
BB	Variable Name Card
BC	Score Aggregate Weighting Card(s)
BD	Field Data Card(s)
BE	T-Test Card(s)
BF	Linear Regression Card(s)
BG	Relative Error Card(s)

For each minor card group, a detailed card by card description is presented so as to define the variable fields which appear within the card group and to inform the user of any options which may be available. Depending upon options to be selected by the user, certain minor card groups or cards within a minor card group may not need to be inputted to the MVP program. Where this circumstance might arise, the data input manual informs the user how to avoid inputting the unnecessary information.

MVP INPUT

Major Card Group A
Model Configuration Cards

Major Card Group A is comprised of input data necessary to inform MVP about the configuration of the user's model and the various aggregating and variable weighting schemes the user wishes to apply to the verification analyses. These input data or input variable may be thought of as being system or state variable independent. The following is a card by card description of Major Card Group A.

AA. Title Card

```
1                                                                  80
LINEAR REGRESSION OF AUG. 1973, RIVER SURVEY, REACHES 1,2&3
FORMAT (20A4)
```

Card columns 1-80 contain any information the user feels would be helpful in describing the run and identifying the ouput for later reference.

AB. WASP Model Configuration

```
    5       10       15       20       25       30
NOSYS    NOSEG    MXTIM    MXDMP    NOAGG    NOWGT
FORMAT (6I5)
```

NOSYS = number of systems in the user's WASP model.

NOSEG = number of segments in the user's WASP model.

MXTIM = maximum number of timesteps (print intervals) generated by the user's WASP simulation run. This parameter includes time zero if it is in the simulation. Example: If a user performed a 360 day simulation with printout every 10 days, then the MXTIM = (360/10) + 1 (for time equal zero) = 37.

MXDMP = maximum number of variables dumped by WASP from each system. MXDMP is determined by the user when writing the WASPB kinetic subroutine. See the WASP manual for further details, if necessary.

NOAGG = number of aggregate scroes, aggregated across segments and WASP variables, to be computed, present maximum is 26.

NOWGT = number of segment weighting vectors specified for spatial aggregation; at least one must be specified and there is presently a maximum of three permitted.

AC. Print Interval - Aggregation Card

10	20	25
TBEG	TEND	TMULT

FORMAT (2F10.5, I5)

Since the dates and times of the field statistical data will normally differ from the WASP print intervals, this card allows the user to specify parameters to structure (aggregate) the WASP results to parallel and correspond timewise to the field data.

TBEG = WASP print interval time which the user selects as the starting time point for the MVP analyses.

TEND = WASP print interval time which the user selects as the ending time point for the MVP execution.

TMULT = number of WASP timesteps (print intervals) to be aggregated over the time axis into each MVP Time Interval. If not specified the default is one.

Note: These three parameters must satisfy the following equation for proper MVP execution.

$$\frac{\text{TEND - TBEG}}{\text{TMULT *("WASP PRINT INTERVAL")}} = K \qquad \text{where K - 1,2,3 ...}$$

Example: Given the WASP print intervals ind ays to be:

0, 10, 20, 30, 40, ..., 340, 350, 360.

The user should specify a Print Interval - Aggregate Card of:

a. 0.0 360.0 3
to aggregate MVP results into twelve monthly averages

MVP interval:	1	2	3	12
WASP time:	0^+-30	30^+-60	60^+-90	330^+-360

or

b. 150.0 230. 2
to aggregate MVP results into four semi-monthly summer averages

MVP interval:	1	2	3	4
WASP time:	150^+-170	170^+-190	190^+-210	210^+-230

AD. System Bypass Option Card

2	4	
SYSBY(1)	SYSBY(2)	SYSBY(NOSYS)

FORMAT (19I2)

SYSBY(K) = 0, perform any and all statistical tests specified by the user for any of the variables associated with system K.

= 1, bypass all statistical tests specified by the user for any and all variables associated with system K.

This option permits the user to bypass statistical analyses of any system or combination of systems without having to restructure his input data deck, i.e., it is not necessary to remove the input data cards for a system if it is not to be included in the analyses.

AE. Segment Weighting Vector(s) Card

The user must specify from one to three weighting vectors according to the user's input for NOWGT. The number of elements in each weighting vector is equal to NOSEG, the number of segments in the user's WASP model. The user inputs the first weighting vector, normally the segment volumes, from 1 to NOSEG, eight weighting vector elements per card, using as many cards as necessary. If more than one weighting vector is to be input, start each new vector on a new input card.

10	20		80
WGT(1,IWT)	WGT(2,IWT)	....	WGT(NOSEG,IWT)

FORMAT (8F10.5)

WGT(K,IWT) = weighting vector element K, for weighting vector IWT. K runs from 1 to NOSEG. IWT runs from 1 to NOWGT.

Note: The weighting vector(s) information is utilized by MVP when statistical tests are performed over segment aggregates. Thus if: C_i = concentration of a state variable in segment i, and V_i = the volume of segment i, then for mass conservation

$$C_{AG} = \frac{\sum_{i}^{n} C_i V_i}{\sum_{i}^{n} V_i}$$

where n is the number of segments to be included in this aggregation.

AF. Variance Option Card

Since the WASP program does not calculate standard deviations for its theoretically calculated means, this input card allows the user to specify what the standard deviation of the WASP data will be. The user must also supply an option regarding degree of freedom calculation.

```
       8   10   15      23
   SOPT    SC        TOPT
   FORMAT (2A4, I2, 5X, 2A4)
```

SOPT = 'S TIMES', WASP standard deviations will be a multiple of the field data standard deviations dependent upon the value of SC.

= 'S EQUAL', WASP standard deviations are assumed equal to the field data standard deviations (i.e., SC = 1.0).

= 'S CONST', WASP standard deviations will be assumed to be equal to a constant as specified by SC.

SC = integer value specified in conjunction with SOPT with the meaning as indicated above.

Note: According to the above information SOPT = 'S TIMES' and SC = 1 is equivalent to SOPT = 'S EQUAL'.

TOPT = 'T CONST', Student's T and Snecedor's F distribution parameters are given an average value throughout MVP analyses independent of the actual number of degrees of freedom.

= 'T VARYS', Student's T and Snecedor's F distribution parameters are exact and are a function of the degrees of freedom for small values of degrees of freedom.

Note: Specifying the average value distribution parameter option (T CONST) allows for faster MVP execution, while specifying the exact parameters option (T VARYS) produces more accurate output results. However, result differences become small if the degrees of freedom involved are greater than ten. Thus, it is recommended to use the degrees of freedom varying option (T VARYS) only if the degrees of freedom are consistently small.

AG. Score Aggregate(s) Card

The user is permitted to form a number of spatial and/or temporal aggregations to see how different areas of the model (ex., surface layers vs. bottom layer, summer vs. winter) score relative to one another utilizing the Student's "t" test as the scoring criteria. If NOAGG was specified as zero than this section is bypassed. If NOAGG is greater than zero, then NOAGG input cards are specified

```
   12      16        20        24
TINT   NUMSG   NSV(1)    NSV(2)    NSV(NUMSG)
FORMAT (F12.6, 17I4/(20I4)
```

TINT = a time within a MVP time interval over which the scope aggregation will performed. If TINT is inputted as zero, then all available time intervals will be aggregated for this score aggregation.

NUMSG = number of segments within this score aggregation.

NSV(K) = segments which are to be included in this score aggregate. For a given score aggregation, the various segments should be unique.

Major Card Group B
MVP Field Data and Statistical Test Options

Major Card Group B consists of the statistics of the observed field data and various statistical option cards for any or all of the variables (state variables or otherwise) in the user's WASP model. These cards, which are read for each system (from 1 through NOSYS) in the model include the WASPB variable names, additional aggregate weighting vectors, observed field data, and user determined options to perform the three available statistical procedures. The following is a card description of Card Group B.

BA. System Card

For each system of the NOSYS systems incorporated in the WASP simulation, a system card must be inputted. This card has the character string '*SYSTEM' in columns 1-8, optionally followed by an integer denoting the current system. WASP systems are inputted in increasing numerical order from 1 to NOSYS. The system-card must be supplied whether or not the particular systems is bypassed. However, the system card is the only input card which must always be supplied. All other input cards are optional under certain circumstances.

```
      4       10
*SYSTEM     ISYS
FORMAT (2A4, I2)
```

BB. Variable Name Card

This input card must always be supplied unless the system bypass option for this system is set on (i.e., SYSBY = 1)

```
      8          16
NAME(1)   NAME(2)    ....    NAME(MXDMP)
FORMAT (10A8)
```

The user supplies MVP with names for the MXDMP variables of the current system.

NAME(K) = name of the kth variable in the ISYSth system dump file. The user should refer to the appropriate WRITE statement in the WASPB kinetic subroutine.

BC. Score Aggregate Weighting Card(s)

For a discussion of score aggregation weighting vectors refer to the MVP Theory section. NOAGG score aggregate weighting cards must be supplied, here, in each non-bypassed system. If NOAGG has been inputted as zero, bypass this section. The first score aggregate weighting card is utilized in computing score aggregate number one, the second card for score aggregate two, etc., up to and including the NOAGGth card.

8	16		
VARWT(1)	VARWT(2)	....	VARWT(MXDMP)

FORMAT (10F8.4)

The ith card, where $1 \leq i \leq NOAGG$, should contain MXDMP weights. If the user does not want to include the jth variable of the current system in the ith score aggregate, a zero should be specified for VARWT(J) on the ith score aggregate weighting card. If the user desires equal weights for all variables for the ith score aggregate, a 1. should be specified for all the variables from the current system that the user wishes to include in the overall score.

Note: Since any of the variables in any of the systems may be included (weighted or unbiased) in the score aggregate computation, the score aggregate results are printed at the end of an MVP run.

BD. Field Data Card(s)

The user must now supply the observed field station data to the MVP program for the current system. For each field station, the user must supply MVP with two types of cards: (1) a Variable-Segment Card, and (2) one or more Data Cards.

1. Variable-Segment Card

4	8
NOVAR	NUMSG

FORMAT (2I4)

NOVAR = The variable number within the current system for which the following data was collected. NOVAR may range from 1 to MXDMP.

NUMSG = The segment number within the WASP model for which the following field data was collected. NUMSG may range from 1 to NOSEG.

Note: A blank Variable-Segment Card signals the end of field data for the current system to the MVP programs.

2. Data Card

10	20	30	40	50	60	70	80
T(1)	X(1)	S(1)	N(1)	T(2)	X(2)	S(2)	N(2)

FORMAT (2(3F10.5, I10)

T(1) = time data was collected for NOVAR variable, NVMSG segment, for current system.

X(1) = mean of the collected data, in same units as WASP output results.

S(1) = standard deviation of the sample data.

N(1) = number of observations in this sample.

There is only one reduction on the input of field data. Data must be entered in increasing time order. Otherwise execution of the MVP analyses run will be terminated. Within one system, a Variable-Segment may be repeated, but only the latest entry will be used by the MVP program, all previous entries with the same Variable-Segment pair within one system will be overridden.

Note: A blank time entry (with blank associated data) signals the end of field data for the current Variable-Segment Card. MVP will then expect a new Variable-Segment Card immediately following the end of the data entries for the last Variable-Segment Card. Two blank cards in a row terminates the input of field data for the current system.

Statistical Performance Option Cards

Minor Card Groups BE through BG initiate the performance of MVP statistical tests on the inputted field data and WASP results for the system under current analyses. Each of these card groups has a title card which MVP recognizes in order to implement the appropriate statistical test. Within each system, the user has the option of specifying any, all, or none of the statistical tests. The only restriction is that the user may specify a statistical test only once within each system. Otherwise the MVP run will be aborted without further analyses being performed.

BE. T-Test Card(s)

To initiate the Student's T-Test, the user must input the character string '*T-TEST' in columns 1-7 of the title card for this statistical group. Then the single format for the control cards in this group are as follows:

```
     4       8       12        16        20
NOVAR   IWGT   NUMSG   NSV(1)   NSV(2)   ....   NSV(NUMSG)
FORMAT (20I4)
```

NOVAR = variable within current system, on which T-Test is to be performed.

IWGT = weighting vector #1, #2 or #3 that the user has previously specified, that is to be used for any spatial aggregations. IWGT must always be specified.

NUMSG = 1, for T-Test of one WASP segment results versus a particular field station data that was inputted in the data section of this system.

> 1, for spatial or segmental aggregation, i.e., the number of segments to be aggregated.

NSV(I) = segment number or numbers (segment aggregation) which will be contained in this T-Test. The number of segments specified must be equal to NUMSG. If segmented aggregation is specified, it is permissible for more than one (even all) of the segments to have associated inputted field data.

To exit from the T-Test process, input a blank T-Test control card.

BF. Linear Regression Card(s)

To initiate Linear Regression, the user must input the character string '*REGRESS' in columns 1-8 of the title card for this statistical group. To exit from the Linear Regression process, input a blank card.

A single primary linear regression control card has the following format and may or may not be followed by secondary control cards depending on the options chosen by the user on the primary control card.

1. Primary Control Card

```
1     19   22  25  28  31   43  44     47   50  53
OPTION1     OPT2    OPT3    NOVAR    IWGT    OPT4
FORMAT (A19, 2X, A4, 2X, A4, 11X, I2, 2X, I1, 2X, A4)
```

OPTION1 = 'TIME POINTS BY SEGM', this option indicates that the linear regression graphs under consideration will consist of one or more WASP model segments. The points for the linear regression will have their y and x axis values calculated respectively from station field data and WASP theoretical data where MVP time intervals define the unique points on each graph.

= 'SEGM POINTS BY TIME', this option indicates that the linear regression graph(s) will consist of one or more MVP time intervals as defined on the Print-Interval-Aggregate Card. The points for the linear regression will have then, y and x axis values calculated respectively from station field data and WASP theoretical data where WASP model segments define the unique points on each graph.

OPT2 = 'ALLS': associated with the points of the linear regression, either TIME or SEGM (segment) points. This option indicates all available points will be plotted separately. For example, if the WASP model contained 25 segments and OPTION1 = 'SEGM POINTS BY TIME', was chosen by user than there would be the possibility of 25 points on the linear regression graph.

= 'SPEC' points, either time or segments, will be inputted on secondary control cards. Each time or segment will generate a separate point on the linear regression graph.

= 'COMB' points will be inputted on secondary control card. In this case each plotted point may be composed of more than one MVP time interval or WASP segment.

OPT3 = 'ALLS' associated with the option of specifying more than one linear regression graph per primary control card. For example, if the user has specified 11 time intervals and OPTION1 = 'SEGM POINTS BY TIME', then 11 graphs will be produced, one for each time interval.

= 'ALLC' all possible time or segment graphs, depending on OPTION1 will be combined into one graph.

= 'SPEC' the user will input a secondary control card to specify which time or segment graphs will be generated.

= 'COMB' same as SPEC except that user may combine specified graphs into one graph.

NOVAR = The variable within current system, for which the current linear regression is to be performed. The character string 'VARIABLE' may be input in columns 34-42 as a mnemonic device to assist the user in reading or editing his input data deck.

IWGT = the number of the weighting vector to be utilized for segmental aggregation for this linear regression control card.

OPT4 = 'PLOT', plot all graph(s) associated with this primary control card.

2. Secondary Control Card(s)

If the 'SPEC' or 'COMB' options are inputted by the user for OPT2 and/-or OPT3, it becomes necessary for the user to supply the MVP program with further control information in the form of secondary control card(s). In all cases, any secondary control cards required by OPT2 are inputted before any secondary control cards required by OPT3.

The 'SPEC' option for either OPT2 or OPT3 has one of the two following formats, depending on whether 'SPEC' is associated with segments or time intervals.

A. Segments

4	8	12		
NUMSG	NSV(1)	NSV(2)	....	NSV(NUMSG)

FORMAT (20I4)

NUMSG = number of segments in the 'SPEC' options.

NSV(1) = the segments to be included within the 'SPEC' options.

B. Time Intervals

4	11 20	21 30		
NUMSG	NIV(1)	NIV(2)	....	NIV(NUMSG)

FORMAT (I4, 6X, 7F10.5)

NUMSG = the number of times in the 'SPEC' option.

NIV(1) = the time intervals to be included within the 'SPEC' option. For each MVP time interval wanted in the 'SPEC' option, a time within that interval should be specified in NIV(I).

The 'COMB' option for either OPT2 or OPT3 has one of the following two formats depending upon whether 'COMB' is associated with segments or with time intervals.

NUMSG = the number of times in the 'SPEC' option.

NIV(1) = the time intervals to be included within the 'SPEC' option. For each MVP time interval wanted in the 'SPEC' option, a time within that interval should be specified in NIV(1).

The 'COMB' option for either OPT2 or OPT3 has one of the following two formats depending upon whether 'COMB' is associated with segments or with time intervals.

A. Segments

Card 1:

```
     4
NOCOB
FORMAT (I4)
```

NOCOB = number of combinations, i.e., number of graph(s) or point(s). Each graph or point may be composed of one or more WASP segments.

Cards 2 through NOCOB + 1:

```
     4       8        12
NUMSG   NSV(1)   NSV(2)    ....    NSV(NUMSG)
FORMAT (20I4)
```

NUMSG = number of segments within the ith combination.

NSV(1) = segments within the ith combination. There must be NVSMG SEGMENTS in this combination.

B. Time Intervals

Card 1:

```
     4
NOCOB
FORMAT (I4)
```

NOCOB = number of combinations, i.e., number of graphs or points. Each graph or point may be composed of one or more MVP time intervals.

Cards 2 through NOCOB + 1.

```
     4    11   20    31   30
NUSMG   NIV(1)   NIV(2)    ....    NIV(NUMSG)
FORMAT (I4, 6X, 7F10.5)
```

NUMSG = number of MVP time intervals within the ith combination.

NIV(1) = unique time within a unique MVP time interval within 'COMB' option. For each MVP time interval wanted in the ith combination, the user should include a time within that time interval.

NOTE: Within any primary linear regression control card, no WASP segment or MVP time interval should be specified more than once.

Example of time interval specification:

Assume that the Print Interval-Aggregate Card (Card AC) supplied was as: 0.0 360.0 3 so that the year is broken into 12 monthly average. To include January and February in a 'SPEC' option or in the ith combination of a 'COMB' option, the user should input the secondary control card as

4	11	20	21	30
2		T_1		T_2

where: T_1: $0. \leq T_1 \leq 30.$

T_2: $30. \leq T_2 \leq 60.$

Any values for T_1 and T_2 which satisfy these constraints are valid. We recommend a simple approach by splitting the desired time interval in half. Then

$T_1 = 15.$ and $T_2 = 45.$

BG: Relative Error Card(s)

To initiate Relative Error calculations, the user must input the character string '*REL ERR' in columns 1-8 of the title card for this statistical group. There are two different types of relative error control card: primary and secondary. For each relative error calculation performed there is one and only one primary control card. However, there is at least one and maybe more than one secondary control card.

1. Primary Control Card

4	8	12	15	18
NOVAR	IWGT	NOCOB		POPT

FORMAT (I4, I4, I4, 2X, A4)

NOVAR = Variable within the current system over which the Relative Error calculation is to be performed. Relative error calculations will be performed over each MVP time interval for which the user has specified field data.

IWGT = The number of the weighting vector that the user wishes to use with any segmental aggregation occurring within the relative error calculation.

NOCOB = number of relative error calculations that will be performed in creating a relative error distribution, (also determines the number of secondary control cards).

POPT = 'PLOT' the user should specify this option if he desires a one page plot of the distributive and cumulative relative error distribution for the NOVAR variable in current system over all MVP time interval for which the user has input field data.

= 'NPLT' user does not wish a plot of the relative error distributions.

= '   ' same as 'NPLT'

NOTE: To exit from the Relative Error process, the user should input a blank primary control card.

2. Secondary Control Card

There are NOCOB secondary control cards for each primary control card.

4	8	12		80
NUMSG	NSV(1)	NSV(2)	....	NSV(NUMSG)

FORMAT (20I4)

NUMSG = number of segments to be included in the ith relative error calculation for the NOVAR variable of the current system.

NSV(1) = WASP segments to be included in the ith relative error calculation which $1 \leq i \leq NOCOB$. Segments numbers should be unique, i.e., unduplicated within the ith relative error calculation as well as within the primary control card.

This concludes the description of card input formats for the MVP program.

APPENDIX A

WASP LISTINGS

Appendix A presents supporting documentation for the Lake Ontario eutrophication model, LAKE1. The supporting material includes:

1. A FORTRAN IV compilation listing of the LAKE1 kinetic subroutine, with the appropriate code for generating the dump and plot files for the IBM 370 and DEC PDP computer systems.

2. The WASP input data for running the LAKE1 model, with card group identifiers for following the input structure.

3. The output results, including printer and pen plots, from the WASP - LAKE1 model.

4. The Job Control Language (JCL) for executing WASP on the IBM 370 series computers (JCL as used on the EPA - COMNET system).

5. The task builder and overlay command files needed to taskbuild or link-edit WASP on the DEC PDP computer systems (for a DEC PDP 11/45 running under the RSX-11D operating system).

No attempt has been made here to explain the structure of the LAKE1 model or its associated equations. For this the user is asked to refer to the EPA Report, "Mathematical Modeling of Phytoplankton in Lake Ontario", EPA-660/3-75-005, March 1975.

Finally it should be noted that although the LAKE1 model input data set and kinetic subroutine specify the model to have 10 systems, in actuality the model used only has 8 systems. The last 2 systems, representing the upper trophic levels of zooplankton, were bypassed.

The listings are availabe through the National Technical Information Service, Springfield, Virginia 22161.

APPENDIX B

MVP LISTINGS

Appendix B provides supporting documentation for the verification analysis of the LAKE1 model using MVP. The supporting material includes:

1. The MVP input data for performing a verification analysis on the LAKE1 model. The input data listing has card group identifiers for following the input structure.

2. The output results of the verification analysis.

3. The JCL needed for executing MVP on the IBM 370 series computer (JCL as used on the EPA - COMNET system).

4. The taskbuilding and overlay command files needed to taskbuild or link-edit MVP on the DEC PDP computer systems (for a DEC PDP 11/45 running under the TSX-11D operating system).

The listings are available through the National Technical Information Service, Springfield, Virginia 22161.

APPENDIX C

MAJOR PROGRAM MODIFICATIONS
(DEC PDP-11 VERSION)

For those modelers and programmers who maybe confronted with problems requiring system/segment combinations different than those made available by their current version, the dimensions of the appropriate arrays and direct access files must be changed accordingly. As a guide in accomplishing this, Tables C-1 and C-2 have been developed.

When changing the dimensions of any of the arrays which appear in common the entire program must be recompiled since each common block must be changed identically. All direct access file changes should be either in subroutine FILEOC or in WASPMAIN.

The modeler/programmer should note that system/segment configurations are not the only model parameters which require changes in code. Included below is a list of common changes and the program modifications needed for each.

1. A change in input data such as number of flows, exchanges, etc. may exceed the current storage capacity of arrays and/or direct access files listed in Table C-1.

2. A longer simulation time or a shorter print interval may cause the amount of output to exceed the size of the direct access files listed in Table C-2.

3. When running WASP with MVP the following files must be defined with the same attributes in both programs, i.e., record length, number of records, etc.,

   F10.XXZ: (FILEOC in WASP, MVPMAIN IN MVP)

   all other files which occur both in FILEOC and LOADR.

Future updates of this documentation will include more explicit detail according to responses from the user community.

TABLE C-1. WASP SCALAR VARIABLES - DIRECT ACCESS WORK FILES

Variable	Definition	Account [55,105] value (range)	Account [55,116] value (range)	FILE X (Y, Z, U, POINTER = IXYZ) X=File Identifier, Y=No. of Records, Z=File Record Length DIRECT ACCESS FILE IN SUBROUTINE FILEOC
NOV	Number of volumes read. In general NOV=NOSEG.	1→60	SAME	(NOV+1) is the required number of records (Y) in file X=71.
NOR	Number of exchange coefficients read.	READ	READ	(NOR+1) is the required number of records in file X=70.
NO (Subroutine WASP 6)	In Subroutine 5 NO is the number of boundary concentrations for a particular system, (NO=NOBC (ISYS)).	1→NBCPSY	1→NBCPSY	(NO+1)·(NOSYS) is the required number of records in file X=30. For flexibility Y should be (NBCPSY+1) (NOSYS) to avoid change.
NO (Subroutine WASP 6)	In Subroutine 6 NO is the number of forcing functions for a particular system. (NO-NOWK(ISYS)).	1→NWKPSY	1→NWKPSY	(NO+1)·(NOSYS) is the required number of records in file X=50. For flexibility Y should be (NBCPSY+1) (NOSYS) or (NWKPSY+1) (MXSYS).
NOQ	Number of flows read.	READ	READ	(NOQ+1) is the required number of records in file X=72.
NOSYS	Number of systems the program is to be run with.	1→5	1→16	(On [55,105] only). NOSYS is the number of records required for file X=80. In general should have Y=MXSYS.

TABLE C-1. (CONT.)

Variable	Definition	Account [55,105] value (range)	Account [55,116] value (range)	FILE X (Y, Z, U, POINTER = IXYZ) X=File Identifier, Y=No. of Records, Z=File Record Length DIRECT ACCESS FILE IN SUBROUTINE FILEOC
IREC	Number of print intervals that output has been saved for.	1-NOBRK	SAME	Final value of IREC (largest value) is the number of records required for file X=10. In general should have Y=NOBRK (MAX).
NFUNC	Number of time variable functions for use in WASPB.	READ	READ	(NFUWC+1) is the number of records required for file X=73.
NBCPSY	Maximum number of boundary concentrations per system.	15	5	See NO (Subroutine WASP 5).
NWKPSY	Maximum number of forcing functions per system (Wth's).	35	15	See NO (Subroutine WASP 6).
MXSYS	Maximum number of systems which can be run for particular version.	5	16	See NOSYS.
MXSEG	Maximum number of segments.	60	20	
NOBRK*				

TABLE C-1. (CONT.)

Variable	Definition	Account [55,105] value (range)	Account [55,116] value (range)	FILE X (Y, Z, U, POINTER = IXYZ) X=File Identifier, Y=No. of Records, Z=File Record Length DIRECT ACCESS FILE IN SUBROUTINE FILEOC
(WASP 2)	Number of breaks used to describe the exchange coefficients.	READ	READ	2·(NOBRK+1) is the record length (Z) in words required for file X=70.
(WASP 3)	Number of breaks used to describe piecewise linear volume function (VT).	READ	READ	2·(NOBRK+1) is the record length (Z) in words required for file X=71.
(WASP 4)	Number of breaks used to describe the approximation for flow function.	READ	READ	2·(NOBRK+1) is the record length (Z) in words required for file X=72.
(WASP 5)	Number of breaks used to describe the approximation for bound. conc. function.	READ	READ	2·(NOBRK+1) is the record length (Z) in words required for file X=30.
(WASP 6)	Number of breaks used to describe the approximation for the forcing functions.	READ	READ	2·(NOBRK+1) is the record length (Z) in words required for file X=50.
(WASP 7)	Number of breaks used to describe the approximation for the kinetic functions.	READ	READ	2·(NOBRK+1) is the record length (Z) in words required for file X=73.

TABLE C-1. (CONT.)

Variable	Definition	Account [55,105] value (range)	Account [55,116] value (range)	FILE X (Y, Z, U, POINTER = IXYZ) X=File Identifier, Y=No. of Records, Z=File Record Length DIRECT ACCESS FILE IN SUBROUTINE FILEOC
(WASP 11)	Number of integration steps, i.e., number of T(I)'s.	READ	READ	2•(2NOBRK+1) is the record length (Z) in words required for file X=75.
PRNT	The interval at which model output is to be generated (in days).			File numbers 11 through 10 + NOSYS must be changed so that $y \geq \frac{TOTIM}{PRNT} * (NOSEG)$ File 10 must have $y \geq$ TOTIM/PRINT
TEND	Total simulation time.			File 54 should be changed according to the desired number of points to be plotted and subroutine OUTPUT changed accordingly.

*The value of NOBRK is not in WSPCMN, it is assigned different values in different subroutines (different functions).

TABLE C-2. WASP ARRAYS

Array	Definition	(Maximum) [55,105] DIM.	(Maximum) [55,116] DIM.	Related Arrays, Scalar Variables, Dir. Acc. Files
NBCT,NWKT	Time at which the next break in the function, b.c. or WK respectively, will occur.	(5)	(16)	In general dimension of NBCT, NWKT is = (MXSYS).
SYSBY, RBY,QBY	System, exchange, or flow bypass option.	(5)	(16)	In general dimension of SYSBY, RBY, QBY = (MXSYS).
MVOL,MR, MQ	"m" value in straight line eq. for volume, exchange, flow of.*	MVOL(60) MR,MQ(120)	MVOL(60) MR,MQ(40)	In general dimension of MVOL = (MXSEG). In general dimension of MR, MQ = 2•(MXSEG).
MBC,MWK	"m" value in straight line eq. for bound. conc., forcing functions respectively.	(5,15) (5,35)	(16,5) (16,5)	In general dimension of MBC = (MXSYS, NBCPSY). In general dimension of MWK = (MXSYS, NWKPSY).
IDFRC	For a particular file (associated with the subscript) the number of the current record.	(5)	(16)	In general dimension of IDFRC = (MXSYS).
C,CD	Concentration and its derivative, respectively.	(5,60)	(16,20)	In general dimension of C, CD = (MXSYS, MXSEG).

TABLE C-2. (CONT.)

Array	Definition	(Maximum) [55,105] DIM.	(Maximum) [55,116] DIM.	Related Arrays, Scalar Variables, Dir. Acc. Files
CMAX, CMIN	Maximum, and minimum allowable concentrations.	(5)	(16)	In general dimension of CMAX, CMIN = (MXSYS).
PARAM	Value of parameter ANAME(K) for a particular segment.	(60,6)	(20,10)	In general dimension of PARAM = (MXSEG, NOPAM).
MFUNC, BFUNC	Slopes, intercepts for kinetic functions.	(10)	(10)	In general dimension of MFUNC, BFUNC = (NFUNC) (READ).
CONST	Value of constant ANAME(K).	(75)	(75)	In general dimension of CONST = (NCONS).

☆U.S. GOVERNMENT PRINTING OFFICE: 1983/605-095/1935