

**Implementation of the IWA River Water Quality Model No. 1 in US EPA
WASP 5.0**

by

Arjun Ayillath Nair

S.B., Environmental Engineering Science
Massachusetts Institute of Technology
2001

Submitted to the Department of Civil and Environmental Engineering
in Partial Fulfillment of the Requirements for the Degree of

Master of Engineering
in Civil and Environmental Engineering

at the

Massachusetts Institute of Technology
June 2002

© 2002 Arjun A. Nair. All rights reserved

The author hereby grants to MIT permission to reproduce
and to distribute publicly paper and electronic
copies of this thesis document in whole or in part

Signature of Author.....

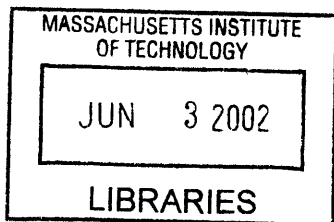
Department of Civil and Environmental Engineering
May 10, 2002

Certified by.....

Peter Shanahan
Lecturer, Department of Civil and Environmental Engineering
Thesis Supervisor

Accepted by.....

Oral Buyukozturk
Chairman, Departmental Committee on Graduate Studies



BARKER

Implementation of the IAW River Water Quality Model No. 1 in US EPA WASP 5.0

Arjun Ayillath Nair

Submitted to the Department of Civil and Environmental Engineering on May 10, 2002
in Partial Fulfillment of the Requirements for the Degree of Master of Engineering
in Civil and Environmental Engineering

Abstract

The history of water quality modeling dates back to the pioneering work of Earle Streeter and Harold Phelps (1925), who described the oxygen deficit in the Ohio River as function of time for a steady source point discharge of pollutant. The development of the Streeter-Phelps model marks the beginning of a period of intense study to expand the model to reflect an increased knowledge of the real system. However, over time these models came to incorporate a number of inconsistencies, reflecting a historic development wherein model layers lacking a common unifying principle were added one by one.

These inconsistencies in model formulation prompted the relatively recent development of the River Water Quality Model No. 1, as “a scientific and technical base from which to formulate standardized, consistent river water quality models and guidelines for their implementation” (Reichert et al., 2001a; Reichert et al., 2001b; Vanrolleghem et al., 2001; Shanahan et al., 2001, Rauch, 1998; Shanahan, 1998; Somlyódy, 1998). Many of the deficiencies in the formulation of current models of traditional water quality pollutants were seen in the description of the conversion submodels. Ultimately, the deficiencies were seen to stem from the lack of a uniform underlying concept in the definition of state variables. Three different conceptual levels were noted: (1) phenomenological (e.g. DO and BOD), (2) biochemical (i.e. nitrogen cycle constituents), and (3) ecological (i.e. algae). To resolve this problem in RWQM1, the definitions for organic constituents are based on elemental mass fractions. From this science-based definition of organic constituents, the effects of biogeochemical transformation processes can be related to all constituents through the elemental mass fractions and other stoichiometric parameters.

The present study examines the implementation of the RWQM1 conversion submodel into an existing simulation program called WASP for Water Quality Analysis Simulation Program. EPA’s WASP 5.0 is a dynamic compartment transport model that has been used in conjunction with modular conversion submodels to predict the response of aquatic systems to pollutant loads. The modular structure of WASP allows a relatively straightforward implementation of user-written conversion subroutines into the existing transport framework. Chapters 2 and 3 discuss such and implementation of RWQM1 into WASP 5.0, while Chapter 4 treats the question of calibration and verification of site-specific implementations of the resulting simulation program entitled RWQMWASP.

Thesis Supervisor: Peter Shanahan

Title: Lecturer, Department of Civil and Environmental Engineering

Acknowledgements

I am indebted to Dr. Peter Shanahan for his unwavering commitment to this project. His words of encouragement were for me a breath of new life when, mired in an endless sea of details, my vision of the goal was obscured.

I would like to thank my fellow Master of Engineering students for journeying with me and fighting by my side against that faceless and unrelenting enemy Time, whose attacks against concentration and commitment came blow after blow, and yet each time met with disciplined resistance. May the Force be with you all.

To my family. You have stood by and supported me through the most difficult years of my life. I thank you and I love you. To God, my Liege, my Master.

Table of Contents

Abstract	2
Acknowledgements	3
Table of Contents	4
List of Figures	6
List of Tables.....	7
1 Introduction to Water Quality Modeling.....	8
1.1 Historical Roots of River Water Quality Modeling	8
1.2 Introducing River Water Quality Model No. 1	10
1.3 Inconsistencies in Model Formulation	10
1.3.1 Definition of Water Quality Constituents	11
1.3.2 State Variables for Bacteria.....	15
1.3.3 Conservation of Mass.....	16
1.3.4 Integration of Models	17
2 Understanding the WASP Framework.....	18
2.1 Common Blocks.....	19
2.2 Input Subroutines	21
2.2.1 WASP1.....	22
2.2.2 WASP2.....	24
2.2.3 WASP3.....	24
2.2.4 WASP4.....	25
2.2.5 WASP5.....	29
2.2.6 WASP6.....	30
2.2.7 WAS6A	31
2.2.8 WASP7.....	31
2.2.9 WASP9.....	34
2.3 Process Subroutines.....	36
2.3.1 EULER	36
2.3.2 DERIV.....	38
2.4 Kinetic Subroutines	41
2.4.1 WASPB	42
2.4.2 EUTROINT	42
2.4.3 Segment Loop	43
2.4.4 Mass Derivatives	44
2.4.4.1 PHYTO.....	45
2.4.4.2 DISSOXYG.....	48
2.4.5 EUTRODMP	51
2.5 Conclusion.....	52
3 Development of RWQMWASP	54
3.1 Outline for Coding a Conversion Submodel	54
3.2 Kinetic Structure	55
3.3 Number Assignments for System Variables	57
3.4 Number Assignments for Constants.....	58
3.5 Developing the Model	59

3.5.1	The RWQM Common Block	59
3.5.2	STOICHP and STOICHV	63
3.5.3	PROCESS	67
3.5.4	Mass Derivatives and SO2	72
3.5.5	Modifications to WASP	76
3.5.6	RWQMINT and RWQMDMP	77
3.6	Conclusion.....	80
4	Future Work: Model Verification and Parameter Identifiability	82
4.1	Model Parameter Estimation.....	83
4.2	Large, Overparameterized Models.....	84
4.3	Conclusion.....	86
5	References	88

List of Figures

Figure 1. Opening the Input File and Reading Record 4 of Data Group A in WASP1	20
Figure 2. Euler Mass Derivative Scheme in EULER subroutine.....	20
Figure 3. Sequence of subroutine calls in WASP4V	22
Figure 4. Initializing Simulation Arrays in WASP1	23
Figure 5. Scaling Factor calculation in subroutine SCALP	25
Figure 6. Input of advective flows for each transport field in WASP4.....	26
Figure 7. Reading in the continuity array in WASP4	27
Figure 8. Reading in piecewise linear time functions for advective flows in WASP4.....	28
Figure 9. Reading in piecewise linear time functions for exchange flows in WASP2	28
Figure 10. Mass balance check for advective flows (FIELD1) in WASP4	29
Figure 11. Reading in of point-source loading time functions in WASP6	31
Figure 12. Data Group I Kinetic Time Function Input in WASP7	33
Figure 13. Initial Concentrations and Dissolved Fractions in WASP9.....	35
Figure 14. Initialization of the array F in WASP1	35
Figure 15. Calculation of Particulate Fractions in WASP9	36
Figure 16. Euler Mass Derivative Scheme in EULER subroutine.....	37
Figure 17. Calculation of mass derivative at final time-step.....	37
Figure 18. Updating time function for flows in WAS8B.....	38
Figure 19. Updating kinetic time functions in WASP8	39
Figure 20. Updating kinetic time functions in WASPB.....	40
Figure 21. Updating forcing functions in WAS8A	41
Figure 22. Adjustment to mass derivative due to PS loads in WAS12.....	41
Figure 23. Initialization of kinetic parameters in EUTROINT	43
Figure 24. Conversion of phytoplankton initial and boundary conditions.....	43
Figure 25. Calculation of average solar radiation in WASPB segment loop.....	44
Figure 26. System name assignments are made from the concentration array	44
Figure 27. Di Toro et al. (1971) Light Formulation.....	46
Figure 28. Summary of PHYTO subroutine	47
Figure 29. Calculation of flow-induced reaeration in subroutine KAHYDRA	50
Figure 30. Calculation of oxygen exchange across the air-water interface in DISSOXYG	51
Figure 31. Calculation of total nitrogen and phosphorous in EUTRODMP	52
Figure 32. Common group for stoichiometric coefficients in RWQM.CMN.....	60
Figure 33. Common group for kinetic process rates in RWQM.CMN.....	61
Figure 34. Common group for additional kinetics submodel variables in RWQM1	62
Figure 35. Equivalence statements for constants grouped under Algae	62
Figure 36. Environmental parameter equivalence statements in RWQM.CMN.....	63
Figure 37. Calculation of stoichiometric coefficients for Nitrite in STOICHP	64
Figure 38. Often repeated terms in expressions for stoichiometric coefficients calculated in STOICHV.....	65
Figure 39. Assignments for Organic Matter to COD conversion factors.....	66
Figure 40. Calling sequence of mass derivative subroutines in EUTRO.....	68
Figure 41. Calculation of processes 1a and 1b in subroutine PROCESS	70
Figure 42. Calculation of source and sink rates in SO2.....	72

Figure 43. System bypass option for dissolved oxygen	73
Figure 44. Dissolved oxygen mass derivative calculation in SO2.....	76
Figure 45. Changes in WASP.CMN to increase the number of allowable systems and kinetic constants	77
Figure 46. An error message is called if user-specified NOSYS is greater than SY	77
Figure 47. Initializations in RWQMINT.....	79
Figure 48. Display variables in RWQMWASP	80
Figure 49. Display variables are printed to the dump file in RWMDMP	80

List of Tables

Table 1. Elemental Mass Fractions for Organic Compounds (Reichert et al., 2001a).....	12
Table 2. Solids Types for System Particulate Fractions	34
Table 3. Number assignments for system variables in RWQM1	57
Table 4. Terms whose elemental mass fractions must be specified.....	74

1 Introduction to Water Quality Modeling

1.1 Historical Roots of River Water Quality Modeling

Mathematical models of river water quality were first developed by sanitary engineers motivated by the increasing need to control pollution of the freshwater supply and to protect the public health. It was from early studies of the response of rivers to nutrient loads that the pioneering work of Harold Streeter and Earle Phelps emerged. Formulated as a first-order ordinary differential equation, and integrated to provide a closed-form solution for oxygen deficit as a function of time for a point discharge of a steady source pollutant, the Streeter-Phelps equation recognized two major processes controlling the level of dissolved oxygen in a river: (1) the consumption of oxygen by biochemical oxidation of organic matter (deoxygenation), and (2) mass transfer of oxygen across the air-water interface due to turbulent mixing processes (reaeration) (Streeter and Phelps, 1925).

The development of the deterministic Streeter-Phelps model marks the beginning of period of intensive study, during which investigators sought to determine laboratory and field biodegradation and reaeration coefficients and to find more elegant analytical solutions to the Streeter-Phelps differential equation modified to reflect an increased knowledge of the real system. By then, changing public attitudes toward resource management, and the substantial investment required not just for reducing contaminant discharge, but also by limiting economic development in rivers, necessitated the formulation of extensive water quality management plans or environmental control strategies (Gromiec et al., 1983). The fear of reducing the waste load to a river or requiring excessively costly controls, and then observing little or no improvement in water quality prompted the development of these more complex water quality models. However, the requirement of closed-form analytical solutions limited the applicability of the Streeter-Phelps model, since the refinement of the model to include many cases deemed necessary for an accurate description of a water body produced intractable problems. It was not until the advent of modern computational tools that numerical methods for solving previously intractable problems were implemented (Orlob, 1983).

The first major set of refinements to the two-state-variable Streeter-Phelps model was limited by the requirement of an analytical solution. These include the introduction of (1) a particulate settling rate to model the reduction of biochemical oxygen demand (BOD) in the

water column by sedimentation, (2) a parameter for sediment oxygen demand (SOD) to account for the decomposition of organic matter in the sediments, and the respiration of plants or attached filamentous bacteria in the aerobic zone of the sediments, (3) photosynthesis and respiration, and finally (4) a third state variable introducing nitrogenous biological oxygen demand (NBOD) as distinct from carbonaceous biological oxygen demand (CBOD). (Thomas, 1948; Dobbins, 1964; O'Connor and Di Toro, 1968, 1970).

The second major refinement was the extension of the three-state-variable model to include a simplified nitrogen cycle, which appeared first in QUAL1. The incorporation of the three main nitrogen species—nitrate-nitrogen, ammonia-nitrogen and nitrite-nitrogen—into this numerical model increased the number of state variables to five (TWDB, 1971).

The third and final major refinement in river water quality modeling occurred ten years after QUAL1, with the development of QUAL2, followed a number of years later by QUAL2E (Roesner et al., 1981; Brown and Barnwell, 1987). The most widely used river water quality model to date, QUAL2 further increased the number of water quality constituents to ten¹, adding organic nitrogen, organic phosphorous, dissolved phosphorous and algae biomass measured by chlorophyll-a as water quality components. Additionally, since all biological reactions are strongly temperature-dependent, a general temperature function was introduced. The resulting ecosystem dynamics model, designed specifically to predict the impact of nutrients on aquatic ecosystems in water bodies, considered the following processes: degradation of organic material; growth and respiration of algae; reaeration; sedimentation of algae; organic phosphorous and organic nitrogen; sediment oxygen uptake and nitrogen and phosphorous efflux; and nitrification considered in two stages with nitrite as the intermediate.

Being the most comprehensive stream water quality model developed by the EPA, QUAL2E is the standard in river water quality modeling today, despite having seen very few improvements since 1987. However, certain inconsistencies in the model formulation, reflecting the historical development of water quality models by piecewise addition of model layers lacking a single unified underlying concept, and a lack of rigorous science-based definitions for its water quality constituents, have quite recently prompted the development of the River Water Quality

¹ It should be noted that sediment oxygen demand is actually formulated as a boundary condition, not a separate water quality constituent.

Model No. 1 (RWQM1) as a “scientific and technical base from which to formulate standardized, consistent river water quality models and guidelines for their implementation” (Reichert et al., 2001a; Reichert et al., 2001b; Vanrolleghem et al., 2001; Shanahan et al., 2001, Rauch et al., 1998; Shanahan et al., 1998; Somlyódy, 1998).

1.2 Introducing River Water Quality Model No. 1

The development of RWQM1 by an International Water Association (IWA) Task Group on River Water Quality Modeling headed by László Somlyódy of Budapest University of Technology and Economics was motivated by a number of deficiencies in the state of the art of river water quality modeling. Many of these deficiencies were seen in the description of the conversion submodel for traditional water quality pollutants, which the Task Group has subsequently sought to refine. The result of their work was a suite of biogeochemical conversion submodels, which incorporate 24 water quality components and 30 processes² to characterize the cycling of C, O, N, and P in river systems. Also included are guidelines for selecting submodels appropriate for particular applications of the model. The deficiencies identified by the Task Group are considered separately in the following subsections, along with the means identified by the Task Group for resolving them.

1.3 Inconsistencies in Model Formulation

A number of limitations in current water quality models were identified by the Task Group. Ultimately, these stem from the lack of a uniform underlying concept due in part to the gradual historical evolution of river water quality models wherein additional model layers and state variables were acquired over time, leading to a lack of clear definitions for these state variables. State variables in traditional water quality models and the relevant processes affecting them fall into three groups reflecting three stages of model development (Somlyódy et al., 1998): (1) phenomenological, as in the case of DO and BOD, both based the results of an assay, (2) biochemical, as in the state variables of the simplified QUAL1 nitrogen cycle, which tries to model a series of oxidation-reduction reactions using a first-order kinetics structure, and (3)

² See Chapter 3 of Reichert et al. (2001) for a complete description of the conversion submodels.

ecological, as with algae and other ecosystem variables in QUAL2, whose dynamics try to account for the non-linear growth and decay relationships between organisms.

Because the definition of state variables lacks a unified underlying concept, current water quality models lack robustness and elegance. In RWQM1, state variable definitions are based on elemental mass fractions. Consequently, the processes converting organic matter into other constituents and vice versa can also be defined using these mass fractions. Moreover, since all organic constituent definitions are consistent with each other, artifacts resulting from the lack of a uniform underlying concept are avoided. To take a simple example, even though the process of hydrolysis is essentially the same for all organic matter, in QUAL2E the rates of hydrolysis are different for organic phosphorous and organic nitrogen because of how these variables are defined.

1.3.1 Definition of Water Quality Constituents

QUAL2 and similar models lack rigorous operational definitions for water quality constituents. For example, QUAL2 lumps all organic nitrogen species into a single state variable, even though many forms of organic nitrogen are known to exist (Somlyódy et al., 1998). Similarly, many river water quality models continue to use CBOD to characterize all oxidizable carbon in a sample. Like organic nitrogen, CBOD represents not a unique organic compound, but an aggregate of organic constituents. These constituents cannot be readily distinguished by the CBOD assay. Moreover, since the CBOD test assays for soluble oxidizable carbon, its yield will vary according to the history of the organic material, so that different fractions of the total organic material will be measured by the assay for different substrates (Reichert et al., 2001b and Reichert et al., 2001a).

As mentioned earlier, in RWQM1 the organic matter characterization is based instead on elemental mass fractions of organic compounds that are later expressed in terms of COD to allow integration with current water quality modeling practices. This allows a very rigorous science-based definition of water quality state variables. Moreover, a number of laboratory and analytic methods for measuring water quality constituents are discussed, along with calculations for converting between observed values and actual state variable concentrations (Reichert et al., 2001b and Reichert et al., 2001a).

Elemental analyses of organisms have been applied with increasing frequency in recent years. Environmental scientists have employed analytical techniques such as single-cell X-Ray Microanalysis, along with measurements of uptake and release of nutrients to assess the elemental compositions of a variety of organisms including bacteria and phytoplankton (Norland et al., 1995; Elser et al., 1995). The results of these studies have been used to describe RWQM1's expanded set of organic matter state variables, including biodegradable/inert dissolved and particulate organic matter, algae, consumers, heterotrophs and first and second stage nitrifiers. Table 1 presents literature-based numerical values for elemental mass fractions. The organic constituents listed across the top are as follows: dissolved organic substances (S_S), inert dissolved organic substances (S_I), heterotrophs (X_H), stage 1 nitrifiers (X_{N1}), stage 2 nitrifiers (X_{N2}), consumers (X_{CON}), particulate organic matter (X_S), and inert particulate organic matter (X_I).

Table 1. Elemental Mass Fractions for Organic Compounds (Reichert et al., 2001a)

	S_S	S_I	X_H	X_{N1}	X_{N2}	X_{ALG}	X_{CON}	X_S	X_I	Units
α_C	0.57	0.61	0.52	0.52	0.52	0.36	0.36	0.57	0.61	gC/gOM
α_H	0.08	0.07	0.08	0.08	0.08	0.07	0.07	0.08	0.07	gH/gOM
α_O	0.28	0.28	0.25	0.25	0.25	0.50	0.50	0.28	0.28	gO/gOM
α_N	0.06	0.03	0.12	0.12	0.12	0.06	0.06	0.06	0.03	gN/gOM
α_P	0.01	0.01	0.03	0.03	0.03	0.01	0.01	0.01	0.01	gP/gOM

The mass fractions for all elements besides C, H, O, N and P are neglected³, so that the following constraint is fulfilled for each constituent listed:

$$\alpha_C + \alpha_H + \alpha_O + \alpha_N + \alpha_P = 1 \quad (1.1)$$

where α_C = elemental mass fraction of carbon, α_H = elemental mass fraction of hydrogen, α_O = elemental mass fraction of oxygen, α_N = elemental mass fraction of nitrogen, and α_P = elemental mass fraction of phosphorous. The chemical formula for each of the listed components can then be written by converting the mass fractions to moles:

³ An additional mass fraction α_X is included for all elements other than C, H, O, N and P in the RWQM1 Technical Report. The updated chemical formula for organic matter and OM:COD conversion factor can be found in Reichert et al. (2001).

$$C_{\alpha_C/12} H_{\alpha_H/4} O_{\alpha_O/16} N_{\alpha_N/14} P_{\alpha_P/31} \quad (1.2)$$

For example, since 12 grams is the molar mass of carbon, the factor of 1/12 converts α_C from g C/g organic matter (OM) to moles C / g OM. By formulating the α 's this way, stoichiometric coefficients relating grams of reacting species can be defined. Thus, from mass conservation principles, the stoichiometric (molar) coefficient relating the mass of dissolved oxygen required to degrade organic matter up to a reference mineralization state can be calculated. The mineralization state is established by selecting reference compounds for each element (C, O, N, H and P). Reichert et al. (2001a) give the following relationship:

1 mole $C_{\alpha_C/12} H_{\alpha_H/4} O_{\alpha_O/16} N_{\alpha_N/14} P_{\alpha_P/31}$ combined with $(\alpha_C/12 + \alpha_H/4 - \alpha_O/32 - \alpha_N/56 + \alpha_P/31)$ moles O_2 will produce the reference state of mineralization established by the following compounds: HCO_3^- , H^+ , H_2O , NH_4^+ and HPO_4^{2-} (see Equation (4) of Reichert et al., 2001a)

From this relationship the following OM:COD conversion factor can be obtained for each of the listed water quality components.

$$g \text{ COD} = 32 (\alpha_C/12 + \alpha_H/4 - \alpha_O/32 - \alpha_N/56 + \alpha_P/31) g \text{ OM} \quad (1.3)$$

Thus, from elementary compositions introduced as model parameters, expressions are obtained for each constituent, which in turn can be related to a measurable quantity (to be discussed shortly). It should also be mentioned that all stoichiometric coefficients, which relate the effect of a general process on the concentration of a specific constituent, are also expressed as functions of these elemental composition parameters. This allows simplified adaptation of the model to situations where different compositional fractions seem appropriate. Without this element-based description of organic components, it would be impossible to change stoichiometric coefficients to reflect very different fundamental conditions. This said, the chief limitation of the RWQM1 element-based formalism is that the composition of organic compounds must remain both temporally and spatially constant for a particular case study. A varying composition would be required for example to model the "luxury" uptake of nutrients when non-limiting conditions are encountered. Luxury uptake yields higher fractions of N and P in algae.

As mentioned before, the division of organic material into different types on the basis of elementary mass fractions, introduces a number of water quality constituents whose riverine concentrations will be necessary both for calibration of the model and its application in predictive mode. Since we are able to express all organic matter water quality components in terms of COD, it seems possible to be able to relate these constituents to measurable quantities if total laboratory-measured COD can be separated into fractions corresponding to each water quality constituent. Of course, in the case of dissolved inorganic constituents, simple laboratory analyses permit very easy determination of their concentrations. These constituents include NH_3/NH_4 , NO_2 , NO_3 , $\text{HPO}_4/\text{H}_2\text{PO}_4$, O_2 , Ca^{2+} , and H^+ . For concentrations of organic constituents on the other hand, the approach taken is first to determine the total COD, and then to apply other laboratory techniques to split this total into fractions representing each of the organic water quality constituents.

The COD test measures the oxygen equivalent of all organic material measured by oxidation of organic matter in acid solution by dichromate (Metcalf and Eddy, 2001). COD, although similar to ultimate BOD (UBOD) in practice tends to be higher since (1) many organic substances that are biologically recalcitrant can be oxidized chemically and (2) dichromate oxidation of some inorganic substances increases the apparent organic content (Metcalf and Eddy, 2001). Total COD thus measures all of the water quality constituents listed in the mass fraction Table 1. This can be fractionated into particulate and dissolved species by simple filtration⁴. The filtrate contains only biodegradable and inert organic matter, together forming dissolved COD. The remainder includes all particulate organic species:

$$\text{COD}_{\text{diss}} = \text{COD of } (S_S + S_I) \quad (1.4)$$

$$\text{COD}_{\text{part}} = \text{COD of } (X_H + X_{\text{N1}} + X_{\text{N2}} + X_{\text{ALG}} + X_{\text{CON}} + X_S + X_I) \quad (1.5)$$

Of the dissolved COD fraction the biodegradable component can be determined via typical BOD analysis of the dissolved COD sample. Subtraction of this result from the total dissolved COD allows determination of the dissolved inert component. A similar experiment can

⁴ It should be noted that the definitions of soluble and particulate COD are issues of considerable debate. The threshold separating soluble and particulate fractions will vary depending on pore size of the filter (Metcalf and Eddy, 2001).

be performed on the total COD sample, and the combination of the two results allows determination of the inert particulate fraction. (Lesouef et al., 1992).

The separation of particulate organic matter into fractions corresponding to water quality components is nontrivial, employing techniques such as respirometric analysis. Respirometry measures the biological oxygen consumption rate under well-defined experimental conditions (Spanjers and Vanrolleghem, 1995). A typical respirogram can be separated into regions corresponding to the oxidation of particular organic substrates, so that the full area under the curve represents the total of all biodegradable components. Initial concentrations for each substrate can be estimated by integrating the equation that relates respiration of the substrate to the stoichiometric parameters needed to characterize its oxidative yield. The initial concentration is then varied to obtain the best fit with the measured data (Vanrolleghem et al., 1999). However, one of the major limitations of respirometric analysis is interference due to endogenous respiration (and other endogenous processes).

Other methods for obtaining organism fractions target specific properties of the organism of interest—for example, the measurement of chlorophyll concentrations as surrogates for algae biomass. Alternatively, specific activity measurements target organism-specific processes such as nitrification and photosynthesis.

One final note about the formulation of RWQM1 to include additional water quality constituents and their associated model parameters is that the determination of these parameters will likely necessitate improvements in data collection. However, the insight provided by improved models can in turn lead to more directed monitoring programs (Somlyódy et al., 1998). This, and other implications of increased parameterization are discussed in Chapter 4.

1.3.2 State Variables for Bacteria

In all Streeter-Phelps-based models, the basic idea for the bacteria decomposition process by which dissolved oxygen is consumed is that pelagic bacteria in the water column mediate the oxidation of organic matter. A major shortcoming in the formulation of this process is simply that bacterial groups are not included as state variables. For example, in QUAL2E, the growth of pelagic bacteria is modeled by a constant degradation coefficient, but the rest of their metabolism

is ignored. The result is that any effect on other water quality components due to changes in the population and character of the bacterial population is neglected—dynamic changes in the degradation coefficient are ignored. Moreover, in older models, degradation is ignored altogether in the sediment compartment causing the total degradation to be grossly underestimated. (Reichert et al., 2001b).

1.3.3 Conservation of Mass

To resolve this inability to model degradation processes in the sediment, a sediment oxygen demand term (SOD) was introduced historically as a boundary condition to the water column compartment. In QUAL2E several sediment-water column boundary conditions are introduced: (1) the SOD term removes oxygen from the water column compartment, and (2) benthic flux terms release nitrogen and phosphorous to the water column. Although strictly speaking they are boundary conditions and not conversion processes, both SOD and N-P fluxes are formulated as zeroth order processes. However, problems in mass conservation arise when organic matter is allowed to settle (a sedimentation rate is introduced into the model formulation), causing mass to be removed from the system. The effects of this settled mass are still modeled through the SOD and benthic flux terms and the result is that mass conservation principles cannot be observed. Moreover, from the discussion of ill-defined water quality parameters in an earlier section, it should be clear that the use of BOD as a state variable for organic matter already intrinsically means that mass balances cannot be closed.

A solution to this limitation would be to track mass in both the water column and sediment compartments. Benthic fluxes can then be calculated from the amount of mass settled to the sediment. However, an accurate and robust description of these fluxes would require explicit consideration of the population and composition of degradation-mediating biota. RWQM1 theoretically allows such a treatment of benthic processes. However, because the description of benthic communities is fundamentally different from that of pelagic communities, successful treatment of benthic processes will require additional process descriptions of the attachment and detachment of sessile organisms or benthic biofilms. These improvements would, for example, permit an evaluation of changes in sediment conditions resulting from a transition of the wastewater effluent from non-treated to secondarily treated (Reichert et al., 2001b).

1.3.4 Integration of Models

With the achievement of water quality objectives in many US streams, there is an increasing need to develop complex non-point source models, which involve the interaction of sewerage and storm-water overflow routing models and wastewater treatment models with water quality models of the receiving stream. Unfortunately, the lack of common water quality descriptors in these models has made their combined use a very difficult task (Orlob, 1983). One of the goals of RWQM1 was to ensure compatibility with IAWQ Activated Sludge Models developed for waste water-treatment plants, by including the elements of the ASM models in RWQM1 to facilitate precisely such an integrated analysis (Somlyódy et al., 1998).

It should also be mentioned that since non-point source loads are often driven by storm events involving time-varying waste loads and stream-flow, an accurate prediction of river responses will require models capable of unsteady flow. QUAL2 has limited applicability in situations other than the steady stream-flow, constant discharge regime for which it was intended. This also makes difficult an assessment of the following situations for which RWQM1 is also intended: (1) extreme and surprising pollution events, and (2) artificially influenced rivers such as those regulated by hydropower and dams, for which there are significant diurnal variations in flow. Moreover, because the description of their water quality constituents lacks a consistent underlying basis, and without rigorous science-based definitions of system variables, current models are unable to predict responses to abrupt changes in environmental conditions, loads or hydraulics (i.e. when the system lacks spatial and temporal uniformity) (Somlyódy et al., 1998).

2 Understanding the WASP Framework

RWQM1, though often called a model, is perhaps better described as a conceptual framework for modeling. This is to be further distinguished from water quality simulation programs such as QUAL2, which represents the implementation of a theoretical construct into computer code. The term model actually denotes a site-specific implementation of a set of governing equations, with laboratory and field-based numerical values assigned to model parameters. RWQM1 is more general than this. It includes first a refined conversion submodel, that is to say a set of relations between external inputs and system variable responses for traditional water quality pollutants. This is designed to replace the conversion submodels of existing water quality models. RWQM1 also presents guidelines for the future development of consistent and standardized river water quality models.

In the present study an implementation of RWQM1 was performed within the WASP framework, which was chosen specifically for its unique flexibility in this regard. The Water Quality Analysis Simulation Program (WASP) is one of the many available water quality simulation computer programs with modular conversion subroutines for different types of water quality pollutants. Since RWQM1 has dealt primarily with the refinement of the kinetic component of water quality models, this particular implementation of RWQM1 involved the development and linking of RWQM1 conversion subroutines to the pre-existing transport subroutines WASP.

The Environmental Protection Agency's (EPA) WASP 5.0 is a dynamic-compartment—both water column and sediment—modeling program that has been used to predict the response of aquatic ecosystems to pollution from both man-made and natural sources⁵ (Ambrose et al., 1988; Di Toro et al., 1983; Connolly and Winfield, 1984). While the basic time-varying processes of advection, point and non-point loadings, and boundary exchanges are modeled by the WASP transport routines, the kinetic component of the model can be custom tailored by the user or selected from a library of available routines. Two such kinetic models are already included with WASP—EUTRO5 for traditional water quality and TOXI5 for hazardous pollutants. This unique flexibility allows the user to develop and implement new kinetic

⁵ For a brief description of the underlying mathematical basis of the dynamic-compartment structure, see Chapter 3, Kinetic Structure

subroutines that are linked to the transport component through a separate subroutine called WASPB.

This chapter describes key input, process, and kinetic subroutines of EUTROWASP and illustrates their importance in the overall flow of the program. An understanding of the functional relationships among subroutines will make clearer the discussion in Chapter 3 of changes to the existing subroutines and the additions required for the development of RWQMWASP.

2.1 Common Blocks

The subroutines of WASP can be essentially grouped into four main functional categories—input, process, kinetic and utility. Information is shared among these subroutines primarily through common blocks, where global variables are defined. The two main common blocks needed for the successful compilation of EUTROWASP are WASP.CMN and EUTRO.CMN. EUTROWASP subroutines that need to gain access to one or more of the global memory addresses—where variables, constants or parameters declared in these common blocks are stored—do so by means of the INCLUDE statement in the subroutine heading.

WASP.CMN is the common block used by the WASP driver program to create an executable. The variables declared in WASP.CMN can be grouped into three broad categories: (1) those that constrain the dimensionality of many simulation variables, (2) those that are established to handle reading from input files and writing to output files, and (3) other variables declared as global in order to allow easy passage of information from one subroutine to the next within the program.

Parameters are the first category of variables. They are needed to control dimensionality. Examples of parameters are SY, SG, CX, PR and BC, which establish an upper limit for the number of systems, segments, kinetic constants, parameters and boundary conditions respectively. Parameter values must be changed to constrain or broaden the capabilities of the program.

Input/Output units are integer values assigned in WASP.CMN that allow the main program to read specific information from input files and write information to output files. For example a unit value of 2 for the input unit IN refers to a user-created sequential formatted input

file. In subroutine WASP1, after opening the user's input file with the OPEN statement, IN is seen in a formatted READ statement that reads the following simulation control parameters from Record 4 of Data Group A of that input file: NOSEG, NOSYS, ICFL, MFLAG, JMASS, NEGSLN, INTYP, ADFAC, ZDAY, ZHR, ZMIN, TFLAG (see Figure 1, and Ambrose et al., 1993). It is suggested that the new user not change the unit values for the Input/Output files.

```

C                               User Supplied Input Dataset
C
  OPEN (UNIT = IN, STATUS = 'OLD', ACCESS = 'SEQUENTIAL',
1     FILE = INFIL)
:
:
  READ (IN, 5070, ERR = 1060) NOSEG, NOSYS, ICFL, MFLAG, JMASS,
1     NEGSLN, INTYP, ADFAC, ZDAY, ZHR, ZMIN, tflag
5070 FORMAT(7I5,2F5.0,F3.0,F2.0,i5)

```

Figure 1. Opening the Input File and Reading Record 4 of Data Group A in WASP1

Finally, an example of the third category of variables is NOSEG (see Figure 1), the integer value for the number of segments or expanded control volumes in the user-defined stream network, to be distinguished from SG, the total number of segments the model will handle. A more illustrative example for this category is ISEG, the segment number. Certain subroutines, EULER in particular (see Figure 2), contain algorithms that loop from segment number 1 to segment NOSEG, performing calculations with numerical values that are unique to each segment (i.e. the volume of segment ISEG or the concentration of a particular water quality constituent denoted by ISYS within segment ISEG).

```

DO 1030 ISYS = 1, NOSYS
  IF (SYSBY (ISYS) .GT. 0) GO TO 1030
DO 1040 ISEG = 1, NOSEG
  IF (IPROP .GT. 3) IPROP=0
  IPROP=IPROP + 1
  VOLOLD = BVOL (ISEG)
  VOLNEW = MVOL (ISEG)
  MASS = C (ISYS, ISEG) *VOLOLD
  MDER = CD (ISYS, ISEG)
  C (ISYS, ISEG) = (MASS + DT*MDER) /VOLNEW

```

Figure 2. Euler Mass Derivative Scheme in EULER subroutine

EUTRO.CMN is the Include common block for the EUTRO kinetic subroutines. It contains kinetic constants (to be distinguished from parameters) set by EQUIVALENCE statements to numbers assigned by the systems analyst during subroutine coding (see step (3) in section 3.1). When the values for the kinetic constants are input by the user, the number assigned to the kinetic constant must also be input so that the program recognizes what is being input. Examples of kinetic constants are given on page 40 of the WASP5 Manual, Part B (Ambrose et al., 1993b). These include process rates (i.e. nitrification, denitrification, phytoplankton growth rate, respiration), temperature coefficients for some of those rates (i.e. temperature coefficients for phytoplankton respiration and decomposition), half saturation constants (i.e. half saturation constant for phytoplankton mineralization), and fractions (i.e. nitrogen-to-carbon ratio in phytoplankton, fraction of dead and respired phytoplankton nitrogen recycled to organic nitrogen). For the sake of clarity, there should be some organizational logic to the kinetic constant number assignments. EUTRO makes its assignments by order of water quality state variables.

Of course, not just the process rate constants, but all global variables used in the kinetic subroutines, including those needed for calculating reaeration, water column light penetration, water temperature and benthic fluxes are included in EUTRO.CMN. User-defined kinetic time functions are also declared in EUTRO.CMN. These include time-variable temperature functions, a total daily radiation function, a function describing the fraction of the day with sufficient light for growth, an herbivorous zooplankton population function, a wind velocity function (for wind-driven reaeration), time-variable extinction coefficient functions, ammonium and phosphate sediment flux functions, time-variable velocity functions, time-variable salinity functions, time-variable ambient air temperature functions (for wind-driven reaeration), and a time-variable ice cover function (see Chapters 8 and 10 of Ambrose et al., 1993b). The complexity of the water quality analysis determines which of these can be omitted, and guidelines for six representative levels of complexity are given in the WASP5 Manual Part B (Ambrose et al., 1993b).

2.2 Input Subroutines

WASP4V is the main program control module in WASP. It opens files, operates the calling sequence for the input, simulation, and output subroutines, and it closes files (see Figure

3). Of the subroutines called by WASP4, EULER is the simulation workhorse, calling DERIV at each time-step to recalculate the mass derivatives (Ambrose et al., 1988). EULER then integrates the mass derivatives over a time-step to find the new concentrations for the simulated water quality constituents. In this section, we examine the input, simulation, and kinetics subroutines separately and in relation to the overall flow of the program.

INIT → WASP1 → WASP2 → WASP3 → WASP4 → WASP5 → WASP6
WAS6A → WASP7 → WASP9 → EULER → WNCLOS(1) → CLSCRN

Figure 3. Sequence of subroutine calls in WASP4V

Input subroutines do not perform any simulation computations, but they do read from input files, check for errors in these inputs (i.e. missing information or improperly formatted inputs), and write correctly formatted information to the simulation output. All WASP5 user-input data is divided into ten groups, A through J. For each of these Data Groups, one or more WASP input subroutines are responsible for performing the aforementioned input and write functions. Each of these is considered below. A summary of this information may be found in the WASP4 User's Manual and Programmer's Guide (Ambrose et al., 1988)

2.2.1 WASP1

WASP1 opens the input and output units, reads from Data Group A, stores in the information contained therein to memory, and prints it to the simulation output (.OUT file). Basic simulation information is provided in Data Group A, including the number of modeled state variables, segments, number of model time-steps, calculation time-steps, number of print intervals, use of a restart file, and system bypass options. A number of simulation arrays and arrays are initialized in WASP1. For example, the slope and intercept arrays for a number of piece-wise linear (time-varying) transport functions computed in WASP are initialized in WASP1. Time functions are variables that can be expressed in the following form (Di Toro et al., 1983):

$$VAL = M*TIME + B$$

where,

VAL = the desired value of the function at time TIME
M = slope of piecewise linear function used to approximate the value
B = intercept of piecewise linear time function

Examples of such arrays are:

BBC(SY)	Boundary condition intercepts
BWK(SY)	Forcing function (waste load) intercepts
MBC(SY)	Boundary condition slopes
MWK(SY)	Forcing function (waste load) slopes

All arrays are set using the utility functions SETIA, SETRA, SETRB, and SETXA, which initialize integer, real, three-dimensional real, and double-precision arrays, respectively, to the specified value. For example the following command in WASP1:

```
CALL SETRA (C, SY, SG, 0.0)
```

calls the function SETRA, which reads C as the name of the array, SY as the number of rows in array C, and SG as the number of columns in array C. The value 0.0 is assigned to each of the indices as shown in Figure 4. The C array holds information on the concentrations of simulated water quality state variables.

```
SUBROUTINE SETRA (ARRAY, NROW, NCOL, VALUE)
:
REAL ARRAY (NROW, NCOL)
DO 1000 IROW = 1, NROW
  DO 1000 JCOL = 1, NCOL
    ARRAY (IROW, JCOL) = VALUE
1000 CONTINUE
RETURN
END
```

Figure 4. Initializing Simulation Arrays in WASP1

2.2.2 WASP2

WASP2 reads input from Data Group B (Exchange Coefficients), which contains information on the number of exchange fields, exchange time functions, and exchange bypass options. Dispersion can occur between surface water segments (exchange field 1), and between pore water segments (exchange field 2). Multiple exchange time functions can be specified for each segment. These are read in a manner similar to the reading of advective flows inputs (see WASP4 for a detailed discussion), except that in WASP4 six flow fields are allowed instead of two. For a full description of exchange fields see pages 24-25 of WASP5 User Manual, Part A (Ambrose et al., 1993a). WASP2 sets dispersion coefficients, cross-sectional areas, and characteristic lengths between segment exchange pairs. From these values bulk exchanges are computed, and converted from m^3/sec to m^3/day . Information is stored in memory and printed to the output file.

2.2.3 WASP3

WASP3 reads input from Data Group C (Segment Volumes). If indicated by the ICRD switch specified in Data Group A, these volumes are read from the sequential formatted restart file. Otherwise segment properties are read from the input file. Important segment properties are the segment number (ISEG), the segment immediately below ISEG (IBOTSG), the segment type (1=surface water segment, 2=subsurface water segment, 3=upper bed segment, and 4=lower bed segment), and the segment volume in m^3 (BVOL(ISEG)). Hydraulic geometry parameters must also be specified for the calculation of reaeration or volatilization from the segment. These are not used in transport calculations.

Another utility function, SCALP is used in WASP3. It is called whenever an array has to be multiplied by a scaling factor. In Data Group C, the user may specify a scale factor to multiply each volume. For example, if a segment has a volume of $2.4 \times 10^5 \text{ m}^3$, a value of 2.4 may be input for BVOL(ISEG) and 10,000 for SCALV. The following assignments are then performed:

```
CALL SCALP (BVOL, SCALV, NOSEG)
```


The SCALP utility function completes the operation shown in Figure 6. As before, the Data Group C information contained in the input file is stored in memory and written to the formatted output file.

```
      SUBROUTINE SCALP (BVECT, SCALE, NO)
      :
      INCLUDE 'WASP.CMN'
      REAL BVECT (SG)
      DO 1000 I = 1, NO
         BVECT (I) = BVECT (I)*SCALE
1000 CONTINUE
```

Figure 5. Scaling Factor calculation in subroutine SCALP

2.2.4 WASP4

WASP4 reads advective flows from Data Group D of the input file. Flows can be input for up to six transport fields. These are as follows:

- | | |
|----------------------|--|
| Transport Field 1 | Advective flows in the water column |
| Transport Field 2 | Pore water advective flows |
| Transport Fields 3-5 | Sediment transport velocities for three different solids types—
particulate organic matter, phytoplankton, and inorganic sediment |
| Transport Field 6 | Evaporation and Precipitation |

Each transport field can have multiple flows, specified by multiple piecewise linear time functions. For each time function, the portion of flow occurring between each upstream and downstream segment for each break in the flow time function must be specified. Finally, a flow bypass option allows the transport to be set to zero in one or more systems. This option makes possible the simulation of water quality constituents that are not transported such as rooted plants (not presently included in the simulation). Also, if indicated by the IOQPT switch, flows can be read from a hydrodynamic file created in DYNHYD5 (Chapter 5 of Ambrose et al., 1993b). Finally, as with dispersive exchanges, advective flows are converted from m^3/sec to internal units of m^3/day .

A closer examination of WASP4 reveals that the main set of commands is repeated for all fields as shown in Figure 6.

```

C          LOOP THROUGH FIELDS
C *****
C *   FIELD 1 = Total Water Transport, BQ is continuity array, *
C *           QT contains flows for time function             *
C *   FIELD 2 = Dissolved transport only                      *
C *   FIELDS 3 or greater are solids fields, BQ is array of  *
C *           areas, QT contains settling and scour velocities*
C *****
C
DO 1020 NF = 1, NFIELD

```

Figure 6. Input of advective flows for each transport field in WASP4

The three-dimensional continuity array, BQ seen in the comments section in Figure 6, was initialized in WASP1 to zero, with the following statement:

```
CALL SETRB (BQ, MNF, MNI, S2, 0.0)
```

Here, BQ is the name of the continuity array, MNF is the maximum number of flow fields (6), MNI is the maximum number of time functions for each flow field (10), and S2 is the number of segment pairs, calculated in WASP.CMN by the following integer divide $4*SG/3$. For six segments aligned one against the other along a river reach, there are eight segment pairs (0-1,1-2, 2-3, 3-4, 4-5, 5-6, and 6-0). The first and the last segment pairs arise from flows to and from boundary segments. To emphasize the analogy mentioned in the WASP2 section, it should be noted that a similar continuity array is created for dispersive exchanges (see WASP3), where BR is the continuity array for exchange flows:

```
CALL SETRB (BR, MNF, MNI, S2, 0.0)
```

In WASP4, this continuity array is read from the input file for each of the flow fields. The commands shown in Figure 7 are executed.

```

        READ (IN, 5010, ERR = 1000) NINQ (NF), SCALQ, CONVQ
        WRITE (OUT, 6040) NF, NINQ (NF), SCALQ, CONVQ
5010  FORMAT(I5,2F10.0)
      :
      DO 1040 NI = 1, NINQX
      :
        READ (IN, 5020, ERR = 1000) NOQS (NF, NI)
      :
C
C*****
C          Read in continuity array
C*****
C
        NOQ = NOQS (NF, NI)
        IF (NOQ .GT. S2) GO TO 1050
        READ (IN, 5030, ERR = 1000) (BQ (NF, NI,
1          NQ), JQ (NF, NI, NQ),
2          IQ (NF, NI, NQ), NOQ = 1, NOQ)
5030  FORMAT(4(F10.0,2I5))
      :
      :
1050 CONTINUE
        WRITE (OUT, 6190) NOQ, S2
6190  FORMAT(///10X,'The Number of Flows ',
1    'Specified is ',I5/10X,
2    'The Maximum Dimensioned for This Version of WASP',
3    ' is ',I3/10X,'Respecify Flows or Redimension Parameter "S2"',
4    'in the Common',/,10X,'Block and Recompile.')
        CALL WERR(20,1,0)
        CALL WEXIT('Dimension Error See the Output File',1)
      :
      :

```

Figure 7. Reading in the continuity array in WASP4

NINQ signifies the number of time functions for the field (NF) under consideration. NOQ denotes the number of flow responses for the field under consideration and time function NI. Since these must be defined for each segment pair, a simulation with six water column segments will yield eight unit-flow responses. Thus, the value for NOQ cannot be greater than S2, as indicated by the bolded IF statement.

Thus for each time function in field 1 (1 to NINQX), BQ, the portion of flow between segment pairs JQ, the upstream segment, and IQ, the downstream segment, are read for all segment pairs (1 to NOQ). This process will be repeated for all transport fields. For some transport fields (i.e. evaporation or precipitation), BQ is the area between adjacent segments, not the flow response.

The last major task performed by WASP4 is to read the advective time functions for each field. The following commands are executed by WASP4 to read in the values and times for each piecewise linear time function:

```

1      READ (IN, 5050, ERR = 1000) (QT (NF, NI, NB),
      TQ (NF, NI, NB), NB = 1, NOBRK)

```

Figure 8. Reading in piecewise linear time functions for advective flows in WASP4

For a given field and time function (each field can have multiple), the value for each advective flow (QT) and the break for that flow (TQ) are read by WASP4. In the statement in Figure 8 all of the time function values and their corresponding breaks are read into TQ and QT respectively. A similar READ statement can be found in WASP2 for exchange time function breaks (TR) and exchange flows (RT):

```

1      READ (IN, 5050, ERR = 1020) (RT (NF, NT, NB),
      TR (NF, NT, NB), NB = 1, NOBRK)

```

Figure 9. Reading in piecewise linear time functions for exchange flows in WASP2

Before completing, WASP4 checks continuity for advective flows (Field 1) to ensure that they were properly specified. It does this for all advective flow time functions as shown in Figure 10. The logic is quite simple and very elegant. As the segment loop through each of the unit flow responses, ISEG will be equal to JQ once and only once, that is to say, when it is receiving flow from an upstream segment. Moreover, ISEG will be equal to IQ at the next iteration, that is to say, when it becomes an upstream segment. In the first iteration, the portion of the flow BQ going from IQ to ISEG in is added to QSUMX, while in the second iteration, it is subtracted from QSUMX. All other iterations produce no change in QSUMX. Finally, after iterating NOQS times, if QSUMX is not similar to its initial value (plus or minus rounding error) after these two iterations, continuity has been violated and an error message is printed.

```

DO 1080 ISEG = 1, NOSEG
  DO 1090 NI = 1, NINQ (1)
    QSUMX = 0.0
    DO 1100 NQ = 1, NOQS (1, NI)
      IF (IQ (1, NI, NQ) .EQ. ISEG) QSUMX =
1      QSUMX + BQ (1, NI, NQ)
      IF (JQ (1, NI, NQ) .EQ. ISEG) QSUMX =
1      QSUMX - BQ (1, NI, NQ)
1100    CONTINUE
      IF (QSUMX .GT. 1.0E-06)then
C      WRITE (OUT, 6160) ISEG, NI
      CALL PROMPT('Flows Do Not Balance, Check Continuity',0)
      WRITE (6, 6161)
      ENDIF
1090    CONTINUE
1080  CONTINUE

```

Figure 10. Mass balance check for advective flows (FIELD1) in WASP4

To summarize, the following arrays describing the temporal and spatially varying flow for all flow fields are read into memory and printed to the output file in WASP4. The initialized arrays are shown below:

IQ(MNF,MNI,S2)	upstream segment
JQ(MNF,MNI,S2)	downstream sediment
BQ(MNF,MNI,S2)	portion of flow between two segment pairs
QT(MNF,MNI,MB)	advective flow specified in piecewise linear format for flow field
TQ(MNF,MNI,MB)	time for piecewise linear time function

2.2.5 WASP5

WASP5 reads information from Data Group E (Boundary Conditions) of the input file. Boundary conditions are input as time-varying boundary concentrations. Although boundary concentrations can be different for each water quality component, the number of breaks specifying the broken line approximation must be the same for all boundary segments. A system bypass option allows specification of a zero concentration boundary condition for bypassed systems. As before, information is stored in memory and printed to the formatted output file.

2.2.6 WASP6

WASP6 reads information from Data Group F (Waste Loads) of the input file. Multiple point-source waste loads can be specified for each system. NOWK(ISYS) is the number of waste loads for system ISYS. These can be specified as sources or sinks (negative loads). For each load, the segment receiving that load must be specified, and so must the piecewise linear time function describing that load. This is repeated for each system, but may be bypassed for one more systems. As before, information is stored in memory and printed to the formatted output file.

The following lines of code illustrate an important feature of how WASP6 reads in the time functions for waste loads. An array called FILE50 is set up with dimensions [*two times* all loads for all systems] by [maximum number of breaks]. For each system, WASP6 loops through all of the loads, and the break times, and load values for each break are read into FILE50 in two adjacent columns, denoted by IM and II respectively. The resulting array has column entries describing each loading time function for loads 1 through NOWK(ISYS) for each system along the row dimension. The use of arrays with FILE names to hold piecewise linear time functions is seen throughout WASP and will be important for calculations that update these time functions.

```

DO 1000 ISYS = 1, NOSYS
C
  READ (IN, 5000, ERR = 1010) NOWK (ISYS)
  :
  NO = NOWK (ISYS)
  :
  DO 1040 I = 1, NO
    II = I + I + 2*WK*(ISYS - 1)
    IM = II - 1
    READ (IN, 5020, ERR = 1010) IWK (ISYS, I), NOBRK
    IF (NOBRK .GT. MB) CALL BRKERR (NOBRK)
    READ (IN, 5030, ERR = 1010) (WKT (J), T (J), J = 1, NOBRK)
    :
    NBRK50 (ISYS, I) = NOBRK
    DO 1050 J = 1, NOBRK
      FILE50 (J, IM) = T (J)
1050 CONTINUE
C
    DO 1060 J = 1, NOBRK
      WKT (J) = WKT (J)*WSCAL
      FILE50 (J, II) = WKT (J)
1060 CONTINUE
    :

```

Figure 11. Reading in of point-source loading time functions in WASP6

2.2.7 WAS6A

For non-point source loads, an external non-point source file is required, which should contain information on what system variables and segments receive these loads. If non-point source loads are specified in Data Group F (i.e. if the LOPT switch in Data Group F is set to 1), WAS6A will prompt the user to input the name of a non-point source file created by a run-off model.

2.2.8 WASP7

WASP7 reads inputs from Data Group G (segment parameters), Data Group H (kinetic constants), and Data Group I (kinetic time functions). It saves this information to memory and prints to the output file.

Data Group G parameters were described in the discussion of the EUTRO common block and will not be discussed in great detail here. However, it will be noted that parameters are

spatially variable characteristics of the water body, as distinct from kinetic constants (in Data Group H), which are assumed to be spatially and temporally uniform. While the correctness of this assumption may be disputed, it cannot but be made, and it will be noted that the task of finding a unique set of numerical values for non-varying kinetic constants is already nontrivial. It is for this reason that the question of parameter identifiability will be pursued in later sections.

WASP7 reads in the number of parameters to be used (NOPAM), a descriptive name for each parameter (ANAME(ISC)), a number that uniquely identifies each parameter (ISC), and the scale factor for each parameter (PSCAL(ISC)). Next the values for all used parameters in each segment are set as PNAME(ISEG,ISC).

Data Group H kinetic constant entries are similar to those encountered in the Data Group G, except that these are not segment specific, and they can be categorized not just by name, but by group and sub-group fields (see Chapter 9 of Ambrose et al., 1993). It may not seem entirely logical to organize kinetic constants by order of system, since the kinetic process rates involved tend to affect multiple systems. For example, constants 11, 12, and 13 are the nitrification rate (K_{12C} , [day^{-1}]), the temperature coefficient for nitrification (K_{12T} , []), and the half-saturation constant for nitrification-oxygen limitation (K_{NIT} , [$\text{mg O}_2/\text{L}$]). These constants appear both in the AMMONIA and NITRATE subroutines of EUTRO (see section 2.4).

However, since the process operates on AMMONIA, with NITRATE as a product, the user may wish to organize his input so that these three constants fall into the System 1 (NH₄) Group, and perhaps under the sub-heading of kinetic constants. However, as the reader familiar with the structure of RWQM1 may anticipate, besides the elemental mass fractions characterizing different types of organic material, the model parameters (constants) of RWQMWASP will include a number of constants not akin to those found in EUTRO (see EUTRO.CMN in sub-section entitled Common Blocks), that may rightly fall into several system variable groups. The clearest example of this is offered by the model parameters for chemical equilibrium (processes 16-21 of RWQM1), which may justifiably be grouped under any of the species involved in the equilibrium.

The last set of commands in WASP7 reads inputs from Data Group I (Kinetic Time Functions) as shown in Figure 12. Time functions are entered in piecewise linear format. However, the choice of time functions will depend on the structure of the kinetic routines. As

discussed in the earlier section on EUTRO.CMN (see section 2.1) the kinetics component of EUTRO WASP includes a number of kinetic time functions with number designations as shown on page 71 of the WASP5 Manual, Part B (Ambrose et al., 1993b). Time functions are identified by number (ISC), and described by number of breaks (NOBRK(ISC)) and values (VALT(K)) for each time break T(K). In the following section of code taken from WASP7, for each time function, a name (ANAME), number of breaks (NOBRK) and identifying number (IFUNC) are read. If the number of breaks specified in the input file is greater than the maximum number of breaks (MB⁶) handled by WASP, a number of errors print to the screen via call to BRKERR. If the number of breaks is less than or equal to the maximum number, and greater than zero, the values for the time function (VALT) are read for each break.

```

DO 1120 I = 1, NFUNC
  READ (IN, 5080, ERR = 1000) ANAME (1), nobrk, IFUNC
  IF (NOBRK .GT. MB) CALL BRKERR (NOBRK)
  II = IFUNC + IFUNC
  IM = II - 1
  IF (NOBRK .GT. 0) READ(IN, 5090, ERR = 1000) (VALT (J), T (J),
1      J = 1, NOBRK)
5080  FORMAT(A5,2I5)
5090  FORMAT(8F10.0)
      :
      :
      NBRK73 (IFUNC) = NOBRK
      DO 1130 J = 1, NOBRK
          FILE73 (J, IM) = T (J)
          FILE73 (J, II) = VALT (J)
1130  CONTINUE

```

Figure 12. Data Group I Kinetic Time Function Input in WASP7

A more interesting feature of this code, however, as in the commands encountered in WASP6, is the use of FILE arrays to handle the kinetic time functions. As mentioned before, this motif is seen throughout WASP, and is an important part of the structure of the program. A quick look at WASP.CMN shows that FILE73 (MB, M73) is an array of size MB by M73, [maximum number of time function breaks] by [*twice* the maximum number of environmental time functions]—M73 is set to 2*TF in the parameter declaration section of WASP.CMN. It is clear in Figure 12 that FILE73 is intended to hold the values and time breaks for all environmental

⁶ MB is specified in WASP.CMN. The default maximum value is 400.

time functions. The variable II is twice the number assigned to a given environmental time function and IM is one less than II, so that each break time and value are read into the array in adjacent columns. As a result, each environmental time function entry occupies two columns in the array.

2.2.9 WASP9

WASP9 reads Data Group J (Initial Conditions) from the input file. Segment concentrations at time zero must be specified for each system. The user must also specify whether the system is being transported in its pure or sorbed form by specifying the system solids type. If the system is to be transported in both dissolved and particulate forms, a dissolved fraction and a particulate fraction solids type must be specified. Table 2 shows the system solids types recommended by WASP (see Chapter 11 of Ambrose et al., 1993b):

Table 2. Solids Types for System Particulate Fractions

System	Data Group J Solids Type Variable	Recommended Solids Type for Particulate Fraction
NH3	IFIELD(1)	3 (particulate organic)
NO3	IFIELD(2)	5 (inorganic sediment)
PO4	IFIELD(3)	5 (inorganic sediment)
PHYT	IFIELD(4)	4 (phytoplankton)
CBOD	IFIELD(5)	3 (particulate organic)
DO	IFIELD(6)	5 (inorganic sediment)
ON	IFIELD(7)	3 (particulate organic)
OP	IFIELD(8)	3 (particulate organic)

The user should take care to specify dissolved fractions of 1.0 and 0.0 for DO and Phytoplankton respectively.

Initial concentrations and dissolved fractions are read into memory for each segment (C(ISYS, ISEG), and DISSF) by the following set of statements in WASP9. If indicated by the ICRD switch, initial conditions may be read from the restart file. The computations shown in Figure 13 are performed for all system variables via the DO loop over ISYS = 1,NOSYS:

```

DO 1000 ISYS = 1, NOSYS
:
C   INITIALIZE F -ARRAY AS SPECIFIED BY IFIELD:
C
DO 1010 ISEG = 1, NOSEG
  F (IFIELD (ISYS), ISYS, ISEG) = 1.0
1010 CONTINUE
C
C   READ IN INITIAL CONCENTRATIONS AND DISSOLVED FRACTIONS:
C
  READ (ICRD, 5010, ERR = 1020) (ANAME (I), C
  1   (ISYS, I), F (2, ISYS, I),
  2   I = 1, NOSEG)
C
C   If Initial Conditions are close to Zero set to 1.0e-24
C
DO 1030 I = 1, NOSEG
  IF (C (ISYS, I) .LT. 1.00E-24) C (ISYS, I) = 1.00E-24
1030 CONTINUE

```

Figure 13. Initial Concentrations and Dissolved Fractions in WASP9

The first set of statements in the code initializes the array F, a three-dimensional real array for dissolved fractions. The reader may recall that this array was already initialized for the maximum number of systems, transport fields and segments in WASP1, where it was assigned dimensions [Maximum Number of Flow Fields] by [Maximum Number of Systems] by [Maximum Number of Segments]. The commands shown in Figure 14 are executed in WASP1 to set all dissolved fraction in all transport fields to 0.0, and then to set the dissolved fractions for water column transport (Field 1) to 1.0 in all segments. The reader will notice that the initializations subsequent to those performed in WASP1 are performed not for the maximum number of elements handled by WASP, but to the number of elements specified by the user in the input file (i.e. NOSEG not SG, and NOSYS not SY):

```

CALL SETRB (F, MNF, SY, SG, 0.0)
C
DO 1090 ISYS = 1, NOSYS
  DO 1090 ISEG = 1, SG
    F (1, ISYS, ISEG) = 1.0
1090 CONTINUE

```

Figure 14. Initialization of the array F in WASP1

In WASP9, for each system, the array F is initialized to 1.0 over the entire simulation spatial domain, for the user-specified IFIELD value assigned to the system. After this, the dissolved fractions are only read into the pore water transport field (FIELD 2) of F's transport field dimension. The last set of statements converts concentration values of zero to small number values, to avoid problems of numerical instability.

```
DO 1040 ISEG = 1, NOSEG
    F (IFIELD (ISYS), ISYS, ISEG) = 1.0 - F (2, ISYS, ISEG)
1040 CONTINUE
```

Figure 15. Calculation of Particulate Fractions in WASP9

Finally, the statements in Figure 15 calculate the initial particulate fractions for each system over the entire spatial domain. The particulate fraction assignment to the F array will now vary from one system variable to the next according to the solids field assigned by the IFIELD variable in Data Group J. Thus, the dissolved fraction for each system is stored in F(2, ISYS, ISEG), and the particulate fraction in F(IFIELD(ISYS), ISYS, ISEG).

2.3 Process Subroutines

2.3.1 EULER

Once the input data are read, WASP4V passes control to EULER to begin the simulation (see Figure 3 for sequence of subroutine calls). EULER steps through time performing a first-order integration. Before the simulation can start, timers for flows and exchanges must be initialized to the simulation starting time TZERO. This initialization is performed with the aid of subroutine TINIT, which sets the initial intercepts and slopes for any piecewise linear time function (flow, exchange, kinetic, boundary and waste load) if the simulation starts at any time other than zero.

Initial printouts are performed by WAS13, and then the call is made to DERIV to calculate the mass derivatives. The call to DERIV is made at each time-step and the new mass for each water quality constituent calculated as shown in Figure 16.

```

1020 CONTINUE
C      CALL COLOR ('BLUE','BLACK')
      DO 1030 ISYS = 1, NOSYS
        IF (SYSBY (ISYS) .GT. 0) GO TO 1030
        DO 1040 ISEG = 1, NOSEG
          IF (IPROP .GT. 3) IPROP=0
          IPROP=IPROP + 1
C      IF (MFLAG .LT. 2)CALL OUTSXY (72,5,PROP(IPROP))
          VOLOLD = BVOL (ISEG)
          VOLNEW = MVOL (ISEG)
          MASS = C (ISYS, ISEG)*VOLOLD
          MDER = CD (ISYS, ISEG)
          C (ISYS, ISEG) = (MASS + DT*MDER)/VOLNEW
          CD (ISYS, ISEG) = 0.0

```

Figure 16. Euler Mass Derivative Scheme in EULER subroutine

The calculation takes the mass derivative for each system ISYS and segment ISEG, CD(ISYS,ISEG), and multiplies by the time-step to obtain the change in mass. The result is added to the old mass to obtain a new mass, which must in turn be divided by the new volume in the segment. Finally, the mass derivative is returned to zero for the next calculation.

If the negative solution switch was set to zero in the input file, EULER performs a negative solution correction. With this option, any negative concentration is replaced by a value equal to half the old concentration divided by the new volume. Next, the simulation time is increased by the step-size ($T = T + DT$) and the new day counter is adjusted as necessary. Intermediate printouts are performed at specified print intervals via a call to WAS13.

```

      IF (TIME .GE. TPRINT) CALL WAS13

```

Finally, a new mass derivative is performed and EULER checks for the next time step as shown in Figure 17. If the final simulation time is detected, the new concentrations are calculated (GOTO 1020), display variables are stored, and volumes and concentrations are printed to the restart file (if specified). Control is then passed back to WASP4V.

```

      IF (T .LT. TEND - .00001) CALL DERIV
      :
      IF (T .LT. TEND - .00001) GO TO 1020

```

Figure 17. Calculation of mass derivative at final time-step

2.3.2 DERIV

DERIV, called in EULER, takes the mass derivatives at each time-step. It first makes a call to WAS8B to update piecewise linear time functions for dispersion and flows. This subroutine returns an array QINT(NF,NI) of updated flows and another array BRINT(NF,NI) of updated exchanges for the NI flow time functions in each field designated by NF. If IQOPT is set in the input file to 3 or greater, QINT(1,1) is set to 1.0 and returned, and DERIV will obtain new field 1 flows and volumes from the hydrodynamic file by calling HYDROIN.

```

DO 1000 NF = 1, NFIELD
  NINQX = NINQ (NF)
  IF (NINQX .EQ. 0) GO TO 1000
C
C      LOOP THROUGH INFLOWS
C
DO 1010 NI = 1, NINQX
  IF (IQOPT .GE. 3 .AND. NF .EQ. 1) THEN
    QINT (1, 1) = 1.0
  ELSE
    IF (TIME .GT. TNQ (NF, NI)) ITIMQ (NF,
1      NI) = ITIMQ (NF, NI) + 1
    IF (ITIMQ (NF, NI) .GE. NBRKQ (NF, NI)) THEN
      NOBRK = NBRKQ (NF, NI)
      TNEXT = TQ (NF, NI, NOBRK) - TQ (NF, NI, 1)
      DO 1020 J = 1, NOBRK
1020        TQ (NF, NI, J) = TQ (NF, NI, J) + TNEXT
      CONTINUE
      ITIMQ (NF, NI) = 1
    END IF
    IT = ITIMQ (NF, NI)
    TNQ (NF, NI) = TQ (NF, NI, IT + 1)
    SLOPE = (QT (NF, NI, IT + 1) - QT (NF, NI, IT)
1      ) / (TQ (NF, NI, IT + 1)
2      - TQ (NF, NI, IT))
    DELT = TNQ (NF, NI) - TIME
    QINT (NF, NI) = QT (NF, NI, IT + 1) - SLOPE*DELT
  END IF
1010 CONTINUE

```

Figure 18. Updating time function for flows in WAS8B

ITIMQ, a counter array used for obtaining correct slopes and intercept values for time-varying advective flows, and ITIMR, a similar counter for exchange coefficients are updated. Similarly, TNQ, an array of times at which the next break point occurs is also updated, along with TNR, the same array for exchange flows.

Next, a call is made to WASPB to compute the kinetic mass derivatives. In WASPB, several steps can be followed. For the moment it will suffice to note that kinetic time functions and time steps are updated by a call made to WASP8:

```

IF (TIME .GE. NTF) CALL WASP8 (MFUNC, BFUNC,
1  NFUNC, 4, ITIME, NTF, 73)

```

In WASP8, new slopes and intercepts are calculated, and a variable is set to the next time that the kinetic time function in question is to be updated. The FILE## motif comes up again, and we see that all kinetic time functions entries are read from FILE73 into WASP8 for the update algorithms.

```

SUBROUTINE WASP8 (SLOPE, B, NO, IOPT, ITIME, NT, IFILE)
:
DO 1060 I = 1, TF
  IF (NBRK73 (I) .GT. 0) THEN
    IF (NFUNT (I) .LT. NT) NT = NFUNT (I)
    IF (TIME .LT. NFUNT (I)) GO TO 1060
    II = I + I
    IM = II - 1
    ITIME = ITIME (I)
    NOBRK = NBRK73 (I)
    DO 1070 J = 1, NOBRK
      T (J) = FILE73 (J, IM)
1070   CONTINUE
      IF (ITIME .LT. NOBRK) GO TO 1080
      TNEXT = T (NOBRK) - T (1)
      DO 1090 J = 1, NOBRK
        T (J) = T (J) + TNEXT
        FILE73 (J, IM) = T (J)
1090   CONTINUE
      ITIME = 1
1080   CONTINUE
      DO 1100 J = 1, NOBRK
        VAL (J) = FILE73 (J, II)
1100   CONTINUE
      B (I) = VAL (ITIME + 1)
      SLOPE (I) = (VAL (ITIME) - VAL (ITIME + 1)) /
1      (T (ITIME) - T (ITIME + 1))
      NFUNT (I) = T (ITIME + 1)
      ITIME (I) = ITIME + 1
      IF (NFUNT (I) .LT. NT) NT = NFUNT (I)
      END IF
1060 CONTINUE

```

Figure 19. Updating kinetic time functions in WASP8

B is the new intercept, SLOPE(I) is the slope for the next time function, and NFUNT(I) is the next update time. The index I, represents the number of the time function in question. These updated values are used to evaluate all of the kinetic time functions directly in WASPB. The slope MFUNC is multiplied by DELTA (= current time – new update time) and added to BFUNC(I), the old value of the time function:

$$\begin{aligned}
 \text{TEMP (4)} &= \text{MFUNC (4)} * (\text{TIME} - \text{NFUNT (4)}) + \text{BFUNC (4)} \\
 \text{ITOT} &= \text{MFUNC (5)} * (\text{TIME} - \text{NFUNT (5)}) + \text{BFUNC (5)} \\
 \text{FDAY} &= \text{MFUNC (6)} * (\text{TIME} - \text{NFUNT (6)}) + \text{BFUNC (6)} \\
 \text{WIND} &= \text{MFUNC (7)} * (\text{TIME} - \text{NFUNT (7)}) + \text{BFUNC (7)} \\
 \text{KE (1)} &= \text{MFUNC (8)} * (\text{TIME} - \text{NFUNT (8)}) + \text{BFUNC (8)}
 \end{aligned}$$

Figure 20. Updating kinetic time functions in WASPB

Finally, the kinetic derivatives are calculated in WASPB, and control is passed back to DERIV. The details of the WASPB algorithms are examined in the next section

DERIV makes a final call to WAS12 to obtain the transport and loading derivatives. WAS12 goes through a number of steps to calculate mass derivatives due to advective flow, dispersive exchange, point-source waste loading, and runoff waste loading. These are subsequently added to the kinetic derivative. For the sake of confining the discussion to the DERIV subroutine, we have opted to pass over these calculations, but the reader is referred to Ambrose et al., 1988 and Di Toro et al., 1983 for more detailed information.

It will suffice to say that IQ, JQ, and IR, JR segment pairs drive the computation of advective and dispersive transport, with calls being made to WAS8B to update variable flows. Adjustments are made to upstream and downstream concentrations. For point-source and non-point source forcing functions, the mass derivative calculations are driven by the IWK vector and INPS vector respectively. Loads are updated, this time by calls to WAS8A. The total mass derivatives in segments affected by loads are then adjusted appropriately by the loads.

The FILE## motif appears yet again, and this time, the piecewise linear time functions for loads are updated by reference to the FILE50 array. The next break time and the corresponding value for all loading time functions of a particular system, are read from the FILE50 array as shown in Figure 21.


```

NO = NOWK (ISYS)
:
DO 1060 I = 1, NO
:
NOBRK = NBRK50 (ISYS, I)
DO 1070 J = 1, NOBRK
    T (J) = FILE50 (J, IM)
1070 CONTINUE
    IF (ITIME .LT. NOBRK) GO TO 1080
    TNEXT = T (NOBRK) - T (1)
    DO 1090 J = 1, NOBRK
        T (J) = T (J) + TNEXT
        FILE50 (J, IM) = T (J)
1090 CONTINUE
    ITIME = 1
1080 CONTINUE
    DO 1100 J = 1, NOBRK
        VAL (J) = FILE50 (J, II)
1100 CONTINUE
    BWK (ISYS, I) = VAL (ITIME + 1)
    MWK (ISYS, I) = (VAL (ITIME) - VAL (ITIME + 1)) /
    1 (T (ITIME) - T (ITIME + 1))
    NWKT (ISYS, I) = T (ITIME + 1)
    ITIMW (ISYS, I) = ITIME + 1
    IF (NWKT (ISYS, I) .LT. NT) NT = NWKT (ISYS, I)
1060 CONTINUE

```

Figure 21. Updating forcing functions in WAS8A

The updated intercept $BWK (ISYS, I)$ (for loading function I of system $ISYS$), slope $MWK (ISYS, I)$, and the updated break time $NWKT (ISYS, I)$ are used for the calculation of loads in WAS12, as shown in Figure 22, and the mass derivative is adjusted appropriately as mentioned before.

```

DWKTIM = TIME - NWKT (ISYS, J)
I = IWK (ISYS, J)
IF (I .LE. 0) GO TO 1090
LOAD = (MWK (ISYS, J)*DWKTIM + BWK (ISYS, J))*1000.
CD (ISYS, I) = CD (ISYS, I) + LOAD

```

Figure 22. Adjustment to mass derivative due to PS loads in WAS12

2.4 Kinetic Subroutines

The kinetics submodel, whose computations are carried out entirely from WASPB, describe the transformation rates among eutrophication state variables. Since many of the

transformation processes are described by first-order rate equations, the current concentrations are required to calculate the change in mass for the next time-step (dM/dt). These are passed back to EULER, where they are integrated along with the loading and transport derivatives, as discussed in the last section. In the following paragraphs, we have attempted to bring out the most salient features of the important kinetic subroutines.

2.4.1 WASPB

If WASP4V is responsible for control flow in WASP, the main control module for the conversion submodel is WASPB, calling other subroutines when appropriate to calculate the mass derivatives. At specified print intervals, segment-specific state variable concentrations are stored to memory and printed to output (.DMP file). As discussed before, all kinetic time functions are updated before the calls to system variable subroutines are made. Each of these major tasks is considered below. The reader will recall that WASPB is called by DERIV to evaluate the kinetic mass derivatives.

2.4.2 EUTROINT

At the first time-step WASPB calls EUTROINT to initialize parameters, counters, and functions for the rest of the simulation:

```
IF (INITB .EQ. 0) THEN  
CALL EUTROINT
```

INITB is then set to 1.0 so that the initialization is not repeated at the next time step. Kinetic constants are set to some non-zero value to avoid erroneous computations that would produce a fatal error (i.e. divide by zero). Other constants are set to some default term used by WASP. For example IS1, the saturation light intensity for phytoplankton, is set to a default value of 300 Ly/day in the case that it is not specified by the user. Other examples include FOP and FON, the algal fractions of organic nitrogen and phosphorous. Parameter initializations may be organized in a way that improves the readability of the code. For example, all temperature coefficient initializations are made in a single block of code in EUTROINT, shown in Figure 23.

```

C Check for Zero Temperature Correction Factors and readjust to 1.0
C
CCSC
  XARG = ABS(K1320T)
  IF (XARG .LT. ROMIN) K1320T = 1.0
CCSC
  XARG = ABS(K140T)
  IF (XARG .LT. ROMIN) K140T = 1.0

```

Figure 23. Initialization of kinetic parameters in EUTROINT

EUTROINT also converts initial and boundary conditions for the phytoplankton system from units of $\mu\text{g-Chla/L}$ to internal calculational units of mg-Carbon/L as shown in Figure 24.

```

DO 1010 ISEG = 1, NOSEG
  :
  C (4, ISEG) = C (4, ISEG)*CCHL/1000.
  :
NO = NOBC (4)
DO 1020 I = 1, NO
  II = I + I + 2*bc*(4 - 1)
  NOBRK = NBRK30 (4, I)
  DO 1030 J = 1, NOBRK
    FILE30 (J, II) = FILE30 (J, II)*CCHL/1000.

```

Figure 24. Conversion of phytoplankton initial and boundary conditions

The conversion of phytoplankton boundary conditions is performed by reference to FILE30, an array of [maximum number of systems] by [*twice* the total number of boundary conditions for all systems]. For a given system, each boundary condition, expressed as a time function, occupies two columns of this array, the time in the first column and the corresponding values in the second column. Here, the values at all breaks for all phytoplankton boundary conditions are converted to mg C/L .

2.4.3 Segment Loop

After calling EUTROINT and updating the kinetic time functions, a segment loop is performed in WASPB to initialize each segment with values for each simulated time function, to set up sediment segment flags, and finally to calculate the mass derivatives. It is important that

the kinetic time function values are updated for each segment before the calculation of mass derivatives, since these calculations rely precisely on the kinetic time functions. For example, the average solar radiation incident upon the surface water segments must be known before phytoplankton growth can be calculated as shown in Figure 25.

```

DO 1000 J = 1, MXSEG
  RLGHTS (J, 2) = 1.0
  :
  DO 1000 ISEG = 1, NOSEG
  :
  itotmp = itot * itotlim(iseg)
  IAV = 0.9*ITOTmp*RLGHTS (ISEG, 2)/FDAY

```

Figure 25. Calculation of average solar radiation in WASPB segment loop

Here, IAV represents an average daily solar incidence, consistent with the use of time-steps in most eutrophication models on the order of one day. Another example of a quantity that must be initialized for each segment is temperature, since most process rates are adjusted by a temperature correction factor.

Before the mass derivatives are computed, the assignments shown in Figure 26 are made to improve the readability of the kinetics equations:

```

NH3 = C (1, ISEG)
NO3 = C (2, ISEG)
OPO4 = C (3, ISEG)
PHYT = C (4, ISEG)
CBOD = C (5, ISEG)
DO = C (6, ISEG)
ON = C (7, ISEG)
OP = C (8, ISEG)

```

Figure 26. System name assignments are made from the concentration array

2.4.4 Mass Derivatives

The mass derivatives due to transformation processes are calculated toward the end of the segment loop in WASPB in separate subroutines, one specifically for each system variable. In the following paragraphs, we consider the PHYTO and DISSOXYG subroutines—PHYTO

because of the need to model light-growth relationships and nutrient limitation, and DISSOXYG since it is the only system for which a first-order boundary process is a major source or sink.

2.4.4.1 *PHYTO*

The phytoplankton subroutine is called first because it affects all other systems (see section entitled *PROCESS*). First the maximum growth rate is calculated for surface water segments, then adjusted for temperature and ambient light conditions. One thing to note is the presence of a system bypass option. If the phytoplankton system is bypassed in the input file, but the effects of the phytoplankton population still wish to be modeled, a constant saturation growth rate can be modeled by specifying a value for kinetic constant number 41—K1C, the phytoplankton saturation growth rate (day^{-1}).

The phytoplankton growth rate is reduced by ambient light conditions by a call to the Di Toro or Smith light routine. The modeling frameworks developed by Di Toro et al. (1971) and Smith (1980) extend from an earlier formulation by Steele (1962). In the water column, the light intensity to which the phytoplankton are exposed is spatially non-uniform. At the surface and near-surface of the air-water interface, photoinhibition can occur at high light intensities, whereas at sufficient depth, light is significantly attenuated. Photosynthesis will be reduced by natural and algal-related turbidity. Both the Di Toro et al. and Smith formulation account for the effects of photoinhibition by light supersaturation, and light attenuation in the water column.

```

CCHL1 = CCHL
TCHLA = PHYT/CCHL1
KESHD = (0.0088*1000.*TCHLA + 0.054*(1000.*TCHLA)**0.6667)
SKE = KESG (ISEG)
IF (IKE .GT. 0 .AND. IKE .LE. 5) SKE = SKE*KE (IKE)
SKE = SKE + KESHD
TEMP1 = SKE*H
C
C           Get average solar radiation during daylight hours
C
TEMP2 = IAV/IS1
TEMP3 = EXP ( - TEMP1)
IAVBOT=IAV * TEMP3
RLGHTS (ITO, 2) = RLGHTS (ISEG, 2)*TEMP3
RLIGHT = 2.718*FDAY/TEMP1*(EXP ( - TEMP2*
1  TEMP3) - EXP ( - TEMP2))
RLGHTS (ISEG, 1) = RLIGHT

```

Figure 27. Di Toro et al. (1971) Light Formulation

In the Di Toro et al. formulation (Figure 27), a light extinction coefficient from phytoplankton self-shading (K_{ESH} , m^{-1}) is calculated from the chlorophyll concentration in the segment. Extinction due to non-algal light attenuation is subsequently added. Finally, these are combined with the average daily light intensity (IAV, Ly/day) and the phytoplankton saturating light intensity (IS1, Ly/day) specified as constant 47 in the input file to calculate the instantaneous depth-averaged ($H = DEPTHG(ISEG)$, m) growth reduction. The phenomenon of photoinhibition due to supersaturating light intensities is included in the formulation.

$$X_{pl} = \frac{e}{K_a H} fday \left[\exp\left\{ \frac{-I_{ave}}{I_s} \exp(-K_a H) \right\} - \exp\left(\frac{-I_{ave}}{I_s}\right) \right] \quad (2.1)$$

Another important feature of the PHYTO subroutine is the computation of growth rate reduction by nutrient limitation, performed following the light limitation algorithms. The two possibilities for modeling nutrient limitation are the multiplicative approach ($f = \text{growth limitation factor} = f_1 * f_2 * \dots * f_n$), and the minimum factor approach ($f = \min[f_1, f_2, \dots, f_n]$), based on Leibig's law of the minimum. The nutrient limitation option can be specified in the WASP input file by the NUTLIM switch. A value of 1 selects the multiplicative formulation for nutrient limitation. The limitation factors (XEMP1 and XEMP2) are expressed as Monod-type equations:

$$f(N) = \frac{NH3 + NO3}{KMNG1 + (NH3 + NO3)} \quad \text{and} \quad f(P) = \frac{IP * f_d}{KMPG1 + (IP * f_d)} \quad (2.2)$$

Here, *KMNG1* (constant 48, mg-N/L) is the nitrogen half-saturation constant for phytoplankton growth, and *KMPG1* (constant 49, mg PO₄-P/L) is the phosphorous half-saturation constant for phytoplankton growth, and *IP*f_d* represents the inorganic phosphorous in the water column and does not include the fraction of PO₄⁻ that sorbs to solids.

The remaining algorithms in PHYTO compute algal respiration, non-predatory (K1D, day⁻¹), and zooplankton predatory (K1G, L/cell-day) death rates. In the very last statement, all of the phytoplankton sources PHYT*GP1 and sinks PHYT*DP1 (units [M/L³-T) are summed and multiplied by the segment volume (see Figure 28). The resulting units of the mass derivative CD(4, ISEG) are mass per time, and it is calculated in a similar fashion for all other system variables.

```

GITMX1 = K1C*K1T**STP20
GITMAX = GITMX1
:
GIT1 = RLIGHT*GITMAX
:
GP1 = RNUTR*GIT1
:
RESP = K1RC*K1RT**STP20
:
DP1 = RESP + K1D + K1G*ZOO*ZOOSG(ISEG)
:
CD (4, ISEG) = (GP1 - DP1)*PHYT*VOL

```

Figure 28. Summary of PHYTO subroutine

Finally, the reader may recall that the *total* mass derivative for each system variable is added to the existing concentration in the EULER subroutine by multiplying the mass derivative by the time-step (DT), adding this to the old mass (i.e. the mass found in the old segment volume, C(ISYS, ISEG)*VOLOLD), and then dividing this sum by the new segment volume (VOLNEW). It is through this general scheme that new concentrations for system variables are obtained after each time-step:

$$C (ISYS, ISEG) = (MASS + DT*MDER) / VOLNEW$$

2.4.4.2 DISSOXYG

In the discussion of the dissolved oxygen subroutine, we are concerned mainly with the calculation of reaeration, since this involves an interaction with a system boundary, namely the air-water interface. In EUTRO, the user may specify a single reaeration constant (K2, constant 82), or allow the model to calculate reaeration based on the river flow or wind velocity, depending on which value, the flow or the wind-induced, is larger. To illustrate how EUTRO handles reaeration, without entering into considerable detail, we consider the calculation of flow-induced reaeration.

Flow-induced reaeration is calculated for a free-flowing reach, using the Covar (1976) method, which calculates reaeration as a power function of river velocity and hydraulic depth using one of three formulas. Each formula is valid over a particular range of depths and/or river velocities. The Owens et al. (1964) formula is automatically selected for depths below 2 feet. For greater depths, faster moving streams will require the Churchill et al. (1962) formula, while deeper, slower moving rivers will require the O'Connor-Dobbins (1956) formula (Ambrose et al., 1993a). All three formulas take the following form, where V_j and D_j are the stream velocity and depth respectively:

$$K_{qi}(20^\circ C) = REAK(V_j)^{EXPREV}(D_j)^{EXPRED} \quad (2.3)$$

The three parameters *REAK*, *EXPREV* and *EXPRED* vary according to the choice of formula. Figure 29 shows the calculation of flow-induced reaeration in subroutine KAHYDRA. The logic of the selection algorithm is as follows:

- (1) If the average depth (AVEDEPE) is less than 0.61 meters, the Owens formulation is used and the equation parameters are set appropriately.
- (2) If the average depth is greater than 0.61 meters and the river sufficiently slow, the O'Connor-Dobbins formulation is automatically used.
- (3) Finally, if the average depth is greater than 0.61 m, and the velocity is also greater than the threshold value, a positive transition depth is set, signifying the depth at which the river shifts to the O'Connor-Dobbins formulation for the velocity in question.

For the Churchill formulation to be used, the average depth must be less than the transition depth established in step three. After the selection of the appropriate equation, parameters are set, and reaeration is calculated. The last statement in KAHYDRA uses the segment temperature to adjust the temperature-dependent reaeration rate. Finally, back in subroutine DISSOXYG, the calculated wind-induced and flow-induced reaeration rates are compared, and the larger one is selected and adjusted by the percentage of ice cover (XICECVR).

```

AVDEPE = H
AVVELE = ABS(VEL)
C
IF (AVDEPE .GT. 0.61) GO TO 1000
C
C      Use Owen's formulation for reaeration
C
      REAK = 5.349
      EXPREV = 0.67
      EXPRED = - 1.85
      GO TO 1010
C
1000 CONTINUE
C
      IF (AVVELE .GE. 0.518) GO TO 1020
      TRANDP = 0.0
      GO TO 1030
1020 CONTINUE
      TRANDP = 4.411*(AVVELE**2.9135)
1030 CONTINUE
C
      DIF = AVDEPE - TRANDP
      IF (DIF .GT. 0.0) GO TO 1040
C
C      Use Churchill's formulation for reaeration
C
      REAK = 5.049
      EXPREV = .969
      EXPRED = - 1.673
      GO TO 1050
C
1040 CONTINUE
C
C      Use O'Connor-Dobbins formulation for reaeration
C
      REAK = 3.93
      EXPREV = 0.5
      EXPRED = - 1.5
C
1050 CONTINUE
C
1010 CONTINUE
C
C      Calculate reaeration coefficient
C
      K20 = REAK*(AVVELE**EXPREV)*(AVDEPE**EXPRED)
      IF (K20 .GT. 24.) K20 = 24.
      K2HYDRA = K20*1.028**STP20
1060 CONTINUE
      RETURN
      END

```

Figure 29. Calculation of flow-induced reaeration in subroutine KAHYDRA

When the reaeration rate is not calculated by the flow or wind-induced algorithms, three scenarios are possible. First, if the segment is below the water surface, DISSOXYG sets the reaeration rate to zero. If this is not the case, and a constant reaeration rate has been specified by the user, an adjusted value is calculated by making the appropriate temperature and ice-cover calculations. Finally, if the user has opted to use a reaeration time function, the reaeration rate is obtained by multiplying the segment specific reaeration rate constant with the reaeration time function.

Finally, the exchange rate of oxygen across the air-water interface requires the further specification of salinity. The following relationship calculates the natural log of the saturated DO concentration (DO_{sat}) from the water temperature (TK, in Kelvin) and salinity (CL), and a simple manipulation calculates the final exchange rate (see Figure 30).

$$\begin{aligned}
 &RLNCS = - 139.34411 + (1.575701E05/TK) - (6.642308E07/TK**2) + \\
 &1 \quad (1.243800E10/TK**3) - (8.621949E11/TK**4) - \\
 &2 \quad (CL*(3.1929E-02 - (19.428/TK) + (3.8673E03/TK**2))) \\
 C & \\
 &CS = EXP (RLNCS) \\
 C & \\
 &SR190 = KA*(CS - DO)
 \end{aligned}$$

Figure 30. Calculation of oxygen exchange across the air-water interface in DISSOXYG

Additional sources and sinks of oxygen are growth of phytoplankton, CBOD oxidation, and algal respiration. All of these are summed, and the mass derivative is calculated in the same general manner seen before:

$$\begin{aligned}
 &CD (6, ISEG) = (SR190 + SR19PA + SR19PB - SK19P \\
 &1 \quad - SK1913 - SK1918 - SK19S)*VOL
 \end{aligned}$$

2.4.5 EUTRODMP

After computing the kinetic mass derivatives for each system for time-step $T + DT$, if a print interval has been reached, WASPB calls EUTRODMP (IDISK = 1) to store state variables and to print display variable concentrations for the current time. Display variables are variables of interest that are calculated from known information. For example, different nitrogen and

phosphorous concentrations, with algae contributions included, are calculated as shown in Figure 31.

```
TOP = FOP*PCRB*PHYT + OP
TIP = (1.0 - FOP)*PCRB*PHYT + OPO4
TON = FON*NCRB*PHYT + ON
TIN = (1.0 - FON)*NCRB*PHYT + NH3 + NO3
TN = TIN + TON
TP = TIP + TOP
```

Figure 31. Calculation of total nitrogen and phosphorous in EUTRODMP

In the first line of Figure 31, FOP is the fraction of dead phytoplankton phosphorous recycled to the organic phosphorous pool. This is multiplied by the concentration of phytoplankton phosphorous [mg OP/L] (PCRB*PHYT) and added to the existing organic phosphorous concentration. A similar calculation is performed for total organic nitrogen. Finally, to obtain total nitrogen and phosphorous, the organic and inorganic concentrations are summed.

Following the calculation of display variables, a formatted WRITE statement prints their values to the DMP file. Moreover, since EUTRODMP is called within the segment loop of WASPB, display variables are printed for all segments. This essentially marks the end of the WASPB execution sequence, and control is returned to DERIV to continue the mass derivative calculations (see section 2.3.2).

2.5 Conclusion

In this chapter we have sought to develop a basic understanding of the structure of WASP 5.0, its important input, process and kinetic subroutines, and the functional relationships among them. The discussion of the programming details within many of these subroutines is provided on the one hand to familiarize the reader with frequently used variables, and in general how input data are stored and manipulated in WASP. On the other hand, by presenting sequences of executed statements from many of the WASP subroutines, we hope that the reader will have understood at some level of detail “how WASP works,” and have gained an intuition of how changes might be made to alter or expand its capabilities. In Chapter 3, we turn our attention to the exercise of coding a new WASP kinetics module, which involves a number of changes to

existing WASP subroutines, the replacement of others with similar subroutines, and the development of some new subroutines. Here, our discussion will be framed mainly in terms of steps required to fit RWQM1 into the existing WASP structure. Together, Chapters 2 and 3 constitute a primer for students who wish to carry out a similar procedure.

3 Development of RWQMWASP

The implementation of kinetic routines in WASP is nontrivial, and has prompted Di Toro et al. (1983) to present a procedure for developing, programming and debugging a new kinetic submodel. The first step, model development, requires a sufficient understanding of the physical, chemical and biological principles that form the model structure. For the development of RWQMWASP, this step has already been performed with the formulation of the RWQM1 conversion submodel water quality components, the stoichiometric coefficients and process rates. The second and third steps—programming the conversion submodel within WASP and debugging the code—require programming experience in FORTRAN, and will typically involve setting the model print interval equal to the time-step size, and taking a few integration steps in order to check WASP results by hand calculations.

3.1 Outline for Coding a Conversion Submodel

Di Toro et al. (1983) have proposed certain steps for coding a kinetic submodel into WASP. These steps, to be performed not by the conversion submodel developer but by the systems analyst, involve essentially the following:

- (1) Reviewing the kinetic structure and checking for consistent units
- (2) Assigning numbers to each system component
- (3) Assigning numbers to kinetic constants, time functions, and other model parameters to be set by the user (facilitated by the use of EQUIVALENCE statements, with CONST, MFUNC, BFUNC and PARAM arrays)
- (4) Determining which variables need to be initialized upon first entry into WASPB
- (5) If any new piecewise linear time functions are needed by the kinetic routines (that are not already included in EUTRO) developing the code necessary to update slopes and intercepts and compute function values
- (6) Coding the segment loop to evaluate the kinetic portion of the time derivative for each water quality component
- (7) Coding the logic necessary for outputting display variable values computed in WASPB

The basic procedure outlined here was essentially followed to develop RWQMWASP. In the following subsections, this general outline is followed, with emphasis on the major tasks involved in the development of RWQMWASP.

3.2 Kinetic Structure

Before a conversion submodel can actually be implemented in WASP, it is important to identify how, in mathematical terms, the conversion submodel fits into the general scheme describing water quality changes in a river. The fate of substances is due to (1) physical transport and exchange processes and (2) biological, chemical, biochemical and physical conversion processes. These two components are expressed by the following partial differential equation for mass conservation,

$$\frac{\partial \hat{C}}{\partial t} = -u \frac{\partial \hat{C}}{\partial x} - v \frac{\partial \hat{C}}{\partial y} - w \frac{\partial \hat{C}}{\partial z} + \frac{\partial}{\partial x} \left(\varepsilon_x \frac{\partial \hat{C}}{\partial x} \right) + \frac{\partial}{\partial y} \left(\varepsilon_y \frac{\partial \hat{C}}{\partial y} \right) + \frac{\partial}{\partial z} \left(\varepsilon_z \frac{\partial \hat{C}}{\partial z} \right) + \hat{r}(\hat{C}, \hat{p}) \quad (3.1)$$

where \hat{C} is the n-dimensional mass concentration vector for n system variables [M/L³], u , v , and w are the spatial coordinate velocities [L/T], ε_x , ε_y and ε_z are the turbulent diffusion coefficients for the spatial coordinate directions [L²/T], and r is an n-dimensional vector of rates of change of n system variables [M/L³-T] described as a function of the vector p , model parameters subject to calibration.

In principle, equation (3.1) can be considered in three dimensions to resolve the full dynamics of the system. However, in many cases it will be sufficient to follow a conceptual approach, whereby the system is divided into m interconnected, homogeneous tanks, thereby transforming a single n-dimensional partial differential equation into n by m ordinary differential equations. This is the approach used by WASP, as is immediately evident from the calculations in previous sections involving segment-specific values represented by NOSYS by NOSEG dimensional arrays (i.e. C(NOSYS,NOSEG)). The resulting mass balance is expressed by the following vector equation:

$$\frac{d(VC')}{dt} = Q_i C'_i - QC' + V * R'(C', P') \quad (3.2)$$

Now C' , R' and P' are concentration, conversion rate and parameter vectors, different from those given in equation (3.1) and V is the reactor volume. It should also be noted that in the box model approach used here, boundary conditions or interface terms may be formulated by specifying an inward or outward flux (i.e. the NH_3 and PO_4 sediment fluxes in QUAL2), or may be included directly as a sink or source term in the conversion processes vector R' (i.e. reaeration in the DISSOXYG subroutine).

Each of the derivative terms in equation (3.2) has units of mass per time, and is hence a mass derivative. In WASP, the reader will recall that CD(ISYS, ISEG) is the array of mass derivatives used in the kinetic subroutines. Hence, in the development of kinetic subroutines within WASP, it is important to formulate the differential equations for process rates with units that permit simple conversion to mass derivatives. In RWQM1, this has been done. Consider the example of the rate for algal aerobic endogenous respiration:

$$\frac{dX_{ALG}}{dt} = k_{death,ALG,To} e^{\beta_{ALG}(T-T_o)} X_{ALG} \quad (3.3)$$

Here, X_{ALG} has units $[\text{M}/\text{L}^3]$, $k_{death,ALG,To}$, the specific death rate of algae has units $[\text{T}^{-1}]$, and β_{ALG} , the temperature correction factor for algae growth rate has units $(^\circ\text{C}^{-1})$. Thus, the derivative has units $[\text{M}/\text{L}^3\text{-T}]$. In WASP, all process rates are formulated with these same units. To calculate the mass derivative, the process rates are multiplied by the segment volume and then stored in the CD array.

Moreover, in the WASP formalism, concentrations are typically expressed in mg/L, time is expressed in days, and temperature is always in $^\circ\text{C}$. WASP often requires the user to input velocities and flows in units of m/sec and m^3/sec , or otherwise specify conversion factors to convert the specified values. The specification of units is important because a number of internal conversions are performed by WASP to evaluate the mass derivatives. For this reason, these units conventions are kept in RWQM1 to allow as seamless an integration as possible with the existing code.

3.3 Number Assignments for System Variables

Because of the complexity of the RWQM1 conversion submodel, it was often necessary during development to refer back to the RWQM1 report (Reichert et al., 2001b) for reference and clarification. This process was aided by strict adherence to the RWQM1 formalism, which allowed for a very rapid identification of RWQM variables and equations in the code during long coding and debugging sessions. One example of this strict adherence is found in the use of RWQM1 state variable number assignments, shown in Table 3 and discussed in detail in later sections. Additionally, the *names* used to identify both state variables, and model constants tended to follow the RWQM1 naming conventions, with some truncations for variables with longer names.

Table 3. Number assignments for system variables in RWQM1

System Variable	Number Assignment	System Variable	Number Assignment
SS (Dissolved organic substances)	1	SH (Hydrogen ions)	13
SI (Inert dissolved organic substances)	2	SOH (Hydroxyl ions)	14
SNH ₄ (Ammonium)	3	SCa (Calcium ions)	15
SNH ₄ (Ammonia)	4	XH (Heterotrophic organisms)	16
SNO ₂ (Nitrite)	5	XN1 (Stage 1 Nitrifiers)	17
SNO ₃ (Nitrate)	6	XN2 (Stage 2 Nitrifiers)	18
SHPO ₄ (Inorganic dissolved phosphorous)	7	XALG (Algae and macrophytes)	19
SH ₂ PO ₄ (Inorganic dissolved phosphorous)	8	XCON (Consumers)	20
SO ₂ (Dissolved oxygen)	9	XS (Particulate organic material)	21
SCO ₂ (Dissolved carbon dioxide)	10	XI (Inert particulate organic material)	22
SHCO ₃ (Bicarbonate)	11	XP (Phosphate adsorbed to particles)	23
SCO ₃ (Dissolved carbonate)	12	XII* (Particulate organic material)	24

* This second class of particulate organic material was not used in RWQMWASP, but an extension to multiple classes of particulate organic material varying in composition can be easily made.

3.4 Number Assignments for Constants

The kinetic constants in RWQM1 can be organized essentially into three categories based on their location in the conversion submodel: (1) constants that specify elemental mass fractions, (2) constants required for the specification of stoichiometric coefficients, and finally (3) kinetic constants that appear only in process rate equations.

The first category of constants, the elemental mass fractions, identify the mass of N, O, P, C, or H per unit mass of organic matter and can thus be defined only for organic constituents. All members of this class are denoted by the Greek letter α (see Table 1. Elemental Mass Fractions for Organic Compounds):

$$\alpha_{N,X}; \alpha_{O,X}; \alpha_{P,X}; \alpha_{C,X}; \alpha_{H,X}, \text{ where } X = \text{XALG, XS, XH, XN1, XN2, XCON, XI, or SS}$$

The members of the second class of constants are also fractions like those in category 1. However, rather than pertaining to a single constituent, they describe transformations between constituents, and are therefore involved in the specification of stoichiometric coefficients. Examples are FIBAC and YALGD, the fraction of respired heterotrophic and autotrophic mass that becomes inert, and the yield of organic and inert matter resulting from algal death, respectively. Both of these examples relate a particulate organic constituent to a dissolved constituent. Category 2 constants are needed because the biochemical details of the breakdown of particulate organic constituents into a number of other constituents, and also of the growth of organisms by nutrient uptake, are not clearly defined and require the specification of an empirical parameter. Like category 1 constants, each constant in group 2 may be related to a single organic constituent.

The members of the third category of constants are used in the process rate equations. Here, the organization of constants by system variable may not seem entirely logical at first glance, since the kinetic process rates in question will affect multiple systems (see section 2.2.8). However, most of the processes considered in RWQM1 are driven or mediated by organisms such as algae, bacteria and nitrifiers, and thus the kinetic constants in those processes may be grouped with the corresponding system variable. For example, $k_{\text{death,ALG,To}}$, $K_{\text{NO3,H,anox}}$, and

K_{O_2, N_1} are the specific death rate for algae, the saturation growth coefficient for anoxic growth of heterotrophs on nitrate, and the saturation/inhibition coefficient for aerobic endogenous respiration of 1st stage nitrifiers, respectively. Each of these constants has in common the fact that an organism drives the reaction in which the constant is found, either by growing, respiring or dying.

Finally, a small group of kinetic constants cannot be grouped under an organic water quality constituent. This includes constants involved in setting out mass relationships between two inorganic constituents (processes 16 through 21). Since these are chemical equilibrium reactions, it is not entirely clear how the equilibrium constants would be grouped.

However, since most of the kinetics constants are clearly related to one of the organic constituents—XALG, XCON, XH, XN1, XN2, XS, XI, or SS—constants were grouped accordingly and numbers were sequentially assigned from 11 to 138 by EQUIVALENCE statements (see EUTRO.CMN in section 2.1). The constants that were not grouped with one of the above organic components were included in a final set of EQUIVALENCE statements. These assignments were carried out in a file called RWQM.CMN, which we discuss in the next section.

3.5 Developing the Model

The coding of RWQM subroutines came in four major steps: (1) the replacement of the EUTRO common block statements, (2) coding of subroutines to calculate stoichiometry and process rates, (3) coding of subroutines to calculate the mass derivatives, and (4) the modification of WASPB and other WASP subroutines to link the RWQM conversion submodel to the rest of WASP. Since these tasks were performed in more or less the order they are listed, they are considered in this order in the following paragraphs.

3.5.1 The RWQM Common Block

To replace EUTRO.CMN, an RWQM.CMN file was created for the assignment of global memory addresses to variables used in multiple kinetic subroutines. This file is organized essentially into three main sections: (1) variable declarations, (2) common groups, and (3) equivalence groups. In the declarations section, all global variables are declared. Following

the declarations, the variables are assigned to memory addresses within either a common group or an equivalence statement group, depending on whether they are user-specified constants, or unassigned variables.

The variables organized into common groups included stoichiometric coefficients, process rates, and additional kinetics submodel variables. Stoichiometric coefficients set out mass relationships between system variables and the individual processes affecting them, and are based on conservation principles (for a detailed discussion of stoichiometric coefficients, see Reichert et al., 2001b). In RWQM1, stoichiometric coefficients are defined in terms of elemental mass fractions (category 1 constants) and/or yields/yield fractions (category 2). Stoichiometric coefficients thus vary depending on the values specified for the model constants and are unique for those numerical values. However, because model constants are invariable within a single simulation, stoichiometric coefficients need to be calculated only once at the beginning of the simulation (see RWQMINT). Variable names for stoichiometric coefficients are denoted by an S, followed by two numbers indicating first the system variable, and then the process rate. The number assignments to process rates are taken directly from RWQM1. Figure 32 shows one of the common groups for stoichiometric coefficients in RWQM.CMN.

```
COMMON/RWQM2 /
. S11A, S11B, S13A, S13B, S1P15, S31A, S32, S36, S38,
. S310, S313, S34, S35, S39A, S311, S314, S312A, S312B,
. S312C, S312D, S312E, S315, S319, S419, S53A, S53B, S55,
. S57, S61B, S63A, S64, S67, S69B, S69A, S71A, S71B, S72, S76,
. S77, S78, S710, S713, S73A, S73B, S74, S75, S79A, S79B,
. S711, S714, S712A, S712B, S712C, S712D, S712E, S715, S720,
. S722, S723, S820, S91A, S91B, S92, S96, S98, S910, S913, S95,
. S97, S99A, S99B, S911, S914, S912A, S912B, S912C, S912D,
. S912E, S915, S1016, S111A, S111B, S112, S116, S118,
. S1110, S1113, S113A, S113B, S114, S11P5, S117, S119A,
. S119B, S1111, S1114, S1112A, S1112B, S1112C, S1112D
```

Figure 32. Common group for stoichiometric coefficients in RWQM.CMN

The reader will note the two bolded variables in Figure 32. These represent deviations from the naming convention described above, and were necessary because the original convention produces identical variable names for the two coefficients. The first bolded stoichiometric coefficient sets the mass relationship between system 1 and process 15, while the

second coefficient relates system 11 with process 5. The example shown in Figure 32 was the only one of its kind encountered.

The second class of variables assigned to common groups contains process rate variables. Since process rates are calculated at every time-step, the values assigned to these variables change with each time-step. Moreover, process rates must be calculated for each segment since they depend on system variable concentrations, so that new assignments occur NOSEG times at every time-step. Figure 33 shows the common group for rate processes in RWQM.CMN.

```
COMMON/RWQM1 /  
.  PROC1A, PROC1B, PROC2, PROC3A, PROC3B, PROC4, PROC5,  
.  PROC6, PROC7, PROC8, PROC9A, PROC9B, PROC10, PROC11,  
.  PROC12A, PROC12B, PROC12C, PROC12D, PROC12E, PROC13,  
.  PROC14, PROC15, PROC16, PROC17, PROC18, PROC19, PROC20,  
.  PROC21, PROC22, PROC23
```

Figure 33. Common group for kinetic process rates in RWQM.CMN

Process rates have units of mass of a given system component per volume per time. Stoichiometric coefficients essentially convert the mass units from those of one system to those of another. The combination of process rates with stoichiometric coefficients thus ultimately allows the determination of mass derivatives for all systems. For this reason, the computation of stoichiometric coefficients and process rates is kept independent of the calculation of mass derivatives in RWQMWASP. The reasons for this separation are discussed further in the next section.

The last category of variables in common groups includes additional kinetic subroutine variables, shown in Figure 34. The first 22 variables DS through DCO3 will be immediately recognized as names designating RWQM1 system variables. System variable concentrations are assigned to these variables in the WASPB segment loop, and these variables are used in the process rate calculations to improve the readability of the kinetics equations (see section 2.4.1). Variables like SEDSEG, H, VOL and STP20 are used throughout the kinetics submodel in decision algorithms or kinetics calculations. The reader will also recognize a number of variables used specifically in the calculations for reaeration and ambient light conditions (see sections 2.4.4.1 and 2.4.4.2). All remaining variables identify kinetic time functions.

```

COMMON/RWQM6 /
. DS, DO2, DNH4, DNH3, DNO3, DNO2, DHPO4, DH2PO4, ZH,
. ZN1, ZN2, ZALG, ZCON, ZS, ZP, ZI, DCO2, DH, DHCO3, DOH,
. DCA, DCO3, AVDEPE, AVVELE, CFOREA, EXPRED, EXPREV,
. KOREA, REAK, DIF, TRANDP, KA, K20, K2HSAVE, K2WSAVE,
. REAR, KE, TEMP, ITOT, FDAY, WIND, VELN, SALIN, KAINIT,
. AIRTMP, XICECVR, SALFN, ITOTLIM, ITOTMP, SR190, SEDSEG,
. VOL, STP20, VEL, IAV, STP, H, LGHTSW, PHIMX, CCHL, XKC

```

Figure 34. Common group for additional kinetics submodel variables in RWQM1

The last main section of RWQM.CMN contains groups of EQUIVALENCE statements that assign numbers to the kinetic constants and environmental parameters. Most constants are grouped under the following organic constituent names—Stage 1 Nitrifiers, Stage 2 Nitrifiers, Consumers, Algae, Heterotrophs, Dissolved and Particulate Biodegradable Organic Matter, and Dissolved and Particulate Inert Matter. The remaining constants are grouped under the heading Hydrolysis and Equilibrium Constants. Unique number identifiers are assigned for each group (in no particular order within groups), ranging from 11 to 138. Figure 35 shows the equivalence group for algae in RWQM.CMN

```

C=====
C          *-*-* ALGAE CONSTANTS *-*-*
C=====
EQUIVALENCE
. (CONST(54),YALGD), (CONST(55),FIALG), (CONST(56),BETALG)
. , (CONST(57),KGROALGT), (CONST(58),KNALG), (CONST(59),KNH4ALG)
. , (CONST(60),KHPO4ALG), (CONST(61),K1C), (CONST(62), KI)
. , (CONST(63),KRESPALG), (CONST(64),KO2ALG)
. , (CONST(65),KDEATHAL), (CONST(66),ALPCXALG)
. , (CONST(67),ALPHXALG), (CONST(68),ALPOXALG)
. , (CONST(69),ALPNXALG), (CONST(70),ALPPXALG)
. , (CONST(71),ALPXXALG)
C

```

Figure 35. Equivalence statements for constants grouped under Algae

The last set of assignments in RWQM.CMN sets equivalences for environmental parameters. These parameters are specified in Data Group G for each segment, and are stored in an array PARAM(ISEG,ISC). The equivalence statements at the end of RWQM.CMN assign the value of parameter number ISC for segment 1 to the first index of an array NOSEG long, bearing the name of parameter ISC. This initialization is performed mainly to improve the readability of calculations in WASPB involving environmental parameters.

```

C=====
C      *-*-* ENVIRONMENTAL PARAMETER EQUIVALENCES *-*-*
C=====
C
C      EQUIVALENCE
C      .      (PARAM(1,1), VELFN(1)) , (PARAM(1,2), SAL(1))
C      .      , (PARAM(1,3), TMPSG(1)) , (PARAM(1,4), TMPFN(1))
C      .      , (PARAM(1,5), REARSG(1)) , (PARAM(1,4), KESG(1))
C      .      , (PARAM(1,10), RLGHTS(1,1)) , (PARAM(1,13), ITOTLIM(1))
C      .      , (PARAM(1,19), IAVBOT(1)) , (PARAM(1,6), KEFN(1))

```

Figure 36. Environmental parameter equivalence statements in RWQM.CMN

3.5.2 STOICHP and STOICHV

STOICHP is the kinetic subroutine that calculates the values for all stoichiometric coefficients. EUTRO has no analogous subroutine because its stoichiometric coefficients constant and even hardwired into the code. Moreover, because its organic constituents are (1) based on different underlying concepts and (2) lack clear operational definitions, there would be no easy formula for deciding how to change them to reflect changes in organic matter composition (see section 1.3.1). In RWQM1, all organic constituents are defined by their user-specified elemental mass fractions, and the resulting stoichiometric coefficients are unique to the numerical values of those parameters.

In STOICHP, the stoichiometric coefficients are organized by system, which are calculated for all processes affecting a given system. Figure 37 shows the calculation of stoichiometric coefficients for Nitrite (system 5), S53A, S53B, S55, S57.

```

C      *--*--*--*--* System 5 - Nitrite (NO2) *--*--*--*--*
C
C      Anoxic growth of Heterotrophs with NO3
C
      IF (CODXH.NE.0.0) THEN
          S53A = (14.0 * (-1.0 * ONED16 * SP3A1) + (ONED2 * SP3A2) +
1          (ONED6 * SP3A3) + (FIVED28 * SP3A4) + (FIVED62 * SP3A5) +
2          (ONED2 * SP1A8 * SP3A6)) / CODXH
          ELSE
              S53A = 0.0
          ENDIF
C
C      Anoxic growth of Heterotrophs with NO2
C
      IF (CODXH.NE.0.0) THEN
          S53B = (14.0 * (ONED24 * SP3B1) - (ONED3 * SP3B2) -
1          (ONED9 * SP3B3) - (FIVED93 * SP3B5) -
2          (ONED3 * SP1A8 * SP3B6)) / CODXH
          ELSE
              S53B = 0.0
          ENDIF
C
C      Growth of Stage 1 Nitrifiers
C
      IF (CODXN1.NE.0.0) THEN
          S55 = ((1.0 / YN1) - ALPHNXN1) / CODXN1
          ELSE
              S55 = 0.0
          ENDIF
C
C      Growth of Stage 2 Nitrifiers
C
      IF (CODXN2.NE.0.0) THEN
          S57 = (-1.0 * (1.0 / YN2)) / CODXN2
          ELSE
              S57 = 0.0
          ENDIF

```

Figure 37. Calculation of stoichiometric coefficients for Nitrite in STOICHP

Here, the reader will recognize a number of terms that have not previously been discussed. These include the terms beginning SP and COD, and also coefficients like ONED5 and FIVED93. Each of these is considered below in the context of STOICHP.

Assignments are made to all SP terms in subroutine STOICHV, which is devoted solely to the calculation of these variables. SP terms are commonly used expressions found in the definition of many stoichiometric coefficients. For example, SP3A1 through SP3A6, seen in bold in Figure 37, are defined in STOICHV by the following statements:

SP3A1 = (ALPHAOSS / YHANONO3) - ALPHAOXH
 SP3A2 = (ALPHAHSS / YHANONO3) - ALPHAHXH
 SP3A3 = (ALPHACSS / YHANONO3) - ALPHACXH
 SP3A4 = (ALPHANSS / YHANONO3) - ALPHANXH
 SP3A5 = (ALPHAPSS / YHANONO3) - ALPHAPXH
 SP3A6 = (ALPHAXSS / YHANONO3) - ALPHAXXH

Figure 38. Often repeated terms in expressions for stoichiometric coefficients calculated in STOICHV

The terms denoted by SP3A1 through SP3A6 are used to calculate the stoichiometric coefficient relating the anoxic growth of heterotrophs with NO_3^- to System 5 (Nitrite). These terms may be found in at least one other stoichiometric coefficient relating the anoxic growth of heterotrophs with NO_3^- to System 6 (Nitrate). Other expressions may appear in the definitions of as many as six stoichiometric coefficients. The main reason for assigning commonly used expressions to new variables is to reduce computational time, since the calculations in Figure 38 are performed only once. With as many as sixty-five stoichiometric coefficients containing one or more terms seen in the definition of at least one other coefficient, the computational savings can be significant. Moreover, the use of SP terms to denote commonly used expressions also improves the readability of the stoichiometric coefficient assignments in STOICHP.

COD terms convert from units of grams OM to grams COD, and are necessary factors for any stoichiometric coefficient relating an organic constituent to anything else (g OM/ g Y, where Y is any other constituent). COD conversions are seen frequently in the STOICHP subroutine because once again many of the RWQM1 processes are driven or mediated by organic constituents, yielding mass derivatives expressed in the mass units of the organic constituent involved (i.e. g XH). Stoichiometric coefficients then set out mass relationships between these organic constituents and others.

The COD terms are needed to convert g OM (i.e. g XH) to g XH-COD. They are defined at the beginning of STOICHP, and the reader will immediately recognize in Figure 39 the conversion factor of equation (1.3) discussed in section 1.3.1, reproduced again below:

$$\text{g COD} = 32 (\alpha_C/12 + \alpha_H/4 - \alpha_O/32 - \alpha_N/56 + \alpha_P/31) \text{ g OM}$$

All COD conversion factors are expressed this way except that the α terms are specific for the organic constituent in question as shown in Figure 39.

```

C           Defining COD conversion factors
C
      CODSS = 32.0 * (ONED12*ALPHACSS + ONED4*ALPHAHSS - ONED32*ALPHAOSS
1          - THREED56*ALPHANSS + FIVED124*ALPHAPSS + ONED4*ALPHAXSS
2          * SP1A8)
C
      CODSI = 32.0 * (ONED12*ALPHACSI + ONED4*ALPHAHSI - ONED32*ALPHAOSI
1          - THREED56*ALPHANSI + FIVED124*ALPHAPSI + ONED4*ALPHAXSI
2          * SP1A8)
      :
      :
```

Figure 39. Assignments for Organic Matter to COD conversion factors

To understand these COD conversion factors in some more detail, consider again the stoichiometric coefficient relating the anoxic growth of heterotrophs with NO_3^- to System 5 (Nitrite), S53A in Figure 37. The units of the stoichiometric coefficient without the COD conversion factor are grams NO_2 -nitrogen/grams XH, since this stoichiometric coefficient relates the growth of heterotrophs (XH) to the loss of NO_2^- . The CODXH term converts the stoichiometric coefficients to units of grams NO_2 -nitrogen/grams XH-COD, so that the heterotroph concentration may be input in terms of COD mass. The reader should also note the coefficient 14.0, which converts moles NO_2^- to grams NO_2 -nitrogen. This ensures proper conversion of process rates when the NO_2^- concentration is specified by the user in terms of grams NO_2 -nitrogen, as a laboratory assay would yield.

Finally, to complete the discussion of COD conversion factors, a few words should be said about the IF statements in Figure 37. These are included to prevent a divide by zero in the case that one or more of the COD terms are not set. This occurs when the user bypasses one or more of the organic constituents and therefore does not include its elemental mass fractions in the input file. These IF statements may later be omitted if default values are assigned to all elemental mass fractions not specified by the user. The reader may wish to refer to section 3.5.4 for a discussion of how RWQMWASP handles system bypasses.

To conclude our discussion of STOICHP, a few words should be said about fractional coefficients like ONED5 and FIVED93, which simply represent the real fractions 1/5 and 5/93. Fraction coefficients like these, much like the COD and SP terms, are seen multiple times in the definition of RWQM1 stoichiometric coefficients. The fractional coefficients in particular have a basis in the molar masses, molar ratios, and charges of the constituents that make up the organic matter. As before, the primary reason for defining fractional coefficients is to reduce the total number of instructions carried out by the program. The author also believes that establishing these fractions in a separate section of code offers some insight into the scientific basis of the stoichiometric coefficient expressions.

3.5.3 PROCESS

It has been mentioned several times already that many of the processes included in RWQM1 are driven or mediated by dying, growing or respiring organisms. As a result, the process rates have the mass units of the organism involved. For example, the kinetics equation for the growth of algae on ammonia (PROC9A), while it includes concentration terms for nutrients, and a term for ambient light conditions, has overall units of mass of algae per volume per time. To relate the rate of growth of algae to the rate of loss of ammonia or phosphate, stoichiometric coefficients expressed in terms of the nitrogen and phosphorous mass fractions must be applied. These stoichiometric coefficients have units $\text{g NH}_4\text{-N/g XALG-COD}$, and $\text{g HPO}_4\text{-P/g XALG-COD}$, respectively.

The effect of a single process on multiple systems suggests that process rates are best computed independently of mass derivatives, even in the case of the organic systems in terms of which many of the process rates are expressed (i.e. PROC9, $[\text{M algae/L}^3\text{-T}]$). Thus, the process rates would be computed in a separate subroutine at each time-step, and would be included in the segment loop of WASPB. Then, to take the particular case of the algae mass derivative subroutine, PROC9 would be multiplied by an algae-specific stoichiometric coefficient for the mass derivative calculations. In this particular case, a stoichiometric coefficient of 1.0 would be used, since 1 gram of algae is produced for every gram of algae produced.

The use of a separate subroutine to calculate process rates is specific to RWQMWASP. EUTRO avoids a separate process rate subroutine by calculating process rates within the mass

derivative subroutines and assigning these to global variables during the calling sequence of these subroutines. The consequence is that the mass derivative subroutines must be called in a specific order since a process rate used in a later subroutine must first be calculated and assigned to memory in an earlier one (see Figure 40). Moreover, proceeding in this way has the additional result of blurring the relationship between systems and processes, since it diminishes the role of stoichiometric coefficients.

```

C                               Compute derivatives
C                               For PHYT, OP, OPO4, ON, NH3, NO3, CBOD, DO
C
C      CALL PHYTO
C
C      CALL ORGANOP
C
C      CALL INORGANP
C
C      CALL ORGANICN
C
C      CALL AMMONIA
C
C      CALL NITRATE
C
C      CALL CBODSV
C
C      CALL DISSOXYG

```

Figure 40. Calling sequence of mass derivative subroutines in EUTRO

To give an idea of the dependency of later subroutines on earlier ones in EUTROWASP, we consider the death and respiration of algal matter and its effect on different phosphorus species. The phytoplankton death and respiration rate (DPP) is calculated in PHYTO and subsequently used in ORGANOP since some fraction (FOP) of dead and respired algal matter goes into the organic phosphorous pool. But since the remaining fraction (1-FOP) goes into inorganic phosphorous, INORGANP also requires the prior calculation of phytoplankton death and respiration (DPP). Finally, because INORGANP also requires the calculation of organic phosphorous mineralization (SK58), carried out in ORGANOP, it must be called third in the sequence.

RWQMWASP avoids this chain of dependency by calculating all process rates in a separate subroutine called PROCESS. The processes for algae death (PROC11) and aerobic

endogenous respiration of algae (PROC10) are calculated here. As a result, the segment loop mass derivative subroutine calls for SHPO4, XS, and XI can be executed in any order.

There are also a number of interesting parallels between RWQM1 and EUTRO in the phosphorous cycling scheme. These reveal some fundamental differences in the formulation of processes and state variable definitions (see section 1.3). First of all, RWQM1 considers death and respiration separately. Through these processes algal matter is recycled to the inorganic phosphorous or particulate organic (biodegradable or inert) matter pools. In EUTRO, a similar scheme is found. However, not only are separate organic material pools established for nitrogen and phosphorous (ORGANOP and ORGANICN), an artificial distinction at best, but the inert matter pool is nowhere to be found. Moreover, in EUTRO, FOP and (1-FOP) are the only two terms needed to express the fraction of phosphorous going into the organic and inorganic phosphorous pools, respectively.

In RWQM1, a more rigorous formulation is found. Algal respiration yields inert particulate matter (XI) based on the ratio FIALG, the fraction of particulate organic matter that becomes inert during algal death. The remaining biodegradable organic phosphorous ($ALPPXALG - FIALG * ALPHAPXI$) goes into the inorganic phosphorous pool. Algal death, considered separately, also releases particulate organic matter to the inert pool, again expressed by FIALG. However, this time an empirical yield coefficient (YALGD, $g(XS+XI)/gXALG$) must be employed since not all of the dead algal matter will enter these pools. $YALGD * FIALG$ is released to the inert organic matter pool, and $YALGD * (1 - FIALG)$ goes to the biodegradable particulate organic matter pool. Finally, what remains of particulate organic phosphorous after the release of particulate organic matter to the XS and XI pools ($ALPPXALG - (1 - FIALG) * YALG * ALPHAPXS - FIALG * YALGD * ALPHAPXI$) goes to the inorganic phosphorus pool. A similar formulation is employed for nitrogen cycling, with the same inert and biodegradable particulate organic matter pools being involved. Thus, because organic constituents are consistently defined, and elemental mass fractions are employed, the RWQM1 formulation is clearly more robust. Moreover, the important role of stoichiometric coefficients is brought out by this last example, allowing us to understand why the processes rates are better calculated independently of mass derivatives.

To complete the discussion of PROCESS, several more of its salient features are noted. For one, process rates must be calculated in pieces, since the use of some terms depends on prevailing environmental conditions. The code in Figure 41 performs the rate calculation for processes 1a and 1b, aerobic growth of heterotrophs with NH_4^+ and NO_3^- , respectively. The reader will note that the process rates are calculated as a product of several expressions. Several of these expressions are found in other process calculations, a fact that alone justifies their use. However, at least two of the terms, given by K1A5 and K1A7 for kinetic process 1a terms 5 and 7, are important because they are omitted from the kinetic expression if certain bacterial nutrient requirements are met. K1A7, the phosphorous limiting term, can be ignored if the fraction of phosphorous in the dissolved substrate ($\text{YHAER} * \text{ALPHAPSS}$, gP/gSS) is greater than the bacterial fraction of phosphorous. Similarly, if the fraction of nitrogen in the substrate (gN/gSS) is greater than the bacterial nitrogen fraction, the ammonium limitation term in process 1a (K1A5) can be omitted. Moreover, process 1b (growth with NO_3^-) can be omitted altogether. It is essentially in this manner that RWQM1 handles nitrogen and phosphorous limitation.

```

K1A8 = EXP (BETH * STP20)
C
K1A1 = KGROHAER * K1A8
K1A2 = (DS / (KSHAER + DS))
K1A3 = (DO2 / (KO2HAER + DO2))
K1A4 = DNH4 + DNH3
K1A5 = (K1A4 / (KNHAER + K1A4))
K1A6 = DHPO4 + DH2PO4
C
IF (ALPHAPXH .LT. (YHAER * ALPHAPSS)) THEN
  K1A7 = 1.
ELSE
  K1A7 = (K1A6 / (KHPO4HAE + K1A6))
END IF
C
K1B1 = (KNHAER / (KNHAER + DNH4 + DNH3))
K1B2 = (DNO3 / (KNHAER + DNO3))
C
C
IF (ALPHANXH .LT. (YHAER * ALPHANSS)) THEN
  K1A5 = 1.
  PROC1A = K1A1 * K1A2 * K1A3 * K1A5 * K1A7 * ZH
  PROC1B = 0.
ELSE
  PROC1A = K1A1 * K1A2 * K1A3 * K1A5 * K1A7 * ZH
  PROC1B = K1A1 * K1A2 * K1A3 * K1B1 * K1B2 * K1A5 * ZH
END IF

```

Figure 41. Calculation of processes 1a and 1b in subroutine PROCESS

The one other major difficulty encountered in coding subroutine PROCESS was the question of how to model ambient light conditions. Processes 9a and 9b in RWQM1 include the following light limitation term:

$$\frac{I}{K_I} \exp\left(1 - \frac{I}{K_I}\right)$$

where, K_I is the saturation coefficient for growth of algae (essentially IS1, constant 47) and I is the average incident light (IAV). This formulation, which is based on the Steele (1965) light limitation scheme, would require the user to specify IS1, and also the kinetic time functions ITOT (the total daily solar radiation) and F (fraction of day with sufficient light for growth), as before. IAV is then internally calculated. However, since the Di Toro and Smith schemes are expanded versions of the Steele formulation (see section 2.4.4.1), a better approach might be to keep the Di Toro and Smith calculations, and to replace the above non-dimensional expression in PROC9A and PROC9B with the non-dimensional growth reduction factor calculated in either the Di Toro or Smith subroutines. All the light expressions in WASPB would be retained, and those in PHYTO would be transferred to PROCESS. The time function for the time-variable extinction coefficient (KE) would also have to be retained as would all environmental parameters used in the light calculations (see section The RWQM Common Block).

This second option was chosen, and the following lines from PHYTO were copied to PROCESS:

```
IF (LGHTSW .LE. 1.0) call ditoro
IF (LGHTSW .GE. 2.0) call smith
```

PROC9A and PROC9B were multiplied by RLIGHT, the growth reduction factor, to model growth limitation from natural and algal-related turbidity and photoinhibition due to light supersaturation.

3.5.4 Mass Derivatives and SO2

This section is entitled Mass Derivatives and SO2, because we have sought to discuss SO2, the dissolved oxygen subroutine, as a special kind of mass derivative subroutine because it involves a boundary interaction that was not explicitly formulated in RWQM1. All mass derivative subroutines are called in the segment loop exactly as they were in EUTRO. Figure 42 is an excerpt from SO2, showing a number of source and sink rate calculations.

```
C      AAN: Changed DO to DO2 to reflect RWQM nomenclature
C
C      SR190 = KA*(CS - DO2)
C
C      Aerobic growth of Heterotrophs with NH4+
C
1010  DSO21 = S91A * PROC1A
C
C      Aerobic growth of Heterotrophs with NO3
C
      DSO22 = S91B * PROC1B
C
C      Aerobic endogenous respiration of Heterotrophs
C
      DSO23 = S92 * PROC2
C
C      Growth of Stage 1 Nitrifiers
C
      DSO24 = S95 * PROC5
```

Figure 42. Calculation of source and sink rates in SO2

Rates are assigned to local variables with names beginning DSO2, for delta SO2, or the change in the dissolved oxygen concentration per unit time. Each calculation is a product of a stoichiometric coefficient and process rate. Stoichiometric coefficients for oxygen are identified by the number 9 for System 9, and processes are indicated in comments. The same convention is used in all mass derivative subroutines. However, a unique feature of SO2 is the need to calculate a boundary source/sink term, which is formulated like other process rates and denoted in both EUTRO and RWQM1 by the term SR190. The calculation of the oxygen exchange rate across the air-water interface occurs by the same group of statements found in EUTRO's DISSOXYG subroutine, described already in section 2.4.4.2 (see Figure 30).

A system bypass option was required to enable the user of RWQMWASP to avoid explicitly modeling one or more system variables. The system bypass option is formulated at the beginning of all mass derivative subroutines with an IF statement. Figure 43 shows the system bypass option for dissolved oxygen.

```
IF (SYSBY (9) .EQ. 1) THEN
  CD (9, ISEG) = 0
ELSE
```

Figure 43. System bypass option for dissolved oxygen

By setting the mass derivative to zero, the system will undergo no change due to transformation. However, the user may still wish to specify a constant concentration⁷ for a bypassed system so that its effect on other systems can be modeled. This may be done by specifying an initial concentration for the bypassed system in all relevant segments, and then selecting both the system bypass option in Data Group A to bypass the kinetic mass derivative, and the flow bypass option in Data Group D (Record 32) to bypass the flow mass derivative for the system in question. However, it should be noted that Reichert et al. (2001b) set guidelines on how to go about reducing the number of explicitly modeled system variables, and great care must be taken not to bypass systems that are fundamental to the successful modeling of the water quality response in a river (see also Vanrolleghem et al., 2001). For example, dissolved oxygen and dissolved organic matter are considered fundamental to the model, as they appear several times in multiple process rates, in many cases controlling whether the rate reduces to zero. The elemental mass fractions for dissolved organic matter are also seen in countless stoichiometric coefficient expressions, since it serves as an intermediate between particulate organic matter and dissolved inorganic constituents (i.e. it is an important source of nutrients for growth, but also a product of degradation).

When a system is bypassed totally, and its concentration is set to zero, the zero concentration may eliminate one or more processes from the model, and this may be desirable. In the case of bypassed organism systems (XALG, XH, XN1, XN2, and XCON), the elemental mass fractions and other category 2 constants for the bypassed system are no longer needed. If they are retained, irrelevant stoichiometric coefficients are calculated, but the elimination of

⁷ Technically a constant mass is modeled because segment volumes may change.

process rates reduces the resulting source/sink rates to zero. For example, if the algae system is bypassed, and its concentration is set to zero, processes 9a, 9b, 10 and 12a are eliminated. The user need not specify any of the algae constants (all classes) because they only appear in stoichiometric coefficients involving those processes. Elemental mass fractions for algae are used only in stoichiometric coefficients for processes 9a, 9b, 10 and 12a. Although none of these mass fractions appear in denominators, even if they did, they would not invoke a divide by zero if not specified by the user because of how the variables are initialized (see section 3.5.6). Category 2 constants including $f_{I,ALG}$ and $Y_{ALG,death}$ appear only in stoichiometric coefficients for processes 10 and 11 (respiration and death). Thus, an organism constituent's constants appear only in the stoichiometric coefficient expressions involving the processes that have been reduced to zero because of that constituent's bypass. As a result, a total organism system bypass will allow complete bypass of that system's constants. However, it should be noted that when a constant concentration is specified for the bypassed system, all of its category 1, 2 and 3 constants must be specified because the associated process rates are no longer zero. Thus, relevant stoichiometric coefficients must be calculated.

Now we consider the other organic systems. All elemental mass fractions for inert and biodegradable organic matter should also be specified for only the following particulate and dissolved species: XS, SS, and XI. This is because the mass fractions for these different dead organic matter pools are found in stoichiometric coefficient expressions for many different processes as the Table 4 shows. A number of these process rates do not reduce to zero even if the system in question is bypassed and has an initial concentration of zero.

Table 4. Terms whose elemental mass fractions must be specified

Organic Matter	Process stoichiometric coefficients in which elemental mass fractions appear
XS	11,12a-e,14,15
SS	1a,1b,3a,3b,15
XI	2,4,6,8,10,11,13,14

For example, even if XS is bypassed and its initial concentration set to zero, processes 11, 12a through 12e, 14 and 15 would still be modeled (so long as the remaining necessary

parameters have values), and these require the computation of stoichiometric coefficient expressions containing the elemental mass fractions for XS. Consequently, these must still be specified by the user even though the system has been totally bypassed.

Both process and stoichiometric coefficient bypass schemes are not currently included in the model, although a process bypass scheme may be desired. For example, the user may wish to model only aerobic processes if the modeled river or stream is well-aerated. A formal process bypass scheme would allow the user to eliminate all anoxic processes. However, this may seem redundant since an oxygen inhibition effect has already been incorporated into the anoxic process rate equations precisely to account for such conditions. In short, a more robust simulation will allow the model to determine the relative importance of certain processes.

However, it should be noted that in the current version of RWQM_{WASP}, the user may bypass a process simply by not specifying one or more of its kinetic constants, particularly those labeled “maximum” or “specific” growth, death, or respiration rates (see Table 3.4 of Reichert et al., 2001b), whose non-inclusion will reduce a process rate to near-zero, since they are initialized to a near-zero value (see section 3.5.6). This method is not recommended. The choice of “bypassing” a process should be made by the program from concentrations and parameter values. In this case “bypassing” would mean not entirely eliminating the process rate from the simulation, but calculating a low value for the rate. Thus, all relevant kinetic constants should be specified unless it is already clear beforehand that a given system bypass will already eliminate one or more processes, in which case its kinetic constants can be omitted. Clearly, the user will have much more control over an RWQM_{WASP} simulation over a EUTRO one.

Finally, a stoichiometric bypass scheme will be totally undesirable, since it raises concerns over mass conservation. Indeed the very notion is counterintuitive, since processes cannot operate on one system without affecting the other systems involved in the transformation process.

To summarize, the system bypass scheme adopted for the current version of RWQM_{WASP} is driven (1) by the option to bypass mass derivative calculations and (2) by setting a constant or zero concentration for the bypassed system. Because the number of user-specified parameters in RWQM_{WASP} has increased compared to EUTROWASP, the successful user must have a strong working knowledge of RWQM₁, along with a precise understanding of

systems be modeled for a particular application of the program, and consequently what parameters need to be specified. The utility of the system bypass scheme described above may seem doubtful, and such doubt is justified. In fact, an area for future work will be to implement a more logical and user-friendly or “smart” system bypass scheme in RWQMWASP.

One final point should be made about the mass derivative subroutines, and this has to do with the final mass derivative calculation. For SO₂ and the other system variables, this occurs in a statement nearly identical to that found in EUTRO. The difference in RWQM1 is that for many systems, many more source and sink terms are to be found, as Figure 44 clearly shows.

```

CD (9, ISEG) = (DSO21 + DSO22 + DSO23 + DSO24 + DSO25 + DSO26 +
1      DSO27 + DSO28 + DSO29 + DSO210 + DSO211 + DSO212 +
2      DSO213 + DSO214 + DSO215 + DSO216 + DSO217 + DSO218 +
3      DSO219 + SR190) * VOL

```

Figure 44. Dissolved oxygen mass derivative calculation in SO₂

3.5.5 Modifications to WASP

The modifications to the existing WASP code occur primarily in the WASPB subroutine. Additional small changes were made to WASP.CMN to expand the capabilities of the model. The changes in WASPB were elementary, involving primarily the substitution of existing system variable names and subroutine calls with RWQM1 system variables and calls to the RWQM1 mass derivative subroutines. Additional calls to EUTROINT and EUTRODMP were replaced with calls to the RWQMINT and RWQMDMP subroutines, discussed in the next section. The EUTRO kinetic time functions were kept and no additional ones were specified. However, the initialization of several variables required for the zooplankton grazing time-function calculations have been temporarily commented out until the formulation of grazing is worked out in some more detail. Benthic flux calculations, particularly the call to subroutine BENTFLUX have also been commented out, since a primary goal of RWQM1 is to avoid formulating benthic processes as boundary conditions, instead modeling the benthic compartment side-by-side with the water column compartment. Finally, the variable initializations for light-limitation and reaeration calculations were retained.

Changes in WASP.CMN were also elementary, involving simply an increase in the number of model parameters and system variables allowed in WASP. Figure 45 illustrates the changes made to the parameter declarations in WASP.CMN (in bold). The number of maximum allowable systems was increased to 23 and maximum allowable constants to 140 (recall that kinetic constants were assigned numbers from 11 to 138). The result is that arrays initialized in WASP1 having SY or CX indices along one dimension, are now updated to allow the user to select up to 23 systems (NOSYS) and 138 kinetic constants without an error message being called, as seen in Figure 46.

```
PARAMETER (SY=23, SG=300, S2=4*SG/3, CX=140, PR=21, bc=50, WK=50,
. TF=23, MP=50, MB=400, MD=8, MDU=MD*SG, MB1=MB+1, M30=2*BC*SY,
. M50=2*WK*SY, M70=S2+1, M73=2*TF, M75=2*MB, MNF=6, MNI=10, SY1=SY+1,
```

Figure 45. Changes in WASP.CMN to increase the number of allowable systems and kinetic constants

```
IF (NOSYS .GT. SY) THEN
    CALL GETMES (6, 1)
    WRITE (OUT, 6080) SY, NOSYS
6080  FORMAT(//,7X,'Program was Compiled for ',I2,' systems ',
1      'you are trying to run',/,7X,' a simulation with ',
2      I2,' systems. Simulation ABORTED need to recompile',
3      /,' change "SY" in the WASP Common Block to the',
4      'number of systems needed')
```

Figure 46. An error message is called if user-specified NOSYS is greater than SY

3.5.6 RWQMINT and RWQMDMP

RWQMINT and RWQMDMP serve the same role as their EUTRO counterparts. RWQMINT initializes all kinetic constants and some other frequently used variables in WASPB. The conversion of phytoplankton boundary conditions from $\mu\text{g-Chla/L}$ to internal calculational units of mg-Carbon/L was omitted as phytoplankton concentrations in RWQM1 are specified in terms of grams XALG-COD. Other variable WASP calculations were kept, such as the calculation of segment cross-sectional surface area ($\text{BVOL(ISEG)/DEPTHG(ISEG)}$). However, the calculations for benthic NH_4^+ and PO_4^- fluxes were omitted.

Kinetic constants and parameter initializations were also made in RWQMINT, and these constitute of the bulk of RWQMINT initializations. First, all kinetic constants and yield coefficients (categories 3 and 2) were set to very small numbers simply to avoid divide-by-zero errors. Additionally, default values were set for constants used mainly in the light-limitation scheme. These were taken directly from EUTROINT. For example LGHTSW, the light switch, is set to 1.0 to use the Di Toro et al. (1971) light formulation. CCHL and PHIMAX, the other constants shown in Figure 47, are the carbon to chlorophyll ratio (used only when LGHTSW = 1.0) and the maximum quantum yield constant in grams of Carbon per mole of photons (used only when LGHTSW=2.0), respectively. Elemental mass fractions were initialized to very small numbers in the current version of RWQMWASP, although setting these to default values may facilitate the bypassing of organic systems. For example, it would allow the user to input a constant concentration for a bypassed organic system without having to specify its elemental mass fractions. Finally all temperature correction factors were initialized to zero. These initializations are summarized in Figure 47, which gives examples from several initialization types.

```

XARG = ABS(FIBAC)
IF (XARG .LT. ROMIN) FIBAC = 1.00E-20
C
XARG = ABS(YN1)
IF (XARG .LT. ROMIN) YN1 = 1.00E-20
C
XARG = ABS(YN2)
IF (XARG .LT. ROMIN) YN2 = 1.00E-20
:
:
XARG = ABS(IS1)
IF (XARG .LT. ROMIN) IS1 = 300.
CCSC
XARG = ABS(CCHL)
IF (XARG .LT. ROMIN) CCHL = 30.
CCHL1 = CCHL
C
XARG = ABS(PHIMAX)
IF (XARG .LT. ROMIN) PHIMAX = 720.
:
:
XARG = ABS(BETHYD)
IF (XARG .LT. ROMIN) BETHYD = 1.0
C
XARG = ABS(BETCON)
IF (XARG .LT. ROMIN) BETCON = 1.0

```

Figure 47. Initializations in RWQMINT

RWQMDMP performs exactly the same functions as its counterpart in EUTRO. A call to RWQMDMP is made within the segment loop at each print interval from WASPB to store variables and print display variable concentrations for the current time. In RWQMWASP, display variables constitute primarily the system variable concentrations. Additional display variables are segment depth, wind speed, segment temperature, segment water velocity, the wind and flow-driven reaeration rates, and the effective reaeration rate, shown in Figure 48. The last two variables are included to satisfy the following write condition:

```

WRITE (IDMP, 6030) (DNAME (J), J = 1, 6 + (12*NCHEM))
6030 FORMAT (A25)

```

where NCHEM = 2, and the maximum index for J is equal to the number of display variables. This WRITE statement prints these names to the dump file at the beginning of the simulation.

```

DATA DNAME/
  *'Segment Depth  m',           'Temperature      C',
  *'Wind Speed     m/sec',       'Water Velocity  m/sec',
  *'SS              gCOD/L',     'XCON            gCOD/L',
  *'XI              gCOD/L',     'XS              gCOD/L',
  *'SNH4            gN/L',       'Effective KA    day-1',
  *'Wind KA        day-1',       'Hydraulic KA   day-1',
  *'SNH3            gN/L',       'XP              gCOD/L',
  *'SNO2            gN/L',       'SHPO4           gP/L',
  *'SH2PO4         gP/L',       'SO2             gO/L',
  *'SCO2            gC/L',       'SHCO3           gC/L',
  *'SH              gH/L',       'SOH             gH/L',
  *'Light Top Seg  lang./d',     'SCA             gCa/L',
  *'XH              gCOD/L',     'XN1             gCOD/L',
  *'XN2            gCOD/L',     'XALG            gCOD/L',
  *'KSHAER         gCOD/L',     'KNHAER          gN/L' /

```

Figure 48. Display variables in RWQMWASP

The main write statement in RWQMDMP prints the values of the 30 display variables in segment ISEG to the dump file (Figure 49). With each iteration of the segment loop, the values are printed for each segment. The output of segment concentrations to the dump file constitutes the main function of RWQMDMP, and indeed the output file itself is perhaps the most important product of the simulation. Processing and analysis of the dump file ultimately yields information regarding the water quality response of the river to a given load.

```

WRITE (IDMP, 6040) ISEG, TIME, DEPTHG (ISEG),
1  STP, WIND, VEL, DS, ZCON, ZI, ZS, DNH4,
2  KA, K2WSAVE, K2HSAVE, DNH3, ZP, DNO2, DHPO4, DH2PO4,
3  DO2, DCO2, DHCO3, DH, DOH, IAV, DCA,
4  ZH, ZN1, ZN2, ZALG, KSHAER, KNHAER
6040 FORMAT(1X, I5, F12.4, 3X, /, 6 (E11.3) , /, 6 (E11.3) )

```

Figure 49. Display variables are printed to the dump file in RWMDMP

3.6 Conclusion

In this chapter, we have examined the steps thus far completed in the development of RWQMWASP. All essential subroutines have been written and successfully compiled and linked to WASP. Important new subroutines, PROCESS, STOICHP, and STOICHV calculate the segment-specific process rates at each time-step and the stoichiometric coefficients at the beginning of the simulation. A very simple system bypass scheme has been implemented in the

mass derivative subroutines and the problems with it have been noted. The development of an improved, “smarter” system bypass scheme will be an important consideration for future work.

Several tasks remain to be completed, the major ones being model validation, and also the calibration and verification of the site-specific implementations of the simulation program. Operational validation of the RWQMWASP is the assessment of its correctness, that is to say, whether it functions as it should and computes correct values. This may involve testing the simulation results against an analytical solution, and/or hand calculations over a few integration time-steps. The next major task, calibration and verification, is really a set of tasks, which we consider in detail in the next chapter because they bear directly on the predictive ability and therefore usability of site-specific applications of RWQMWASP.

4 Future Work: Model Verification and Parameter Identifiability

A water quality model is the site-specific application of a theoretical construct. It consists essentially of (1) a theoretical construct and (2) the specification of numerical values to model parameters. By this definition, RWQMWASP is not a water quality model but a simulation program whose function is to carry out site-specific applications of the equations that form its theoretical basis, namely RWQM1. In his review article, Beck (1987) gives an overview of the different classes of models, distinguishing between those that try to give a description of the behavior of a system without relying on information about its actual internal workings (*a priori* models), and those that attempt to give an internal description of the system (mechanistic models). RWQM1 doubtless belongs to this second class of models, making considerable use of prior knowledge of the system, and constructed from laboratory and *in situ* field data. Models of this second class are termed mechanistic because they are based on the most recent scientific hypotheses concerning how processes work. In this way, models of this class are very closely tied with the reductionist world-view.

Regardless of type, the development of an appropriate site-specific model requires essentially three steps: (1) parameter estimation or calibration (2) model verification, and (3) model post-audit (Thomann, 1982). Parameter estimation refers to the fitting of a model to a field data set. For mechanistic models, this data set should preferably be different from the data (multiple data sets) used to develop the model. The selection of parameters should be theoretically defensible, for example falling within the range of reported values in the literature. After calibration, the model should be verified, or tested against additional field data and external conditions different from those used to calibrate the model. The importance of using independent field data sets reflecting different conditions cannot be overemphasized. Moreover, calibration and verification are not effective if there is no clear signal against which to model. One problem is that the improvement in water quality in many streams due to decreases in pollutant loading (at least for conventional water quality parameters) has left no clear “signal” against which to calibrate a water quality model. This creates a need to gather data for model calibration during unusual or extreme conditions, when the stream may experience a sudden decrease in water quality (Reichert et al., 2001b and Shanahan et al., 1998).

A model is finally considered verified when the difference between calculated and observed values satisfies a given criterion or the null hypothesis of a statistical test. In the end, values for model parameters should not vary widely between calibration and verification runs, and the modeler must be careful to specify the conditions under which the model can be said to be verified (Thomann, 1982). Oftentimes if a verification fails, the data set ends up being used for calibration, and another data set must be selected for a second verification run. This process is repeated until a true verification is achieved. The last step in model development is post-audit, the subsequent evaluation of the predictive ability of the model by comparing modeled and observed results following implementation of a control strategy (Thomann, 1982; Thomann and Mueller, 1987).

In the following paragraphs, we consider the task of calibration for mechanistic models, and also in the particular case of RWQM1, which belongs to the sub-class of large, overparameterized mechanistic models. The reason for discussing calibration lies in the extreme difficulty of identifying numerical values for parameters in large, overparameterized models.

4.1 Model Parameter Estimation

A number of parameter estimation or calibration problems arise with the development of mechanistic models, primarily because an emphasis on determinism masks the difficulties of accounting for uncertainty. In reality, model parameters are never known for certain. This is especially true in the case of larger models, for which the ability to estimate a unique set of parameters from limited data is severely compromised. As a result, a considerable body of literature has been devoted to treating questions of parameter estimation. Three of the most widely used techniques are considered below.

The first approach makes use of Bayesian methods to estimate a parameter from a prior distribution. In the field of water quality modeling, the Hornberger-Spear algorithm (also called regional sensitivity analysis or RSA), and the generalized likelihood uncertainty estimation method (GLUE) belong to this category of approaches (Hornberger and Spear, 1981, 1983; Freer, 1996). The RSA approach aims to identify those points or regions in the entire parameter space which lead to model outputs that match the data well. However, this method suffers both

from long computational times and also the problem of choosing an appropriate prior distribution.

A second widely applied strategy is to separate the model parameters into those that can be estimated given available data, and those that are not easily estimated and therefore fixed at *a priori* values. This approach ensures estimation of the unknown parameters. The main drawback of this approach is that the grouping of parameters is very much a subjective decision. Moreover, the approach can lead to biased estimates of unknown parameters (Brun et al., 2001).

Another very successful approach is the classical non-linear least squares estimation of parameters. Covariance matrix and likelihood contouring techniques are used to diagnose the quality of parameter estimation. High absolute values for off-diagonal elements of the correlation matrix, and long flat alleys on likelihood contours indicate strong dependencies of the model output on parameter estimates. This means essentially that the effect on model output of changing different parameter values can be self-canceling. In other words, a unique set of parameter values cannot be determined (Brun et al., 2001).

Although the classical non-linear approach has clear merits as a diagnostic tool, there are a number of drawbacks. For one, the correlation matrix is difficult to interpret in the case where more than two parameters are estimated. Similarly, likelihood contours cannot be displayed graphically for functions of more than two variables. Thus when dealing with multiple nonidentifiable parameters, it becomes nearly impossible to compute least squares estimates (Brun et al., 2001).

4.2 Large, Overparameterized Models

A number of large simulation models have appeared in recent years. These are closely tied to the reductionist worldview, and are thought to summarize current scientific knowledge about water quality processes in mathematical terms (Brun et al., 2001). The increase in the complexity of these models (i.e. non-linear interactive eutrophication models), the dramatic increase in the number of system variables (e.g. twenty or more), and the increase in the physical dimensionality of the models (i.e. from 1-D streams to fully three-dimensional estuaries, bays and lakes) have all led to a decrease in the ability of the modeler to comprehend the model output (Thomann, 1982).

Successful modeling of water quality is further impeded by an acute lack of data against which to calibrate and verify the model. The financial and practical burden of conducting multiple field studies under different environmental conditions, and clearly measurable water quality impacts, limits the scope of many field study campaigns. Most data collection studies tend to focus on intensive field studies of short duration. Even when longer campaigns are planned, data collection procedures are hardly ever followed. Instead, *ad hoc* designs, characterized by limited spatial and temporal frequency, fail to develop sufficient spatial and temporal resolution to accommodate the calibration of large, complex, dynamic conversion models (Reichert et al., 2001b and Shanahan et al., 1998).

Because of the limitations discussed above and in the previous section, the goal of parameter estimation for large environmental simulation models is never to identify true parameter values, rather it is to find reasonable parameters to adequately describe the behavior of the system. In large models, many sets of adequate parameter values will be found. The task is then to obtain insight into these adequate values through local and regional sensitivity analyses (Brun et al., 2001).

Classical non-linear techniques fail with multiple non-identifiable parameters, but a combination of the techniques described in the last section may permit a successful calibration to experimental data. For “state of the art” models like RWQM1, too large for unique parameter identification, this calibration will typically involve estimating a subset of parameters selected on the basis of the following criteria, and keeping others fixed: (1) prior knowledge on parameter values, universality, and uncertainty; (2) experimental and initial conditions and measurement layout used for data collection; (3) the measured data itself; and (4) information on the identifiability of model parameters for the given measurement layout based on sensitivity functions (Reichert and Vanrolleghem, 2001). Several techniques for identifiable parameter subset selection based on the above criteria have recently been developed (see Henze et al., 2000 and Brun et al., 2001 for a thorough discussion). These enable the modeler to establish which parameters are practically identifiable under which measurement layouts, and consequently how large the resulting model output uncertainties are expected to be.

4.3 Conclusion

Calibrations for two site-specific implementations of RWQM1 have been performed for the River Glatt in Switzerland and the River Lahn in Germany (Reichert, 2001, and Borchardt and Reichert, 2001). In both cases, the simulation program AQUASIM (Reichert 1994), a proprietary code available from EAWAG in which RWQM1 has been implemented, was used. In the first case, several different submodels of varying levels of complexity were selected from the full conversion submodel to demonstrate how an increase in predictive ability could be achieved by RWQM1 (Reichert, 2001). In the second study, a sediment compartment was included in the model to illustrate the importance of modeling the sediment for a meaningful assessment of river response to wastewater and CSO inputs. With this approach, it was possible to quantify sediment oxygen demand and the spatial and temporal resolution of oxygen-depleted sediment zones (Borchardt and Reichert, 2001).

For RWQMWASP, there is still much to be done before such calibration can be performed. In Chapter 2 we sought to develop an understanding of WASP in preparation for the implementation of RWQM1 in WASP. While all of the essential modifications and additions have been completed, as discussed in Chapter 3, several tasks could still be done. The first major task will be to code a smarter system bypass scheme that will allow the user to successfully implement different submodels of the total RWQM1 conversion model in site-specific applications of the RWQMWASP simulation program. Second, because a more robust representation of benthic processes is possible only by explicit consideration of the benthos as a separate modeling compartment, the next major task will be to determine precisely how to model sediment processes in RWQMWASP. Currently, the second WASP transport field (in Data Group D) specifies the movement of porewater in the sediment bed, also allowing dissolved constituents to exchange between the bed and water column. Solids transport can be modeled also, but the processes of sorption and desorption to the sediment, and therefore sediment settling, scour and sedimentation are not fully represented in RWQMWASP. An accurate representation of benthic processes will require explicit consideration of sedimentation and resuspension of sediment, to which the mediating benthic microbial communities are attached (Reichert et al., 2001b). The last major step before calibration of the model is the operational validation of RWQMWASP. Code validation is simply an assessment of the correctness of the

model, that is to say, whether it computes the right values. This will typically involve checking the model output for a simple problem against an analytical solution, or against hand calculations for a few integration steps.

Finally, in order for RWQM_{WASP} to be finally proven, it must be calibrated and verified in multiple site-specific implementations, against different field data sets. However, the improvement in water quality in many streams due to decreases in pollutant loading or implementation of control strategies has necessitated the calibration of models to extreme pollution events during which there is a clear and transient decline in water quality. Moreover, because RWQM1 is such a large environmental simulation model, summarizing in mathematical form the most current scientific findings, the problem of parameter estimation is critical and must be addressed. Here, trying to estimate all parameters is not feasible, and it is recommended instead that a subset of parameters be selected and those parameters estimated through various techniques. However, to avoid the costly and time-consuming process of data collection and the additional troubles of parameter subset selection and estimation, the author's final recommendation is to use existing data sets and other implementations of RWQM1 to prove RWQM_{WASP} and complete the development of version 1.0 of the WASP implementation of River Water Quality Model No. 1.

5 References

- Ambrose, R. B., Wool, T. A. and Schanz, R. W. (1988) *WASP4, A Hydrodynamic and Water Quality Model—Model Theory, User's Manual, and Programmer's Guide*, Report EPA/600/3-87/039, U. S. EPA, Athens, GA, USA.
- Ambrose, R. B., Wool, T. A., and Martin, J. L. (1993a) *The Water Quality Analysis Simulation Program WASP5, Part A: Model Documentation, Version 5.10*. US EPA, Environmental Research Laboratory, Athens, Georgia.
- Ambrose, R. B., Wool, T. A., and Martin, J. L. (1993b) *The Water Quality Analysis Simulation Program WASP5, Part B: The WASP5 Input Dataset, Version 5.10*. US EPA, Environmental Research Laboratory, Athens, Georgia.
- Beck, M. B. (1987) Water quality modeling: A review of the analysis of uncertainty. *Water Resources Research*. **23**(8), 1393-1442.
- Beck, M. B. and van Straten, G. (eds) (1983) *Uncertainty and Forecasting of Water Quality*. Springer-Verlag, Berlin.
- Borchardt, D., and Reichert, P. (2001) River water quality model no. 1 (RWQM1): Case Study I. Compartmentalization approach applied to oxygen balances in river Lahn (Germany). *Water Science and Technology*. **43**(5), 329-338.
- Brown, L. C. and Barnwell, T. O. (1987) The Enhanced Stream Water Quality Models QUAL2E and QUAL2E-UNCAS: Documentation and User Manual, Report EPA/600/3-87/007, U.S. EPA, Athens, GA, USA.
- Brun, R., Reichert, P. and Künsch, H. R. (2001) Practical identifiability analysis of large environmental simulation models. *Water Resources Research*. **37**(4), 1015-1030.
- Chapra, S. C. (1997) *Surface Water-Quality Modeling*, WCB/McGraw-Hill, New York.
- Churchill, M. A., Elmore, H. L. and Buckingham, R. A. (1962) The prediction of stream reaeration rates. *TN Report*. Tennessee Valley Authority, Chattanooga, TN.
- Connolly, J. P. and Winfield, R. (1984) A User's Guide for WASTOX, a Framework for Modeling the Fate of Toxic Chemicals. In *Aquatic Environments. Part 1: Exposure Concentration*. Report EPA-600/3-84-077, U.S. EPA, Gulf Breeze, FL.
- Covar, A. P. (1976) Selecting the proper reaeration coefficient for use in water quality models. In Proceedings U.S. EPA Conference on Environmental Simulation Modeling, April 19-22, 1976. Cincinnati, Ohio.

- Di Toro, D. M., O'Connor, D. J., and Thomann, R. V. (1971). A dynamic model of the phytoplankton population in the Sacramento San Joaquin Delta. In *Advances in Chemistry Series, No. 106*, pp. 131-180, American Chemical Society, Washington, D.C.
- Di Toro, D. M., Fitzpatrick, J. J. and Thomann, R. V. (1983) *Water Quality Analysis Simulation Program (WASP) and Model Verification Program (MVP) - Documentation*. Hydroscience, Inc., Westwood, NY, for U.S. EPA, Duluth, MN, Contract No. 68-01-3872.
- Dobbins, W. E. (1964) BOD and oxygen relationships in streams. *Proceedings of American Society of Civil Engineers, Journal of Sanitary Engineering Division* **90**(SA3), 53-78.
- Elser, J. J., Chrzanowski, T. H., Sterner, R. W., Schampel, J. H. and Foster, D. K. (1995) Elemental ratios and the uptake and release of nutrients by phytoplankton and bacteria in three lakes of the Canadian shield. *Microbial Ecology*. **29**(2), 145-162.
- Freer, J., Beven, K. and Ambroise, B. (1996) Bayesian estimation of uncertainty in runoff prediction and the value of data: An application of the GLUE approach. *Water Resources Research*. **32**(7), 2161-2173.
- Gromiec, M. J., Loucks, D. P. and Orlob, G. T. (1983) Stream Quality Modeling. In *Mathematical Modeling of Water Quality*. (ed G. T. Orlob), pp 176-226, Wiley, Chichester.
- Henze, M., Gujer, W., Mino, T. and van Loosdrecht, M. (2000) *Activated Sludge Models ASM1, ASM2, ASM2d and ASM3*. Scientific and Technical Report No. 3, IAWQ, London.
- Hornberger, G. M. and Spear, R. C. (1981) An approach to the preliminary-analysis of environmental systems. *Journal of Environmental Management*. **12**(1) 7-18.
- Hornberger, G. M. and Spear, R. C. (1983) An approach to the analysis of behavior and sensitivity in environmental systems. In *Uncertainty and Forecasting of Water Quality*. (eds M B. Beck and G. van Straten), pp. 101-116, Springer-Verlag, Berlin.
- Lesouef, A., Payaraudeau, M., Rogalla, F. and Kleiber, B. (1992) Optimizing nitrogen removal reactor configurations by on-site calibration of the IAWPRC activated sludge model. *Water Science and Technology*. **25**(6), 105-123.
- Metcalf and Eddy (2001) *Advanced Page Proof of: Wastewater Treatment and Reuse*. McGraw Hill, Dubuque, Iowa.
- Norland, S., Fagerbakke, K. M. and Heldal, M. (1995) Light element analysis of individual bacteria using X-ray microanalysis. *Applied and Environmental Microbiology*. **61**(4), 1357-1362.
- O'Connor, D. J. and Di Toro, D. M. (1968) The distribution of dissolved oxygen balance in streams. *Water Resources Research*. **4**(3), 639-646.

- O'Connor, D. J. and Di Toro, D. M. (1970) Photosynthesis and oxygen balance in streams. *Proceedings of American Society of Civil Engineers, Journal of Sanitary Engineering Division*. **96**(SA2), 547-571.
- O'Connor, D. J., and Dobbins, W. E. (1956). The mechanism of reaeration in natural streams. *Proceedings of American Society of Civil Engineers, Journal of Sanitary Division*. **96**(SA2), 547-571.
- Orlob, G. T. (ed) (1982) *Mathematical Modeling of Water Quality*. Wiley, Chichester.
- Owens, M., Edwards, R. W. and Gibbs, J. W. (1964) Some reaeration studies in streams. *International Journal of Air and Water Pollution*. **8**, 469-486.
- Rauch, W., Henze, M., Koncsos, L., Reichert, P., Shanahan, P., Somlyódy, L. and Vanrolleghem, P. (1998) River water quality modelling: I. State of the art. *Water Science and Technology*. **38**(11), 97-114.
- Reichert, P. (1994) AQUASIM—A tool for simulation and data analysis of aquatic systems. *Water Science and Technology*. **30**(2), 21-30.
- Reichert, P. (2001) River water quality model No. 1 (RWQM1): Case Study II. Oxygen and nitrogen conversion processes in the River Glatt. *Water Science and Technology*. **43**(6), 329-338.
- Reichert, P. and Vanrolleghem, P. (2001) Identifiability and uncertainty analysis of aquatic systems. *Water Science and Technology*. **43**(7), 329-338.
- Reichert, P., Borchardt, D., Henze, M., Rauch, W., Shanahan, P., Somlyódy, L. and Vanrolleghem, P. (2001a) River Water Quality Model No. 1 (RWQM1): II. Biochemical process equations. *Water Science and Technology*. **43**(5), 11-30.
- Reichert, P., Borchardt, D., Henze, M., Rauch, W., Shanahan, P., Somlyódy, L. and Vanrolleghem, P. (2001b) River Water Quality Model No. 1, *Scientific and Technical Report No. 12*, IWA Publishing, London.
- Roesner, L. A., Giguere, P. R. and Evenson, D. E. (1981) *Computer Program Documentation for the Stream Quality Model QUAL-II*. Report EPA 600/9-81-014, US EPA, Athens, GA.
- Shanahan, P., Henze, M., Koncsos, L., Rauch, W., Reichert, P., Somlyódy, L. and Vanrolleghem, P. (1998) River water quality modeling: II. Problems of the art. *Water Science and Technology*. **38**(11), 245-252.
- Shanahan, P., Borchardt, D., Henze, M., Rauch, W., Reichert, P., Somlyódy, L. and Vanrolleghem, P. (2001) River Water Quality Model no. 1 (RWQM1): I. Modelling approach. *Water Science and Technology*. **43**(5), 1-9.

- Smith, R. A. (1980) The theoretical basis for estimating phytoplankton production and specific growth rate from chlorophyll, light and temperature data. *Ecological Modeling*. **10**, 243-264.
- Somlyódy, L., Henze, M., Koncos, L., Rauch, W., Reichert, P., Shanahan, P. and Vanrolleghem, P. (1998) River water quality modeling: III. Future of the art. *Water Science and Technology*. **38**(11), 253-260.
- Spanjers, H. and Vanrolleghem, P. A. (1995) Respirometry as a tool for rapid characterization of wastewater and activated sludge. *Water Science and Technology*. **31**(2), 105-114.
- Steele, J. H. (1962) Environmental control of photosynthesis in the sea. *Limnology and Oceanography*. **7**, 137-150.
- Streeter, W. H. and Phelps, E. B. (1925) *A study of pollution and natural purification of the Ohio River*, Public Health Bulletin No. 146, U. S. Public Health Service, Washington D. C.
- Stumm, W. and Morgan, J. J. (1981) *Aquatic Chemistry*, Wiley, New York.
- Thomann, R. V. (1982) Verification of water quality models. *Journal of Environmental Engineering Division*, ASCE. **108**(EE5), 923-940.
- Thomann, R. V. and Mueller, J. A. (1987) *Principles of surface water quality modeling and control*, Harper & Row, New York.
- Thomas, H. A. (1948) Pollution load capacity in streams. *Water and Sewage Works*. **95**, 409.
- TWDB (1971) Simulation of Water Quality in Streams and Canals, Theory and Description of QUAL-1 Mathematical Modeling System. Report 128, Texas Water Development Board, Austin, TX.
- Vanrolleghem, P. A., Spanjers H., Petersen, B., Ginestet, P. and Takacs, I. (1999) Estimating (combinations of) Activated Sludge Model No. 1 parameters and components by respirometry. *Water Science and Technology*. **39**(1), 195-214.
- Vanrolleghem, P., Borchardt, D., Henze, M., Rauch, W., Reichert, P., Shanahan, P. and Somlyódy, L. (2001) River Water Quality Model No. 1 (RWQM1): III. Biochemical submodel selection. *Water Science and Technology*. **43**(5), 31-40.