SMHI

Nr 91, 2003

Hydrologi

BIOLA

Biogeochemical Lake Model Manual

Charlotta Pers

Hydrologi
Nr 91, 2003

BIOLA
Biogeochemical Lake Model
Manual

Charlotta Pers

Report Summary / Rapportsammanfattning

Issuing Agency/Utgivare	Report number/Publikation
Swedish Meteorological and Hydrological Institute SE-601 76 NORRKÖPING Sweden	Hydrologi nr 91 Report date/Utgivningsdatum November 2003

Author (s)/Författare
Charlotta Pers

Title (and Subtitle)/Titel

BIOLA – Biogeochemical Lake Model - Manual

Abstract/Sammandrag

The biogeochemical lake model, BIOLA, was developed at SMHI for eutrophication studies in Sweden. The model is a one-dimensional (vertical) model that simulates temperature stratification and water quality influenced by weather, inflow and biogeochemical processes in the lake and in the sediments. The central variables simulated are inorganic nutrients and phytoplankton. It may simulate the biogeochemical effects in a lake from a changed environment. The changes may be eutrophication remedy actions, thus when considering different management actions BIOLA may contribute with valuable information.

The Manual describes the BIOLA code. The BIOLA code consists of a couple of subroutines in four Fortran-files to be used together with the equation solver PROBE. The report describes how to use these subroutines to simulate the biogeochemistry in a lake, but only briefly how to use PROBE. The Manual describes data required, different ways to incorporate nutrient loads to the lake, how meteorological data is incorporated etc. It also describes how to change parameter values of the biogeochemical processes included in the model and calibrate the model to a specific lake.

Key words/sök-, nyckelord
biogeochemistry, lake modelling, eutrophication

Supplementary notes/Tillägg	Number of pages/Antal sidor	Language/Språk
	121	English

ISSN and title/ISSN och titel

0283-7722 SMHI Hydrologi

Report available from/Rapporten kan köpas från:

SMHI
SE-601 76 NORRKÖPING
Sweden

Contents

Appendix A. The equations of the biogeochemical processes

Appendix B. Example of a BIOLA CASE subroutine

Appendix C. The code of BIOLA

1 Introduction

1.1 Purpose of the Manual

This manual is intended to provide assistance when using the BIOLA-PROBE computer code. This manual only briefly explains the PROBE code and the PROBE Manual (Svensson, 1998) is strongly recommended to be used in parallel. The use of BIOLA-PROBE demands some knowledge in programming in FORTRAN, an editor and a compiler for this language. The manual assumes the user has knowledge about editing, compiling, linking and running a FORTRAN program.

The manual concentrates on describing BIOLA-PROBE used for a single basin lake model. The manual contains only a short description of how to apply BIOLA to several subbasins. For this application, studies of earlier CASE Reports are recommended (e.g. Svensson, 1986). The CASE Report Series contains reports of basic fluid dynamics, heat and mass transfer applications as well as applications of PROBE in oceanography, hydrology and meteorology. A list of all CASE reports can be found in (Svensson, 1998).

1.2 General features of BIOLA and PROBE

PROBE (PROgram for Boundary layers in the Environment) is an equation solver for one-dimensional transient, or two-dimensional steady, boundary layers. In the one-dimensional form it can be used to simulate the vertical dimension of a lake. The turbulent mixing of the lake is modelled with a two-equation model, the k-ε model, which calculates mixing coefficients. These are combined with two momentum equations to form the hydrodynamical part of PROBE. For lake modelling with BIOLA an equation for heat energy is also used in PROBE.

The BIOLA (BIOgeochemical LAke) model contributes with 14 transport equations, one for each biogeochemical state variable. These are solved by PROBE at the same time as the basic PROBE equations. Actually BIOLA only provides the source and sink terms of the equations, while the rest is handled by PROBE. The BIOLA state variables are thus mixed with the same field of turbulence as momentum.

BIOLA-PROBE is structured so that the user will most often only be concerned with one subroutine (called `CASE`) and one parameter file. PROBE is written in standard FORTRAN 77, but the BIOLA code includes some FORTRAN 90 features.

1.3 Outline of the Manual

After this brief introduction the next section describes the BIOLA model, which state variables and processes that are included, and for which applications the model can be used. The next section deals with necessary and optional data for setting up BIOLA for a lake. The thereafter following sections describe the BIOLA code, how to handle the code during set up of the model and how to calibrate BIOLA. In the Appendices are found a summary of all BIOLA source and sink equations, an example of a `CASE` subroutine and the BIOLA code.

In the manual, the `Courier` font is used for words (subroutines, variable names etc.) used in the code and in examples of the code.

2 BIOLA Model

2.1 The general BIOLA equation

The biogeochemical module is build to be used with PROBE. The transport equations (1-2) are simulated within PROBE, with biological and chemical processes as sink and source terms.

$$\frac{\partial C}{\partial t}+w\frac{\partial C}{\partial z}=\frac{1}{A}\frac{\partial}{\partial z}\left(A\nu\frac{\partial C}{\partial z}\right)+\Phi+\phi+\frac{1}{A}\frac{\partial}{\partial z}(C_{in}Q_{in}) \quad (1)$$

change + advection = diffusion + source + sink + inflow

$$\frac{\partial C}{\partial t}=\Phi+\phi+\frac{1}{A}\frac{\partial}{\partial z}(C_{in}Q_{in}) \quad (2)$$

Sources and sinks are denoted Φ and ϕ respectively. Note that the sinks (ϕ) are negative. These are the sum of the result of the processes modelled by BIOLA. C denotes concentration, C_{in} denotes concentration of inflow, Q_{in} is inflow, z is vertical coordinate, t is time, w is vertical velocity, A is lake horizontal area (varying with depth), and ν is turbulent eddy viscosity. For some variables (e.g. sediment) advection and diffusion is zero, and the transport equation is reduced to (2).

The continuous equations (1-2) are discretised in BIOLA-PROBE. The vertical axis is divided in discrete cells with increasing area (Figure 1). Each state variable has one value for each cell. Time is divided into time steps of typically 10 minutes (600 seconds), because this time step length is often sufficient for a stable solution of the hydrodynamical equations. Smaller time steps may be used if necessary. SI units are used (meter, second, kilogram, Kelvin) with the exceptions that temperature is given in °C and heat thus has the reference point 0°C, and that in BIOLA gram is used instead of kilogram (concentrations are calculated in g m^{-3} which is the same as mg L^{-1} a suitable unit for the state variables of BIOLA).

2.2 Biogeochemical state variables

The biogeochemical model encompasses the elements carbon (C), nitrogen (N), phosphorus (P) and oxygen (O). Totally 14 state variables are included in the model (Table 1). Inorganic

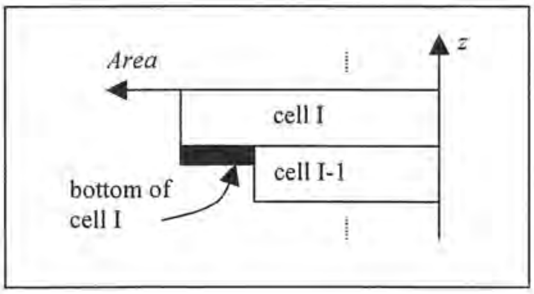

Figure 1. Illustration of cell grid in PROBE and BIOLA.

Table 1. State variables included in BIOLA, with symbol used in Figure 3.

Inorganic variables	Symbol	Unit
Phosphate	PO4	mg P L^{-1}
Dissolved inorganic nitrogen	DIN	mg N L^{-1}
Oxygen gas	O2	mg O_2 L^{-1}
Dissolved phosphorus in sediment water	BIP	g P m^{-2}
Dissolved ammonium in sediment water	BNH	g N m^{-2}
Dissolved nitrate in sediment water	BNO	g N m^{-2}

Living organic variables	Symbol	Unit
Phytoplankton	A	mg C L^{-1}
Cyanobacteria	CB	mg C L^{-1}
Zooplankton	Z	mg C L^{-1}
Planktivorous fish	FA	mg C L^{-1}
Piscivorous fish	FB	mg C L^{-1}
Macrophyte	M	mg C L^{-1}

Dead organic variables	Symbol	Unit
Detritus	D	mg C L^{-1}
Sediment organic matter	B	g C m^{-2}

carbon is not modelled; neither are nitrogen, phosphorus and oxygen bound in organic matter. Nitrogen, phosphorus and oxygen are assumed to be a constant fraction of the organic matter. The relation from the photosynthesis reaction is used (C:O:N:P=106:106:16:1), although the stoichiometric ratio does actually vary for phytoplankton, due to e.g. different nutrient conditions (Correll, 1998), and between different organic matter (Elser and Hassett, 1994; Hessen et al., 2003).

The state variables are concentrations (Table 1). Sediment variables are calculated per unit sediment area, while dissolved and particulate variables in the water are calculated by volume. Sediment variables are not subjected to advection and diffusion in the water. For sediment variables the value for cell I denote the sediment at the bottom of that cell (Figure 1). For the other variables, except macrophyte, the value for cell I denotes the concentration in cell I. Since macrophytes are not free-living, but attached to the bottom, their concentration for cell I denotes the concentration in the whole water column above the bottom of cell I (Figure 2). The macrophyte variable is thus assumed uniformly distributed in the water column. They are not subjected to the advection and diffusion calculated by PROBE. The special treatment of the macrophyte variable gives a horizontal variation in macrophyte concentration in the model.

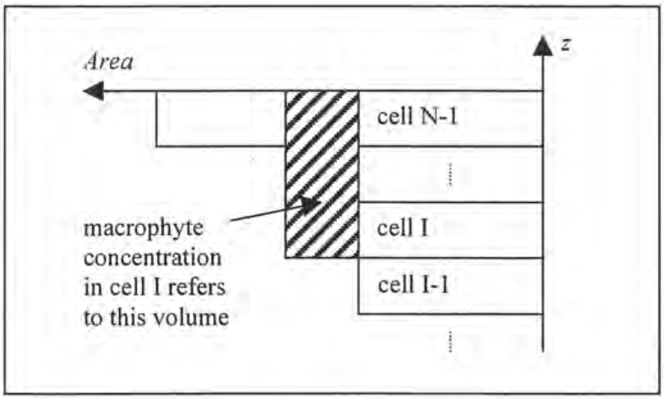

Figure 2. Illustration of macrophyte cell. N-1 is the uppermost cell.

The oxygen variable is allowed to become negative. Negative oxygen denotes presence of hydrogen sulphide, demanding a corresponding amount of oxygen to be oxidised. It is not necessary to include all variables when simulating a lake. Macrophyte and fish variables have been excluded in case studies (Pers and Persson, 2003).

2.3 Modelled biogeochemical processes

The modelled processes give the sources and sinks of the model equations (1-2). They can be seen as flows between state variables (Figure 3). The process equations are given in Appendix A. Table 2 gives an overview of the internal processes modelled and the variables they influence. Totally 44 parameters are used in the process equations. These are given in Table 3 with both the symbol used in the model description (Pers, 2002 and Appendix A) and the parameter name used in the code. In Table 3 is also included a description of the parameter, its default value, and maximum and minimum values found in literature.

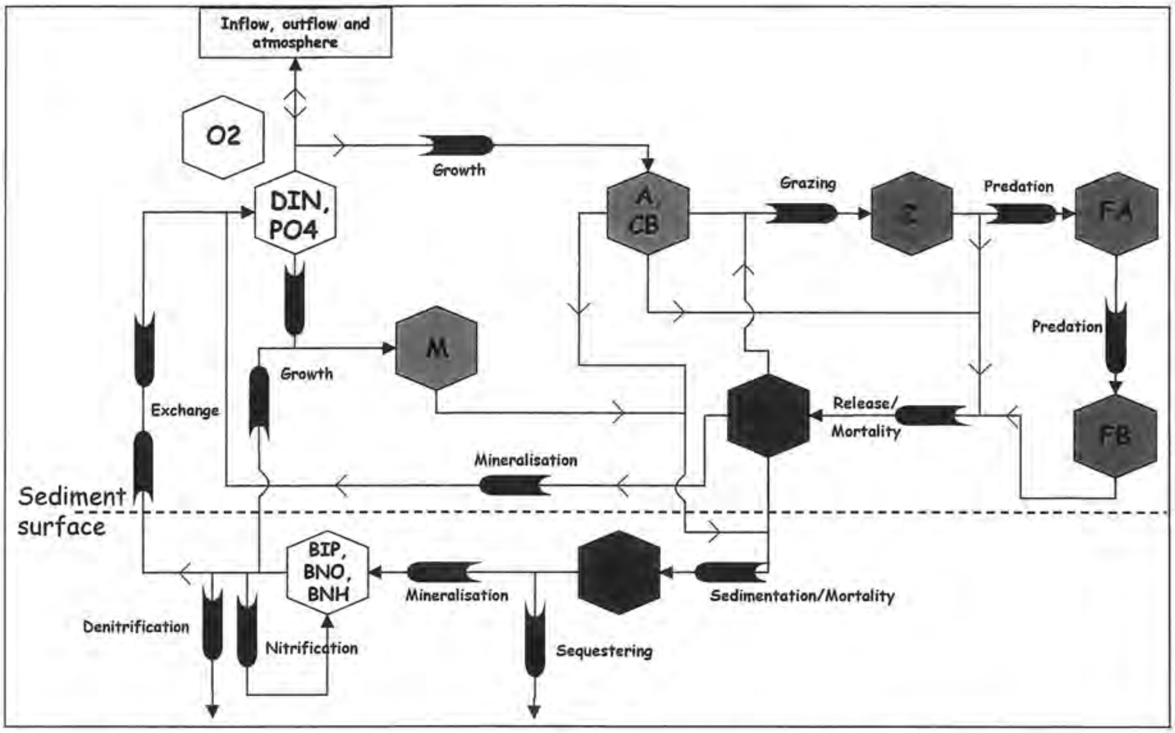

Figure 3. State variables and processes of the biogeochemical model BIOLA.

2.4 Field of application

The BIOLA-PROBE model was developed for eutrophic lakes in Sweden. In addition its application assumes a horizontally homogeneous lake. The model will not be able to account for horizontal variation effects caused by e.g. varying climate, elongated lake or a lake with creeks and sounds. The model has been applied to an oligotrophic lake with limited success (Pers and Persson, 2003).

The purpose of the BIOLA-PROBE model is to simulate the effect of different eutrophication scenarios on the biogeochemical variables in the lake. Changed environment or supply of nutrient affects the lake in different ways, and the model simulates the complex interactions between different processes to find possible effects on the modelled state variables.

Table 2. Modelled internal processes and their influence on the state variables.

Process	Sink (ϕ) of	Source (Φ) of
Phytoplankton growth	PO4, DIN	A, O2
Blue-green algae growth including nitrogen fixation	PO4, DIN	CB, O2
Macrophyte growth	BIP, BNH, PO4, DIN	M, O2
Natural mortality of autotrophs	CB, M	D, B
Zooplankton grazing	A, CB, D	Z, D
Natural mortality of zooplankton	Z	D
Predation on zooplankton	Z	FA
Predation on planktivorous fish	FA	FB
Natural mortality of fish	FB	D
Mineralisation of detritus	D, O2	PO4, DIN
Denitrification	B, BNO, D, DIN	BNH, BIP, DIN, PO4
Nitrification	BNH, O2	BNO
Mineralisation in sediment	B, O2	BNH, BIP
Permanent sequestering in sediment	B	
Exchange of nutrients between water and sediment	BIP, PO4, BNH, BNO, DIN	BIP, PO4, BNH, BNO, DIN
Sinking and sedimentation	A, CB, D	A, CB, D, B

Table 3. Parameters that can be changed in BIOLA, with the BIOLA default value and minimum and maximum values found in literature (from Pers (2002)).

Symbol	Name in code	Parameter	Default value	Minimum value	Maximum value
μ_A	MYMAX	maximum growth rate for phytoplankton at reference temperature (d^{-1})	2.0	0.6	10
	TETAPHY	temperature coefficient for phytoplankton growth	1.07	1.07	1.17
	TPHY	reference temperature for phytoplankton growth (°C)	20	-	-
I_{opt}	IOPTA	optimal light intensity for phytoplankton and cyanobacteria ($W\ m^{-2}$)	50	25	193
k_{PO4}	KPO4	half-saturation concentration for phytoplankton and cyanobacteria uptake of phosphorus ($mg\ P\ L^{-1}$)	0.01	0.00003	0.19
k_{DIN}	KDIN	half-saturation concentration for phytoplankton and cyanobacteria uptake of nitrogen ($mg\ N\ L^{-1}$)	0.02	0.005	0.08
k_w	KW	extinction coefficient of the water (m^{-1})	0.04	0	1
k_M	KMKONST	macrophyte shading coefficient ($m^2\ g^{-1}$)	0.16	0.002	0.16
k_P	KPKONST	shading coefficient for phytoplankton and cyanobacteria ($m^2\ g^{-1}$)	0.3	0.0004	2.3
k_D	KDKONST	detritus shading coefficient ($m^2\ g^{-1}$)	0.2	0.03	0.4
	IZKONST	part of short-wave radiation that penetrates into water	0.5	-	-

Table 3. Continued.

Symbol	Name in code	Parameter	Default value	Minimum value	Maximum value
μ_{ANFIX}	MYMAXA	maximum growth rate for cyanobacteria using DIN at reference temperature (d^{-1})	1.0	0.42	2
μ_{ANFIX}	MYMAXB	maximum growth rate for cyanobacteria using N_2 at reference temperature (d^{-1})	1.0	0.42	2
	KNFIX	Redfield ratio N:P in organic matter	7.2	-	-
	SNF4	coefficient in nitrogen fixation equation	4	-	-
μ_M	MYMAXM	maximum growth rate for macrophytes at reference temperature (d^{-1})	0.5	0.11	10
θ_M	TETAMAC	temperature coefficient for macrophytes	1.04	-	-
	TMAC	reference temperature for macrophyte growth (°C)	20	-	-
I_{Mopt}	IOPTM	optimal light intensity for macrophyte (W m^{-2})	75	-	-
k_{MIP}	KMIP	half-saturation concentration for macrophyte uptake of phosphorus (mg P L^{-1})	0.005	-	-
k_{MIN}	KMIN	half-saturation concentration for macrophyte uptake of nitrogen (mg N L^{-1})	0.01	-	-
d_{sed}	DSED2	thickness of active sediment layer (m)	0.1	-	-
m_{ANFIX}	XMRT	mortality rate for cyanobacteria (d^{-1})	0.0	-	-
m_M	MMRT	mortality rate for macrophytes (d^{-1})	0.05	-	-
k_G	GAMMA	grazing rate (d^{-1})	0.8	0.2	2.5
e	EFFI	efficiency of zooplankton grazing	0.4	-	-
k_g	KG	half-saturation concentration for grazing (mg C L^{-1})	0.5	-	-
s_A	P1	selectivity coefficient for grazing on phytoplankton	1.0	-	-
s_{ANFIX}	P3	selectivity coefficient for grazing on cyanobacteria	0.2	-	-
s_D	P2	selectivity coefficient for grazing on detritus	0.4	-	-
k_{zm}	ZMRT	mortality rate for zooplankton (d^{-1})	0.005	-	-
k_{pA}	PREDZ	maximum predation rate on zooplankton at reference temperature (d^{-1})	0.1	0.05	0.15
k_Z	KZO	half-saturation concentration for zooplankton (mg C L^{-1})	1.5	-	-
θ	TETAG	general temperature coefficient	1.07	1.07	1.17
T_{ref}	TREFG	general reference temperature (°C)	20	-	-
	OMEGA	part of fish predation that is used for growth (both fishes)	1	-	-
k_{pB}	PREDF	maximum predation rate on planktivorous fish at reference temperature (d^{-1})	0.1	0.05	0.13
k_{FA}	KFIA	half-saturation concentration for fish (mg C L^{-1})	0.2	-	-
k_{mort}	FMRT	mortality rate for piscivorous fish (d^{-1})	0.0008	0.0001	0.015
k_d	DDCC	degradation rate at reference temperature (d^{-1})	0.02	0.00005	0.4
k_{dena}	ADEN	maximum denitrification rate per unit area at reference temperature (g N m^{-2} d^{-1})	0.16	0.03	1.2
	TETADEN	temperature coefficient for denitrification	1.07	-	-
	TDEN	reference temperature for denitrification (°C)	20	-	-
k_{den}	BDEN	maximum denitrification rate at reference temperature (mg N L^{-1} d^{-1})	0.03	0.01	0.24
K_m	KBNO	half saturation concentration for nitrogen used in denitrification (mg N L^{-1})	3	0.1	84

Table 3. Continued.

Symbol	Name in code	Parameter	Default value	Minimum value	Maximum value
	DSED	thickness of active sediment layer (m)	0.1	-	-
k_{nitr}	TSC	nitrification rate, depends on d_{sed2} (d^{-1})	0.08	-	-
θ_n	TETANFIC	temperature coefficient for nitrification	1.1	-	-
T_{nref}	TNFIC	reference temperature for nitrification (°C)	21	-	-
	DSED2	sediment depth for comparing available oxygen and ammonium concentrations (m)	0.1	-	-
K_{sm}	MINMAX	mineralisation rate in sediment (d^{-1})	0.002	0.00001	0.03
	TETASMIN	temperature coefficient for mineralisation in sediment	1.07	-	-
	TSMIN	reference temperature for mineralisation in sediment (°C)	20	-	-
K_s	SEQCON	sequestering rate (d^{-1})	0.0	0	0.25
K_{diff}	KDIFF	diffusion coefficient for sediment (m^2 s^{-1})	$1*10^{-9}$	$2.5*10^{-12}$	$2*10^{-8}$
	HAR	aerobic sediment layer (m)	0.03	-	-
V_P	SINKP	sinking velocity of phytoplankton (m d^{-1})	0.8	0.0005	7.5
V_{ANFIX}	SINKAN	sinking velocity of blue-green algae (m d^{-1})	0.0	-	-
V_D	SINKD	sinking velocity of detritus (m d^{-1})	1.2	0.0005	10
	ADSCAPAN	phosphate sorption capacity in anaerobic sediment layer (mg P L^{-1})	51.953	-	-
	ADSCAP	phosphate sorption capacity in aerobic sediment layer (mg P L^{-1})	129.883	-	-
	OXYLIMIT	critical oxygen concentration (mg L^{-1})	1	-	-
	KAD	half saturation concentration for sorption (mg P L^{-1})	0.652	-	-
	DANS	first guess of amount of BIP (g P m^{-2})	4	-	-
	RESP	resuspension coefficient	1	-	-

3 BIOLA-PROBE data requirement

Except for the parameter values all data that will be used during simulation have to be supplied through the subroutine CASE. The data can be in any format that FORTRAN can read. This makes the model flexible. Often ASCII files are used, but sometimes input data are put as constants directly in the code of CASE. The results of the simulation are written on files. In CASE you specify which variables, process flows etc. that you want to extract from the simulation. Thus all input and output data communication is done through CASE.

For set up of the physical part of the model the following information is necessary:

- The **hypsograph** – lake area as a function of z. z is the vertical coordinate. It is directed upward and zero at the deepest bottom. The hypsograph can be implemented in CASE, but it is also possible to let PROBE generate the hypsograph given the surface area, maximum depth and an expansion parameter for the non-linearity of the horizontal area distribution.
- **Upper boundary conditions** for momentum and heat. Often meteorological forcing is used for the upper boundary conditions, but other solutions can be used (see PROBE Manual). In so far published applications of BIOLA (Pers and Persson, 2003), 3h-interval time series of a nearby meteorological observation station has been used, together with some routines for calculation of the upper boundary conditions of momentum and heat. In the process short-wave radiation has been calculated that is also used in BIOLA. Necessary meteorological variables for these routines are wind speed and direction, air temperature, relative humidity and cloud coverage. The last two are used to calculate sensible and latent heat flux and long- and short-wave radiation heat flux.
- The (heat) **extinction coefficient** (BETA). The coefficient is often estimated from mean Secchi depth of the lake (BETA = 1.6 / Secchi depth). This is not the same extinction coefficient that is used for in BIOLA. In BIOLA the short-wave radiation extinction is varying with time and depends on the amount of particles in the water.
- **Inflow and outflow** of water are necessary if you want to include water level variations in your simulation. Daily time series are often used. It is also possible to include precipitation and evaporation for water volume changes.

For set up of the physical model the following information is optional:

- Time series of **temperature in inflow** is necessary if the inflow enters the lake at different levels depending on density (temperature). If it is satisfactory to assume that the inflow blend into the same layer all the time this information is not necessary. In this case the layer next to the uppermost layer is often chosen when a variable surface level is used to avoid numerical difficulties with the uppermost layer.

For set up of the biogeochemical model the following information is necessary:

- **Short-wave radiation** is necessary for simulating growth of algae.

No specific biogeochemical data are necessary for BIOLA but some supply of nutrients needs to be applied. This data can have different forms:

- **Concentration of inflow** of primarily inorganic nitrogen and phosphorus, but also organic carbon and other variables could be used. The concentration could be time series or constant values. An alternative is, if inflow is not used, to have the incoming amount of nitrogen or phosphorus (or other) and add it to the lake surface as a boundary condition.

- **Atmospheric deposition** of inorganic nitrogen and phosphorus. Only a mean value for all days has been implemented. Atmospheric deposition is treated as an upper boundary condition. If time series are wanted this can be included in CASE.
- **Other sources** of inorganic nitrogen and phosphorus and organic carbon, e.g. from industries or urban wastewater. Time series are possible, but in earlier set ups a mean value has been used. Other sources are treated as an upper boundary condition and added to the surface layer.
- Mean depth (or rather distance from maximum depth) of the **border between erosion-transport and accumulation bottoms**. The model can calculate this depth if not known.
- **Starting values** of the state variables. These could be calibrated if not given.
- Local **observed parameter values**, e.g. detritus degradation rate, can be used instead of default values or calibrated values.

Without some data from within the lake or from the outflow, the model can not be calibrated and there is no way to tell how close the model is to simulating the real lake (validation). Thus some of the following information is necessary:

- Observed **temperature profiles** in the lake. (validation)
- Time series of **observed variables in the lake**, profiles or mean values. Observations can be of DIN, phosphate, detritus (or organic carbon), phytoplankton, or any other modelled state variable. DIN, phosphate and phytoplankton concentrations are probably necessary to get a useful model, the other are optional. (the observations are divided into a calibration period and a validation period)
- **Other observations/estimates** that can be compared with the model, i.e. budgets of total nitrogen and phosphorus, observed or estimated internal lake processes, etc. (validation)

4 Description of the code

4.1 Code structure of PROBE

Some knowledge of how the code of PROBE is build facilitates the comprehension of the BIOLA Manual, therefore a brief overview of the PROBE code is given here. The PROBE code is built of 14 subroutines; `MAIN`, `PROBE97`, `CASE`, `STORE`, `GRID`, `AREAD`, `SURF`, `PEA`, `DFAULT`, `PHYS`, `COMP`, `BOUND`, `OUTPUT` and `PLOTLP`. Only the first three will be discussed here. The `MAIN` program writes some information of the run on screen, and calls the `PROBE97` subroutine. `PROBE97` is the version of PROBE that has been used together with BIOLA. The `PROBE97` subroutine is divided into chapters that handle different parts of the simulation, e.g. initial data, boundary conditions, computation (Table 4). The `PROBE97` calls several subroutines for different tasks, but it is only one subroutine that is important for the user to know, the `CASE` subroutine. The `CASE` subroutine is also divided in chapters, and different chapters are called from certain chapters in `PROBE97` (see Table 4). Table 4 is a simplification of the connection between `PROBE97` and `CASE`, e.g. Chapter 3 in `CASE` (short `CASE(3)`) is actually called from a subroutine called in `PROBE97` Chapter 6.

Table 4. PROBE97 chapters and in which of these CASE chapters are called.

PROBE97 Chapters	CASE Chapters
1. Initial data --	1. Initial data
2. Grid and geometry	
3. Starting values	
4. Step control	
5. Boundary conditions ------------------------------------	2. Boundary conditions
6. Advance (solve the equations) ------------------------	3. Sources and sinks
7. Complete (dependent variables are updated)	
8. Print ---	4. Output
9. Decide (if another step, return to Ch. 4)	

PROBE saves all state variables (and some more variables) in an array of rank two called `F`. The first index denote cell and the second denote variable, e.g. `F(I,JTE)` is the temperature (variable `JTE`) in cell I. Cells are numbered from bottom up. The deepest cell is I=2 and the uppermost cell is I=N-1. I=N and I=1 denote the upper and lower boundary. The letters I and J are generally used for cell and variable respectively. The PROBE array `FLUXHZ(J)` is the upper boundary of the variable J. The arrays `QINFL(I)` and `QOUTFL(I)` are the inflow and outflow of water, while `PHIIN(I,J)` is the concentration of variable J of the inflow to cell I (`QINFL(I)`). These are central PROBE variables. All PROBE variables and arrays that are mentioned in this Manual are compiled in a table in Section 4.3.

4.2 Code structure of BIOLA

`BIOLA` is a subroutine called in Chapter 3 of `CASE` (i.e. `CASE(3)`). Chapter 3 handles sources and sinks and all the biogeochemical processes are modelled as sources and sinks of the general equations (1-2). Arguments of `BIOLA` is light (short-wave radiation), a variable telling whether macrophytes will be simulated or not, the z-coordinate for the border between erosion-transport bottoms and accumulation bottoms, the mean surface area (in km^2), and several variables for the accumulation of process flows. Chapter 3 of CASE is called from Chapter 6 (Advance) in PROBE97. The Advance chapter has a loop over all variables (hydrodynamic and biogeochemical). BIOLA is called in CASE(3) for every BIOLA state variable.

The BIOLA subroutine is also divided in chapters (Table 5, the whole code is found in Appendix C). The first chapter (A) makes the preliminary calculations, i.e. all process flows are calculated and checked against the pools (Table 5, right). The preliminaries are only calculated the first time BIOLA is called each time step. Then depending on the present state variable the BIOLA subroutine jumps to that variable's chapter and calculates the net source and sink of the variable (Table 5, left). There is one subroutine for each variable where its sources and sinks are added to the transport equation.

Table 5. The structure of the BIOLA *subroutine (name of called subroutine in capitals in parenthesis).*

BIOLA subroutine Chapters	Chapter A of BIOLA subroutine
A. Preliminaries, calculate all process flows and test them (see right)	Phytoplankton growth (GROWTH)
	Zooplankton grazing (GRAZER)
B. Zooplankton net source (ZOOPL)	Predation on zooplankton (PREDATZ)
C. Phytoplankton net source (PHYPL)	Predation by piscivorous fish (PREDATF)
D. Detritus net source (DETR)	Mortality of piscivorous fish (FISHMORT)
E. Sediment organic matter net source (SEDORG)	Cyanobacteria growth (CBGROWTH)
F. DIN net source (DISSNITR)	Denitrification (DENITR)
G. Cyanobacteria net source (CBACT)	Degradation in sediments (SEDMINE)
H. Macrophyte net source (MACROL)	Nitrification in sediments (NITFIC)
I. Phosphate net source (PHOS)	Adsorbed phosphate in sediment (ADSCOEFF)
J. Oxygen net source (OXYGEN)	Exchange of nitrogen and phosphorus between sediment and water (SEDDIFF)
K. Sediment phosphate net source (BPHOS)	Macrophyte growth (MACROF)
L. Sediment ammonium net source (BAMMO)	Macrophyte mortality (MACMORT)
M. Sediment nitrate net source (BNITR)	Detritus degradation (DDCOMP)
N. Planktivorous fish net source (FISHA)	Sequestering of sediment (SEQUEST)
O. Piscivorous fish net source (PISCI)	Sinking of phytoplankton (SINKPHY)
	Sinking of detritus (SINKDET)
	Sinking of cyanobacteria (SINKCB)
	Zooplankton mortality (ZOOMORT)
	Cyanobacteria mortality (CBMORT)
	Resuspension (ETADVECTION)
	Test of flow sizes (CHKMIN)

The code of the processes' subroutines can be found in Appendix C. Table 6 sorts the parameters (defined in Table 3) into the process subroutines defined in Table 5.

Table 6. Parameters that affect the biogeochemical processes. Parameter definitions are found in Table 3 and subroutines in Table 5.

Subroutine	Parameters
GROWTH	MYMAX, TPHY, TETAPHY, IOPTA, KDIN, KPO4, KW, KPKONST, KDKONST, KMKONST, IZKONST
GRAZER	GAMMA, P1, P2, P3, KG, EFFI
PREDATZ	PREDZ, TETAG, TREFG, KZO, (OMEGA)
PREDATF	PREDF, TETAG, TREFG, KFIA, (OMEGA)
FISHMORT	FMRT
CBGROWTH	MYMAXA, MYMAXB, KNFIX, SNF4, TPHY, TETAPHY, IOPTA, KDIN, KPO4, KW, KPKONST, KDKONST, KMKONST, IZKONST
DENITR	ADEN, BDEN, TDEN, TETADEN, KBNO, DSED
SEDMINE	MINMAX, TSMIN, TETASMIN
NITFIC	TSC, DSED2, TNFIC, TETANFIC
ADSCOEFF	ADSCAPAN, OXYLIMIT, ADSCAP, HAR, KAD, DANS
SEDDIFF	KDIFF, HAR
MACROF	MYMAXM, TMAC, TETAMAC, IOPTM, KMIN, KMIP, DSED2, KW, KPKONST, KDKONST, KMKONST, IZKONST
MACMORT	MMRT
DDCOMP	DDCC, TETAG, TREFG
SEQUEST	SEQCON
SINKPHY	SINKP
SINKDET	SINKD
SINKCB	SINKAN
ZOOMORT	ZMRT
CBMORT	XMRT
ETADVECTION	RESP

Other subroutines used with BIOLA, but called from the subroutine CASE and not from BIOLA, are ETADEPTH and OXYFLX. ETADEPTH calculates the distance from maximum depth of the border between erosion-transport and accumulation bottoms. OXYFLX calculates the oxygen upper boundary condition. These subroutines are discussed in Section 5.6 and 5.7.

4.3 Example of a CASE for a lake

This section will guide you through a simple CASE of PROBE for a lake. The model will calculate surface temperature and temperature profiles of the lake. All PROBE variables and parameters in the example are compiled in Table 7 and Table 8. The variables specific for this example are explained in the running text. I refer to the Tables during reading the example and the rest of the text for description of the different PROBE variables. For more possibilities of the use of PROBE check out the PROBE Manual and suitable CASE reports referenced therein.

Table 7. PROBE97 variable names mention in the text.

Name	Type	Description
AREA(NIM)	Real array	Horizontal area
AREAHZ	Real	Horizontal area of top cell
BETA	Real	Extinction coefficient
CORI	Real	Coriolis' parameter
CPHEAT	Real	Specific heat capacity
DZCELL(NIM)	Real array	Thickness of grid cells
F(NIM,NJM+2)	Real array	State variables, eddy viscosity and temperature
FLUXHZ(NJM)	Real array	Flux at upper boundary
FLXRAD	Real	Short-wave radiation
ICHAP	Integer	Chapter of CASE
IGRID	Integer	Controls the cell grid
IKBHZ(NJM)	Integer array	Index for type of upper boundary
IKBLZ(NJM)	Integer array	Index for type of lower boundary
INDARE	Integer	Index for area distribution
INIOUT	Logical	Controls initial output
IPROBE	Integer	Index of current PROBE
IPRSC	Integer	Index for Prandtl/Schmidt number
ITPLOT	Integer	Controls cross-stream plot
ITURBM	Integer	Index for turbulence model
LSTEP	Integer	Maximum number of time steps
MOVE	Logical	Controls moving surface
N	Integer	Number of grid points
NM1	Integer	Uppermost cell, N-1
NPROBE	Integer	Number of PROBEs
NPROF	Integer	Number of time steps between profile outputs
NSTAT	Integer	Number of time steps between boundary fluxes outputs
PHIIN(NIM,NJM)	Real array	Properties of inflow
PREEVA	Real	Precipitation and evaporation
QINFL(NIM)	Real array	Inflow
QOUTFL(NIM)	Real array	Outflow
RHOREF	Real	Reference density
SOLVAR(NJM)	Logical array	Select variables to be solved
STANTN(NJM)	Real array	Stanton number
TFRAC(20)	Real array	Specification of time step
VST1(NJM)	Real array	Values for starting profile
VST2(NJM)	Real array	Values for starting profile
Z(NIM)	Real array	Middle z-coordinate of cells
ZDIM	Real	Dimension of z-direction
ZST1(NJM)	Real array	z-coordinate for starting profiles
ZST2(NJM)	Real array	z-coordinate for starting profiles
ZSSTRT	Real	Initial maximum depth with moving surface

Table 8. PROBE97 parameter names mention in the text.

Name	Type	Description
JD	Integer	Variable number of dissipation of turbulent kinetic energy
JEMU	Integer	Variable number of turbulent eddy viscosity
JH	Integer	Variable number of heat energy
JK	Integer	Variable number of turbulent kinetic energy
JRHOU	Integer	Variable number of momentum in x-direction ($\rho * u$)
JRHOV	Integer	Variable number of momentum in y-direction ($\rho * v$)
JTE	Integer	Variable number of temperature
NIM	Integer	Maximum number of grid points
NJM	Integer	Maximum number of variables

The `CASE` subroutine accept one argument, `ICHAP`, that tells which Chapter, 1 to 4, of the `CASE` routine the program will step through this time. By IF-statements the right chapter is found. Since the subroutine is called CASE, the FORTRAN construct CASE is redefined in PROBE.

```
      SUBROUTINE CASE(ICHAP)
      IF(ICHAP.GT.1) GOTO 120
C     CHAPTER 1  MODIFY DEFAULT DATA
      RETURN
120   CONTINUE
C     CHAPTER 2  TRANSIENT BOUNDARY CONDITIONS
      IF(ICHAP.GT.2)GOTO 130
      RETURN
130   CONTINUE
C     CHAPTER 3  ADDITIONAL SOURCE TERMS
      IF(ICHAP.GT.3)GOTO 140
      RETURN
140   CONTINUE
C     CHAPTER 4  ADDITIONAL OUTPUT
      RETURN
      END
```

Before Chapter 1, there are declarations of local variables and the main variables of PROBE. The PROBE variables are declared as COMMON variables in the file comp97.inc. In this file the size of the PROBE variables is also defined, e.g. maximum number of cells. The comp97.inc file is included and local variables may be declared here. All local variables are saved so that they will retain their value to the next call for `CASE`.

```
      SUBROUTINE CASE(ICHAP)
      INCLUDE 'COMP97.INC'
      INTEGER ITIMESTEP,IYMD,TSDAY
      SAVE    ITIMESTEP,IYMD,TSDAY
      REAL    UG,VG,FH,FSB,QROUT
      SAVE    UG,VG,FH,FSB,QROUT
      IF(ICHAP.GT.1) GOTO 120
C     CHAPTER 1  MODIFY DEFAULT DATA
      ..
```

In Chapter 1 default values are redefined. The chapter is divided into 'Groups'. In the first group the time step (`TFRAC(2)`), and the number of grid points in time (`LSTEP`) and space (`N`) are given. It is possible to use variable time step (see PROBE Manual). This example gives a lake with 24 cells of constant thickness, and a constant time step of 600 seconds for

1000 steps. The number of time steps per day is a local variable that will be used later (TSDAY).

```
C       CHAPTER 1  MODIFY DEFAULT DATA
C       GROUP 1
        N=26
        LSTEP=1000
        TFRAC(2)=600.
        TSDAY=INT(86400./TFRAC(2))
        ..
```

In the second group the physical dimensions of the lake are given, the maximum depth (ZDIM) and surface area of the lake (AREAHZ). From these variables PROBE can calculate a cell grid and a hypsograph. The default is a uniform area. It is also possible to specify a cell grid and/or a hypsograph on you own. See Section 6.9 for instructions.

```
C       GROUP 2
        ZDIM=18.
        AREAHZ=5.5E6
        ..
```

The third group defines which variables will be calculated. The J corresponding to the variables are defined as JRHOU - x-direction momentum, JRHOV - y-direction momentum, JH - heat energy, JK - turbulent kinetic energy, JD - dissipation of turbulent kinetic energy.

```
C       GROUP 3
        SOLVAR(JRHOU)=.TRUE.
        SOLVAR(JRHOV)=.TRUE.
        SOLVAR(JH)=.TRUE.
        SOLVAR(JK)=.TRUE.
        SOLVAR(JD)=.TRUE.
        ..
```

Group four defines some properties. In this example density at reference temperature (3.98°C) is redefined (default is 1000).

```
C       GROUP 4
        RHOREF=999.975
        ..
```

Group 5 concerns the equation of state. In Group 6 choice between turbulence models can be made. Default is the k-ε model. Group 7 is for source terms. In this example the default value of the Coriolis' parameter (CORI) and the extinction coefficient (BETA) is changed. In Group 7 the inflow/outflow and solar radiation can be initialised, but since they in this example will be read from file the files will be opened.

```
C       GROUP 7
        CORI=1.2E-4
        BETA=1.6
        OPEN(UNIT=8,FILE='FLOW.DAT',FORM='FORMATTED',STATUS='OLD')
        OPEN(UNIT=10,FILE='METEO.DAT',FORM='FORMATTED',STATUS='OLD')
        ..
```

Group 8 is for initial data. Initial values can be given directly in the F-array (see PROBE Manual), but there is a simpler way. With arrays VST1 and VST2, the variables' values in the deepest cell and in the uppermost cell are given. The VST2 value is valid for all cells above ZST2, a z-coordinate. The default of ZST2 is zero, so only specifying VST2 gives the same value in all cells. Arrays ZST1 and ZST2 give the z-coordinates to which VST1 and VST2 values belong. Between ZST1 and ZST2 a linear function is used to determine the initial value of the variable. Here the heat is initialised to correspond to a temperature of 4°C.

```
C       GROUP 8
        VST2(JH)=16.76E6
        VST2(JK)=1.E-5
        VST2(JD)=5.E-8
        ..
```

Group 9 is for boundary conditions. Default is a boundary condition as a prescribed stationary flux, but for momentum a prescribed value (zero) at bottom boundary is used. The value does not need to be defined here since the default is zero. It is possible to define transient boundary conditions (see PROBE Manual).

```
C       GROUP 9
        IKBLZ(JRHOU)=1
        IKBLZ(JRHOV)=1
        ..
```

In Group 10 some limits may be changed. Normally these are not changed, but the minimum value of turbulent kinetic energy and dissipation of turbulent kinetic energy may have to be increased to obtain a steady solution. Group 11 is printout control. Here your own output files may be opened. PROBE has some automatic outputs on the screen. If you do not want these, set INIOUT, ITPLOT, NSTAT and NPROF like below.

```
C       GROUP 11
        INIOUT=.FALSE.
        NSTAT=9000000
        NPROF=NSTAT
        ITPLOT=1
        OPEN(UNIT=15,FILE='RESULTF.DAT',STATUS='UNKNOWN')
        OPEN(UNIT=16,FILE='PHYSPRO.DAT',STATUS='UNKNOWN')
        RETURN
        ..
```

In Group 12 the number of subbasins can be defined (see Section 6.11). In Group 13 a free surface can be allowed, when inflow is not equal to outflow (see Section 6.10). This ends Chapter 1 of CASE.

Chapter 2 sets the transient boundary conditions. It is called every time step. Some of the boundary conditions (FLUXHZ) in this example are read from file (FH, FSB), while others are calculated here from meteorological data (UG and VG are wind components). Short-wave solar radiation will penetrate into the water and be a source of heat. It is not a boundary flux and has its own variable FLXRAD (directed upward). In- and outflows are also set in Chapter 2. Here flow data are assumed to be daily measured discharge ($m^3\ s^{-1}$), and inflow (QINFL) is assumed to be the same as outflow (QOUTFL). The file is read once a day, and the flow assumed constant until next day. The inflow arrives in the uppermost cell (NM1) and has the same temperature/heat (PHIIN(NM1,JH)) as the lake has in that cell.

```
C       CHAPTER 2  TRANSIENT BOUNDARY CONDITIONS
        IF(ICHAP.GT.2)GOTO 130
        READ(10,*) ITIMESTEP,UG,VG,FH,FSB
        FLUXHZ(JRHOU)=-1.3E-3*UG*SQRT(UG**2+VG**2)
        FLUXHZ(JRHOV)=-1.3E-3*VG*SQRT(UG**2+VG**2)
        FLUXHZ(JH)=FH
        FLXRAD=FSB
        IF(MOD(ISTEP,TSDAY).EQ.0)THEN
           READ(8,*) IYMD,QROUT
           QINFL(NM1)=QROUT
```

```
    QOUTFL(NM1)=QROUT
    PHIIN(NM1,JH)=F(NM1,JH)
ENDIF
RETURN
..
```

The meteorological file meteo.dat may look like this.

```
0         7.424     -1.843     -6.334     0.000
1         7.424     -1.843     -6.334     0.000
2         7.424     -1.843     -6.334     0.000
..
```

The water flow file flow.dat may look like this.

```
19900101     1.530
19900102     1.420
19900103     1.220
..
```

Additional source terms can be added to Chapter 3. In Chapter 4 additional output is generated. In this example two files with surface temperature each time step and daily temperature profiles are created.

```
C       CHAPTER 4   ADDITIONAL OUTPUT
        IF(ISTEP.GT.0)THEN
            WRITE(15,*)ISTEP,Z(N),F(NM1,JTE)
            IF(MOD(ISTEP,TSDAY).EQ.0)THEN
                DO I=NM1,2,-1
                    WRITE(16,*)IYMD,Z(I),F(I,JTE)
                ENDDO
            ENDIF
            RETURN
        ENDIF
        END
```

The surface temperature file, resultf.dat, will show time step, z-coordinate of water surface and surface temperature.

```
     1    18.00000       4.001084
     2    18.00000       4.001483
     3    18.00000       4.001781
       ..
```

The temperature profile file, physpro.dat, will show date, z-coordinate and temperature.

```
19900101    17.62500       4.034519
19900101    16.87500       4.037349
19900101    16.12500       4.038846
..
```

That concludes this short introduction to the use of CASE. This introduction is for the readability of this manual only. If you are going to run PROBE (and/or BIOLA) it is strongly recommended that you study the PROBE Manual on how to set up PROBE. The whole example CASE subroutine is compiled below.

```
SUBROUTINE CASE(ICHAP)
INCLUDE 'COMP97.INC'
INTEGER ITIMESTEP,IYMD,TSDAY
SAVE    ITIMESTEP,IYMD,TSDAY
REAL    UG,VG,FH,FSB,QROUT
```

```
      SAVE    UG,VG,FH,FSB,QROUT
      IF(ICHAP.GT.1) GOTO 120
C     CHAPTER 1  MODIFY DEFAULT DATA
C     GROUP 1
      N=26
      LSTEP=1000
      TFRAC(2)=600.
      TSDAY=INT(86400./TFRAC(2))
C     GROUP 2
      ZDIM=18.
      AREAHZ=5.5E6
C     GROUP 3
      SOLVAR(JRHOU)=.TRUE.
      SOLVAR(JRHOV)=.TRUE.
      SOLVAR(JH)   =.TRUE.
      SOLVAR(JK)   =.TRUE.
      SOLVAR(JD)   =.TRUE.
C     GROUP 4
      RHOREF=999.975
C     GROUP 7
      CORI=1.2E-4
      BETA=1.6
      OPEN(UNIT=8,FILE='FLOW.DAT',FORM='FORMATTED',STATUS='OLD')
      OPEN(UNIT=10,FILE='METEO.DAT',FORM='FORMATTED',STATUS='OLD')
C     GROUP 8
      VST2(JH)=16.76E6
      VST2(JK)=1.E-5
      VST2(JD)=5.E-8
C     GROUP 9
      IKBLZ(JRHOU)=1
      IKBLZ(JRHOV)=1
C     GROUP 11
      INIOUT=.FALSE.
      NSTAT=9000000
      NPROF=NSTAT
      ITPLOT=1
      OPEN(UNIT=15,FILE='RESULTF.DAT',STATUS='UNKNOWN')
      OPEN(UNIT=16,FILE='PHYSPRO.DAT',STATUS='UNKNOWN')
      RETURN
120   CONTINUE
C     CHAPTER 2  TRANSIENT BOUNDARY CONDITIONS
      IF(ICHAP.GT.2)GOTO 130
      READ(10,*) ITIMESTEP,UG,VG,FH,FSB
      FLUXHZ(JRHOU)=-1.3E-3*UG*SQRT(UG**2+VG**2)
      FLUXHZ(JRHOV)=-1.3E-3*VG*SQRT(UG**2+VG**2)
      FLUXHZ(JH)=FH
      FLXRAD=FSB
      IF(MOD(ISTEP,TSDAY).EQ.0)THEN
         READ(8,*) IYMD,QROUT
         QINFL(NM1)=QROUT
         QOUTFL(NM1)=QROUT
         PHIIN(NM1,JH)=F(NM1,JH)
      ENDIF
      RETURN
130   CONTINUE
C     CHAPTER 3  ADDITIONAL SOURCE TERMS
      IF(ICHAP.GT.3)GOTO 140
      RETURN
140   CONTINUE
C     CHAPTER 4  ADDITIONAL OUTPUT
      IF(ISTEP.GT.0)THEN
         WRITE(15,*)ISTEP,Z(N),F(NM1,JTE)
         IF(MOD(ISTEP,TSDAY).EQ.0)THEN
```

```
        DO I=NM1,2,-1
           WRITE(16,*)IYMD,Z(I),F(I,JTE)
        ENDDO
     ENDIF
     RETURN
  ENDIF
  END
```

4.4 *Example of set up of BIOLA*

This section will describe how the simple CASE presented for PROBE in Section 4.3 can be expanded to simulate biogeochemistry in the lake with BIOLA. The whole CASE can be found in Appendix B.

Firstly a couple of variables need to be declared. Some of those are in the file biopar.inc, which is included in `CASE`. Others must be declared in `CASE`. These are the arguments of the `BIOLA` subroutine. They are already declared in the subroutine `BIOLA`, and can be copied from there. All these variables are saved between calls for `BIOLA`.

```
      INCLUDE 'BIOPAR.INC'
      LOGICAL MACOFF
      SAVE    MACOFF
      REAL ETD,SAREAKM2
      SAVE ETD,SAREAKM2
      REAL TOTGRZ,TOTPRED,TOTASINKI,TOTASINKO,
     1 TOTGRW,TOTAGRZ,TOTDSINKI,TOTDSINKO,TOTDECOM,
     1 TOTDFEK,TOTDFI,TOTDDEN,TOTDPRED,
     1 TOTSSINKP,TOTSSINKD,TOTSSINKN,TOTSMACM,
     1 TOTSDENI,TOTDIFFN,TOTWDENI,TOTDDC4,TOTGRO,
     1 TOTGRA,TOTNSINKI,TOTNSINKO,TOTNGRA,TOTNGRAX,
     1 TOTXGRZ,TOTMGROM,TOTMACM,TOTDIFFP,TOTPDENI,
     1 TOTDDC5,TOTPGRX,TOTPGROM,TOTOGRO,TOTOGRA,
     1 TOTOGRX,TOTDDC6,TOTONIT,TOTBPMIN,TOTBPDEN,
     1 TOTBPMAC,TOTBNMIN,TOTBNDEN,TOTNNIT,TOTBNMAC,
     1 TOTBNDIF,TOTINIT,TOTBIDEN,TOTBIDIF,TOTSMIN,
     1 TOTGROM,TOTFAGR,TOTFAM,TOTFBGR,TOTFBM,
     1 TOTZMRT,TOTDZM,
     1 TOTPGRO,TOTOSMIN,TOTOGROM,TOTBPDIF,TOTPGRA,
     1 TOTDMRTX,TOTXMRT,TOTDGRZ,TOTSMFI,TOTDETP,
     1 TOTETP,TOTSEQ
      SAVE TOTGRZ,TOTPRED,TOTASINKI,TOTASINKO,
     1 TOTGRW,TOTAGRZ,TOTDSINKI,TOTDSINKO,TOTDECOM,
     1 TOTDFEK,TOTDFI,TOTDDEN,TOTDPRED,
     1 TOTSSINKP,TOTSSINKD,TOTSSINKN,TOTSMACM,
     1 TOTSDENI,TOTDIFFN,TOTWDENI,TOTDDC4,TOTGRO,
     1 TOTGRA,TOTNSINKI,TOTNSINKO,TOTNGRA,TOTNGRAX,
     1 TOTXGRZ,TOTMGROM,TOTMACM,TOTDIFFP,TOTPDENI,
     1 TOTDDC5,TOTPGRX,TOTPGROM,TOTOGRO,TOTOGRA,
     1 TOTOGRX,TOTDDC6,TOTONIT,TOTBPMIN,TOTBPDEN,
     1 TOTBPMAC,TOTBNMIN,TOTBNDEN,TOTNNIT,TOTBNMAC,
     1 TOTBNDIF,TOTINIT,TOTBIDEN,TOTBIDIF,TOTSMIN,
     1 TOTGROM,TOTFAGR,TOTFAM,TOTFBGR,TOTFBM,
     1 TOTZMRT,TOTDZM,
     1 TOTPGRO,TOTOSMIN,TOTOGROM,TOTBPDIF,TOTPGRA,
     1 TOTDMRTX,TOTXMRT,TOTDGRZ,TOTSMFI,TOTDETP,
     1 TOTETP,TOTSEQ
```

Also some case specific variables in this example need declaration.

```
      REAL FOXY,DININ,PO4IN
      SAVE FOXY,DININ,PO4IN
```

The variable SAREAKM2 is assigned a value in Chapter 1, Group 2. It is the surface area of the lake in square kilometres. Simulating BIOLA demands an area distribution with depth other than the default uniform area. Setting the PROBE index INDARE equal to two gives a linear distribution (see PROBE Manual for other alternatives, and Section 6.9 for a user-specified area).

```
SAREAKM2=AREAHZ/1000000.
INDARE=2
```

In Chapter 1, Group 3 it is decided which BIOLA variables the equations will be solved for. In this example BIOLA will be modelled without fish variables. If fish variables will be used the last I value of the DO-loop, JBNO, should be exchanged for JFIB. All variable indices are found in Table 9.

```
DO I=JZOO,JBNO
   SOLVAR(I) =.TRUE.
ENDDO
```

Table 9. BIOLA state variable indices.

Name and value	Type	Description
JZOO=11	Integer	Variable index of zooplankton
JPHY=12	Integer	Variable index of phytoplankton
JDET=13	Integer	Variable index of detritus
JSED=14	Integer	Variable index of organic carbon in sediment
JDIN=15	Integer	Variable index of dissolved inorganic nitrogen
JANF=16	Integer	Variable index of cyanobacteria
JMAC=17	Integer	Variable index of macrophyte
JPO4=18	Integer	Variable index of phosphate
JOXY=19	Integer	Variable index of oxygen
JBIP=20	Integer	Variable index of phosphate in sediment
JBNH=21	Integer	Variable index of ammonium in sediment
JBNO=22	Integer	Variable index of nitrate in sediment
JFIA=23	Integer	Variable index of planktivorous fish
JFIB=24	Integer	Variable index of piscivorous fish

In Chapter 1, Group 7, the parameter file (biopar.par) is opened and read. This is done in a subroutine BIOPAR (see Section 5.4 for more details about this routine). The parameter file contains the values of all parameters for the BIOLA processes. The variable ETD (depth of erosion-transport bottoms) is set here, or it can be calculated with the subroutine ETADEPTH (see Section 5.7). Also a data file with daily nutrient concentrations of inflow is opened for later use.

```
OPEN(UNIT=7,FILE='BIOPAR.PAR',FORM='FORMATTED',STATUS='OLD')
CALL BIOPAR(7)
CLOSE(7)
ETD=10.
OPEN(UNIT=9,FILE='NUTIN.DAT',FORM='FORMATTED',STATUS='OLD')
```

In Chapter 1, Group 8, the starting profile is defined the same way as for PROBE variables. If macrophytes are not to be simulated, setting the variable MACOFF true turn them off.

```
VST2(JPHY)=0.1
VST2(JZOO)=VST2(JPHY)/700.
VST2(JDET)=0.1
VST2(JSED)=4.
VST2(JDIN)=1.1
VST2(JANF)=0.1
VST2(JPO4)=0.01
VST2(JOXY)=10.0
VST2(JBIP)=0.3
VST2(JBNH)=0.01
VST2(JBNO)=0.01
MACOFF=.TRUE.
```

In Group 11 two files are opened for saving some BIOLA results.

```
OPEN(UNIT=17,FILE='NUTPRO.DAT',STATUS='UNKNOWN')
OPEN(UNIT=18,FILE='FLOWZOO.DAT',STATUS='UNKNOWN')
```

In Chapter 2 the boundary fluxes are given. For oxygen the exchange with atmosphere is calculated by the BIOLA subroutine OXYFLX with the input argument wind speed. Atmospheric deposition of DIN and phosphate are parameters read from biopar.par, as is point source emissions of DIN and phosphate (Table 10). All these boundary fluxes are added to the FLUXHZ variable. For more details of these subroutines and variables see Sections 5.4, 5.6 and 6.4. Sources of nutrients or other substances through inflow are also determined here. In this case daily concentration of DIN and PO_4 of inflow are read from file once a day. The values are put in the PROBE array PHIIN.

```
  CALL OXYFLX(SQRT(UG**2+VG**2),FOXY)
  FLUXHZ(JOXY)=FOXY
  FLUXHZ(JDIN)=-NDEPPAR/1000./365./86400.
1                -POINTDIN/SAREAKM2/1000./365./86400.
  FLUXHZ(JPO4)=-PDEPPAR/1000./365./86400.
1                -POINTPO4/SAREAKM2/1000./365./86400.
  IF(MOD(ISTEP,TSDAY).EQ.0)THEN
      READ(9,*) IYMD,DININ,PO4IN
      PHIIN(NM1,JDIN)=DININ
      PHIIN(NM1,JPO4)=PO4IN
  ENDIF
```

Table 10. BIOLA parameters that are not process parameters.

Name	Type	Description
NDEPPAR	Real	Atmospheric deposition of DIN (mg N m^{-2} yr^{-1})
PDEPPAR	Real	Atmospheric deposition of phosphate (mg P m^{-2} yr^{-1})
POINTBIP	Real	Emissions of particulate phosphorus put in sediment from point sources (kg P yr^{-1})
POINTDIN	Real	Emissions of DIN from point sources (kg N yr^{-1})
POINTPO4	Real	Emissions of phosphate from point sources (kg P yr^{-1})
POINTDET	Real	Emissions of organic matter from point sources (kg C yr^{-1})

In Chapter 3 the main BIOLA routine is called. In this routine all the biogeochemical processes is calculated and added as sources and sinks to the transport equations. For details of this subroutine see Section 5.1.

```
IF(ISTEP.GT.0)THEN
    IF(J.GE.JZOO.AND.J.LE.JBNO)THEN
        CALL BIOLA(-FLXRAD,MACOFF,ETD,SAREAKM2,
```

```
1 TOTGRZ,TOTPRED,TOTASINKI,TOTASINKO,TOTGRW,
1 TOTAGRZ,TOTDSINKI,TOTDSINKO,TOTDECOM,TOTDGRZ,TOTDPRED,
1 TOTDFEK,TOTDFI,TOTSMFI,TOTDDEN,TOTSSINKP,TOTSSINKD,TOTSSINKN,
1 TOTSMACM,TOTSDENI,TOTSMIN,TOTDIFFN,TOTWDENI,TOTDDC4,TOTGRO,
1 TOTGRA,TOTGROM,TOTNSINKI,TOTNSINKO,TOTNGRA,
1 TOTNGRAX,TOTXGRZ,TOTMGROM,TOTMACM,TOTDIFFP,TOTPDENI,
1 TOTDDC5,TOTPGRO,TOTPGRA,TOTPGRX,TOTPGROM,TOTOGRO,TOTOGRA,
1 TOTOGRX,TOTOGROM,TOTOSMIN,TOTDDC6,TOTONIT,TOTBPMIN,
1 TOTBPDEN,TOTBPMAC,TOTBPDIF,TOTBNMIN,TOTBNDEN,TOTNNIT,
1 TOTBNMAC,TOTBNDIF,TOTINIT,TOTBIDEN,TOTBIDIF,TOTFAGR,
1 TOTFAM,TOTFBGR,TOTFBM,TOTZMRT,TOTDZM,TOTDMRTX,TOTXMRT,
1 TOTDETP,TOTETP,TOTSEQ)
      ENDIF
  ENDIF
```

Outputs from this BIOLA simulation will be daily profiles of nutrient concentrations while all the sources and sinks of the zooplankton variable, i.e. the process flows associated with zooplankton, will be saved for each time step. The results are saved on file in Chapter 4. After the process flows are written, they are zeroed to be ready for adding the flows until next printout.

```
IF(MOD(ISTEP,TSDAY).EQ.0) THEN
    DO I=NM1,2,-1
        WRITE(17,*)IYMD,Z(I),F(I,JDIN),F(I,JPO4)
    ENDDO
ENDIF
WRITE(18,*)ISTEP,TOTGRZ,TOTPRED,TOTZMRT
TOTGRZ=0.0
TOTPRED=0.0
TOTZMRT=0.0
```

5 Description of the BIOLA code

5.1 The BIOLA subroutine

The `BIOLA` subroutine is located in the FORTRAN-file BIOSO.FOR. The subroutine has a lot of arguments (75). The first four are input arguments, while the rest are output arguments. The input arguments are solar radiation, which is used for algal growth, a variable characterising the bottoms, a variable for turning off the macrophyte variable, and the lake surface area in square kilometres. The output arguments are the accumulated process flows of the simulation. The arguments are described in Table 11.

Table 11. Arguments of BIOLA subroutine; name, type, description and involved state variable for process flows.

Name	Type	State variable	Description
`ETD`	Real		z-coordinate for border between erosion-transport bottoms and accumulation bottoms (m)
`I0`	Real		Short-wave radiation (W m^{-2})
`MACOFF`	Logical		Control for excluding macrophytes
`SAREAKM2`	Real		Surface area (km^2)
`TOTGRZ`	Real	Z	Zooplankton grazing (g C d^{-1})
`TOTPRED`	Real	Z	Predation on zooplankton (g C d^{-1})
`TOTZMRT`	Real	Z	Zooplankton mortality (g C d^{-1})
`TOTASINKI`	Real	A	Phytoplankton from above layer (g C d^{-1})
`TOTASINKO`	Real	A	Phytoplankton sinking out of layer (g C d^{-1})
`TOTGRW`	Real	A	Phytoplankton growth (g C d^{-1})
`TOTAGRZ`	Real	A	Grazing on phytoplankton (g C d^{-1})
`TOTDSINKI`	Real	D	Detritus from above layer (g C d^{-1})
`TOTDSINKO`	Real	D	Detritus sinking out of layer (g C d^{-1})
`TOTDECOM`	Real	D	Mineralisation (g C d^{-1})
`TOTDFEK`	Real	D	Leftovers from zooplankton grazing (g C d^{-1})
`TOTDFI`	Real	D	Leftovers from fish predation on fish (g C d^{-1})
`TOTDDEN`	Real	D	Denitrification in water (g C d^{-1})
`TOTDPRED`	Real	D	Leftovers from fish predation on zooplankton (g C d^{-1})
`TOTDETP`	Real	D	Resuspension (g C d^{-1})
`TOTDZM`	Real	D	Zooplankton mortality (g C d^{-1})
`TOTDGRZ`	Real	D	Zooplankton grazing (g C d^{-1})
`TOTDMRTX`	Real	D	Cyanobacteria mortality (g C d^{-1})
`TOTSSINKP`	Real	S	Sedimentation of phytoplankton (g C d^{-1})
`TOTSSINKD`	Real	S	Sedimentation of detritus (g C d^{-1})
`TOTSSINKN`	Real	S	Sedimentation of cyanobacteria (g C d^{-1})
`TOTSMACM`	Real	S	Macrophyte mortality (g C d^{-1})
`TOTSDENI`	Real	S	Denitrification (g C d^{-1})
`TOTSMIN`	Real	S	Mineralisation (g C d^{-1})
`TOTSMFI`	Real	S	Fish mortality (g C d^{-1})
`TOTETP`	Real	S	Resuspension (g C d^{-1})
`TOTSEQ`	Real	S	Sequestering (g C d^{-1})

Table 11. Continued.

Name	Type	State variable	Description
TOTDIFFN	Real	DIN	Exchange with sediments (g N d^{-1})
TOTWDENI	Real	DIN	Denitrification in water (g N d^{-1})
TOTDDC4	Real	DIN	Mineralisation of detritus (g N d^{-1})
TOTGRO	Real	DIN	Uptake by phytoplankton (g N d^{-1})
TOTGRA	Real	DIN	Uptake by cyanobacteria (g N d^{-1})
TOTGROM	Real	DIN	Uptake by macrophyte (g N d^{-1})
TOTNSINKI	Real	CB	Cyanobacteria from above layer (g C d^{-1})
TOTNSINKO	Real	CB	Cyanobacteria sinking (g C d^{-1})
TOTNGRA	Real	CB	Cyanobacteria growth using DIN (g C d^{-1})
TOTNGRAX	Real	CB	Cyanobacteria growth using N_2 (g C d^{-1})
TOTXGRZ	Real	CB	Zooplankton grazing (g C d^{-1})
TOTXMRT	Real	CB	Cyanobacteria mortality (g C d^{-1})
TOTMGROM	Real	M	Macrophyte growth (g C d^{-1})
TOTMACM	Real	M	Macrophyte mortality (g C d^{-1})
TOTDIFFP	Real	PO4	Exchange with sediment (g P d^{-1})
TOTPDENI	Real	PO4	Denitrification in water (g P d^{-1})
TOTDDC5	Real	PO4	Mineralisation of detritus (g P d^{-1})
TOTPGRO	Real	PO4	Uptake by phytoplankton (g P d^{-1})
TOTPGRA	Real	PO4	Uptake by cyanobacteria during growth on DIN (g P d^{-1})
TOTPGRX	Real	PO4	Uptake by cyanobacteria during growth on N_2 (g P d^{-1})
TOTPGROM	Real	PO4	Uptake by macrophyte (g P d^{-1})
TOTOGRO	Real	O2	Phytoplankton growth (g O_2 d^{-1})
TOTOGRA	Real	O2	Cyanobacteria growth on DIN (g O_2 d^{-1})
TOTOGRX	Real	O2	Cyanobacteria growth on N_2 (g O_2 d^{-1})
TOTOGROM	Real	O2	Macrophyte growth (g O_2 d^{-1})
TOTOSMIN	Real	O2	Mineralisation in sediment (g O_2 d^{-1})
TOTDDC6	Real	O2	Mineralisation of detritus (g O_2 d^{-1})
TOTONIT	Real	O2	Nitrification (g O_2 d^{-1})
TOTBPMIN	Real	BIP	Mineralisation in sediment (g P d^{-1})
TOTBPDEN	Real	BIP	Denitrification (g P d^{-1})
TOTBPMAC	Real	BIP	Uptake by macrophytes (g P d^{-1})
TOTBPDIF	Real	BIP	Exchange with water (g P d^{-1})
TOTBNMIN	Real	BNH	Mineralisation in sediment (g N d^{-1})
TOTBNDEN	Real	BNH	Denitrification (g N d^{-1})
TOTNNIT	Real	BNH	Nitrification (g N d^{-1})
TOTBNMAC	Real	BNH	Uptake by macrophytes (g N d^{-1})
TOTBNDIF	Real	BNH	Exchange with water (g N d^{-1})
TOTINIT	Real	BNO	Nitrification (g N d^{-1})
TOTBIDEN	Real	BNO	Denitrification (g N d^{-1})
TOTBIDIF	Real	BNO	Exchange with water (g N d^{-1})
TOTFAGR	Real	FA	Growth of planktivorous fish (g C d^{-1})
TOTFAM	Real	FA	Predation on planktivorous fish (g C d^{-1})
TOTFBGR	Real	FB	Growth of piscivorous fish (g C d^{-1})
TOTFBM	Real	FB	Piscivorous fish mortality (g C d^{-1})

Some common variables declared in files are needed in BIOLA and the files are therefore included. Other variables are declared locally in the subroutine. Some of these variables (process rates and water flows) are saved between calls on BIOLA. BIOLA is called for every biogeochemical variable during a time step. The first biogeochemical variable is zooplankton (J=JZOO) (Table 9). Therefore, when zooplankton is the current variable, the preliminaries are calculated. The preliminaries include calculation of all the process rates and checking that they do not make the variable go below its minimum value. The subroutines that calculate the process rates are described in next section. The checking subroutine (CHKMIN) is described in section 5.5. After the preliminaries are calculated, depending on the value of J, the appropriate source and sink subroutine is called. This is accomplished with a set of IF statements and GOTO statements. The source and sink subroutines are described in detail in section 5.3.

The inflow, outflow, and advective flows are saved in local variables (SAQIN, SAQOUT, SAQZ) the first time BIOLA is called in the time step. Then these variables can be set to zero for the state variables that do not have advection, i.e. sediment variables, macrophytes and fish, but still be retrieved for later state variables using them. Diffusion is also set to zero for sediment variables, but is not saved locally because it is calculated anew for every J.

```
C********************************************************************
      SUBROUTINE BIOLA(I0,MACOFF,ETD,SAREA,TOTGRZ,TOTPRED,
     1 TOTASINKI,TOTASINKO,TOTGRW,TOTAGRZ,TOTDSINKI,TOTDSINKO,
     1 TOTDECOM,TOTDGRZ,TOTDPRED,TOTDFEK,TOTDFI,TOTSMFI,TOTDDEN,
     1 TOTSSINKP,TOTSSINKD,TOTSSINKN,TOTSMACM,TOTSDENI,TOTSMIN,
     1 TOTDIFFN,TOTWDENI,TOTDDC4,TOTGRO,TOTGRA,TOTGROM,
     1 TOTNSINKI,TOTNSINKO,TOTNGRA,TOTNGRAX,TOTXGRZ,
     1 TOTMGROM,TOTMACM,TOTDIFFP,TOTPDENI,TOTDDC5,TOTPGRO,TOTPGRA,
     1 TOTPGRX,TOTPGROM,TOTOGRO,TOTOGRA,TOTOGRX,TOTOGROM,TOTOSMIN,
     1 TOTDDC6,TOTONIT,TOTBPMIN,TOTBPDEN,TOTBPMAC,TOTBPDIF,
     1 TOTBNMIN,TOTBNDEN,TOTNNIT,TOTBNMAC,TOTBNDIF,TOTINIT,
     1 TOTBIDEN,TOTBIDIF,TOTFAGR,TOTFAM,TOTFBGR,TOTFBM,TOTZMRT,
     1 TOTDZM,TOTDMRTX,TOTXMRT,TOTDETP,TOTETP,TOTSEQ)
C********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'

      INTEGER I
      LOGICAL MACOFF

      REAL I0
      REAL ETD,SAREA
      REAL TOTGRZ,TOTPRED,TOTASINKI,TOTASINKO,
     1 TOTGRW,TOTAGRZ,TOTDSINKI,TOTDSINKO,TOTDECOM,TOTDGRZ,TOTDPRED,
     1 TOTDFEK,TOTDFI,TOTSMFI,TOTDDEN,
     1 TOTSSINKP,TOTSSINKD,TOTSSINKN,TOTSMACM,TOTSDENI,TOTSMIN,
     1 TOTDIFFN,TOTWDENI,TOTDDC4,TOTGRO,TOTGRA,TOTGROM,
     1 TOTNSINKI,TOTNSINKO,TOTNGRA,TOTNGRAX,TOTXGRZ,
     1 TOTMGROM,TOTMACM,TOTDIFFP,TOTPDENI,TOTDDC5,TOTPGRO,TOTPGRA,
     1 TOTPGRX,TOTPGROM,TOTOGRO,TOTOGRA,TOTOGRX,TOTOGROM,TOTOSMIN,
     1 TOTDDC6,TOTONIT,TOTBPMIN,TOTBPDEN,TOTBPMAC,TOTBPDIF,
     1 TOTBNMIN,TOTBNDEN,TOTNNIT,TOTBNMAC,TOTBNDIF,
     1 TOTINIT,TOTBIDEN,TOTBIDIF,TOTFAGR,TOTFAM,TOTFBGR,TOTFBM
     1 TOTZMRT,TOTDZM,TOTDMRTX,TOTXMRT,TOTDETP,TOTETP,TOTSEQ
      REAL SAQIN(NIM),SAQOUT(NIM),SAQZ(NIM)
      SAVE SAQIN,SAQOUT,SAQZ
      REAL GRO(NIM),ZGRAZE(NIM),AGRAZE(NIM),
     1     DETGRZ(NIM),CBXGRZ(NIM),DETFEK(NIM)
      SAVE GRO,ZGRAZE,AGRAZE,DETGRZ,CBXGRZ,DETFEK
      REAL PREZ(NIM),PREFI(NIM),FMORT(NIM),GRA(NIM),GRAX(NIM)
```

```
      SAVE PREZ,PREFI,FMORT,GRA,GRAX
      REAL SDEN(NIM),WDEN(NIM),SMIN(NIM)
      SAVE SDEN,WDEN,SMIN
      REAL NIT(NIM),DIFP(NIM),DIFNH(NIM),DIFNO(NIM)
      SAVE NIT,DIFP,DIFNH,DIFNO
      REAL GROM(NIM),DDC(NIM),MACM(NIM),SEQ(NIM)
      SAVE GROM,DDC,MACM,SEQ
      REAL GMBN(NIM),GMDN(NIM),GMBP(NIM),GMDP(NIM)
      SAVE GMBN,GMDN,GMBP,GMDP
      REAL SINKOP(NIM),SINKOD(NIM),SINKOX(NIM)
      REAL SINKIP(NIM),SINKID(NIM)
      SAVE SINKOP,SINKOD,SINKOX,SINKIP,SINKID
      REAL SINKIX(NIM),SEDP(NIM),SEDD(NIM),SEDX(NIM),ETPROP(NIM)
      SAVE SINKIX,SEDP,SEDD,SEDX,ETPROP
      REAL ZMORT(NIM),MORTX(NIM)
      SAVE ZMORT,MORTX

      IF(J.EQ.JZOO) GOTO 11
      IF(J.EQ.JPHY) GOTO 12
      IF(J.EQ.JDET) GOTO 13
      IF(J.EQ.JSED) GOTO 14
      IF(J.EQ.JDIN) GOTO 15
      IF(J.EQ.JANF) GOTO 16
      IF(J.EQ.JMAC) GOTO 17
      IF(J.EQ.JPO4) GOTO 18
      IF(J.EQ.JOXY) GOTO 19
      IF(J.EQ.JBIP) GOTO 20
      IF(J.EQ.JBNH) GOTO 21
      IF(J.EQ.JBNO) GOTO 22
      IF(J.EQ.JFIA) GOTO 23
      IF(J.EQ.JFIB) GOTO 24
      RETURN

C-----------------------------------------------------------------
CHAPTER A A A A A A PRELIMINARIES A A A A A A A A A
   11 CONTINUE

C     SAVE QINFL,QOUTFL,QZ TEMPORARY
      DO I=1,N
          SAQIN(I)=QINFL(I)
          SAQOUT(I)=QOUTFL(I)
          SAQZ(I)=QZ(I)
      ENDDO

C      CALCULATE NUTRIENT AND CARBON PROCESS RATES
       CALL GROWTH(I0,GRO,MACOFF)
       CALL GRAZER(ZGRAZE,AGRAZE,DETGRZ,CBXGRZ,DETFEK)
       CALL PREDATZ(PREZ)
       CALL PREDATF(PREFI)
       CALL FISHMORT(FMORT)
       CALL CBGROWTH(I0,GRA,GRAX,MACOFF)
       CALL DENITR(SDEN,WDEN)
       CALL SEDMINE(SMIN,SDEN)
       CALL NITFIC(NIT)
       CALL ADSCOEFF(FAKTA)
       CALL SEDDIFF(DIFP,DIFNO,DIFNH,FAKTA)
       CALL MACROF(I0,GROM,GMBN,GMDN,GMBP,GMDP,MACOFF)
       CALL MACMORT(MACM,MACOFF)
       CALL DDCOMP(DDC)
       CALL SEQUEST(SEQ)
       CALL SINKPHY(SINKOP,SINKIP,SEDP)
       CALL SINKDET(SINKOD,SINKID,SEDD)
       CALL SINKCB(SINKOX,SINKIX,SEDX)
```

```
      CALL ZOOMORT(ZMORT)
      CALL CBMORT(MORTX)
      CALL ETADVECTION(ETD,WG,ETPROP)
      CALL CHKMIN(GRO,ZGRAZE,AGRAZE,DETGRZ,CBXGRZ,DETFEK,PREZ,PREFI,
     1 FMORT,GRA,GRAX,SDEN,WDEN,SMIN,NIT,DIFP,DIFNO,DIFNH,GROM,GMBN,
     1  GMDN,GMBP,GMDP,MACM,DDC,SINKOD,SINKID,SEDD,SINKOP,SINKIP,
     1 SEDP,SEQ,SINKOX,SINKIX,SEDX,ZMORT,MORTX,ETPROP)
C-----------------------------------------------------------------------
CHAPTER  B  B  B  B  B  B  ZOOPLANKTON  B  B  B  B  B  B  B  B  B
      CALL ZOOPL(ZGRAZE,PREZ,ZMORT,TOTGRZ,TOTPRED,TOTZMRT)
      RETURN
C-----------------------------------------------------------------------
CHAPTER  C  C  C  C  C  C  PHYTOPLANKTON  C  C  C  C  C  C  C  C  C
   12 CONTINUE
      CALL PHYPL(GRO,AGRAZE,SINKOP,SINKIP,
     1 TOTASINKI,TOTGRW,TOTASINKO,TOTAGRZ)
      RETURN
C-----------------------------------------------------------------------
CHAPTER  D  D  D  D  D  D  DETRITUS  D  D  D  D  D  D  D  D  D  D  D
   13 CONTINUE
      CALL DETR(DETFEK,DETGRZ,PREZ,PREFI,ZMORT,WDEN,DDC,SINKOD,
     1 SINKID,MORTX,ETPROP,TOTDSINKI,TOTDPRED,TOTDFEK,TOTDFI,TOTDDEN,
     1 TOTDSINKO,TOTDECOM,TOTDGRZ,TOTDZM,TOTDMRTX,TOTDETP)
      RETURN
C-----------------------------------------------------------------------
CHAPTER  E  E  E  E  E  E  SEDIMENT ORGANIC MATTER  E  E  E  E  E  E
   14 CONTINUE
      CALL SEDORG(SDEN,SMIN,MACM,SEQ,SEDP,SEDD,SEDX,ETPROP,FMORT,
     1 TOTSSINKP,TOTSSINKD,TOTSSINKN,TOTSMACM,TOTSDENI,TOTSMIN,
     1 TOTSMFI,TOTETP,TOTSEQ)
C     EXCLUDE ADVECTION AND DIFFUSION
      DO I=1,N
          QINFL(I)=0.
          QOUTFL(I)=0.
          QZ(I)=0.
          DIF(I)=0.
      ENDDO
      RETURN
C-----------------------------------------------------------------------
CHAPTER  F  F  F  F  F  F  DISSOLVED INORGANIC NITROGEN  F  F  F  F
   15 CONTINUE
      CALL DISSNITR(WDEN,GRO,GRA,DIFNO,DIFNH,GMDN,DDC,
     1 TOTDIFFN,TOTWDENI,TOTDDC4,TOTGRO,TOTGRA,TOTGROM)
C     INCLUDE ADVECTION AGAIN
      DO I=1,N
          QINFL(I)=SAQIN(I)
          QOUTFL(I)=SAQOUT(I)
          QZ(I)=SAQZ(I)
      ENDDO
      RETURN
C-----------------------------------------------------------------------
CHAPTER  G  G  G  G  G  G  CYANOBACTERIA  G  G  G  G  G  G  G  G
   16 CONTINUE
      CALL CBACT(GRA,GRAX,CBXGRZ,SINKOX,SINKIX,MORTX,TOTNSINKI,
     1  TOTNSINKO,TOTNGRA,TOTNGRAX,TOTXGRZ,TOTXMRT)
      RETURN
C-----------------------------------------------------------------------
CHAPTER  H  H  H  H  H  H  MACROPHYTES  H  H  H  H  H  H  H  H  H  H
   17 CONTINUE
      CALL MACKOL(GROM,MACM,TOTMGROM,TOTMACM,MACOFF)
C     EXCLUDE ADVECTION AND DIFFUSION
      DO I=1,N
          QINFL(I)=0.
```

```
          QOUTFL(I)=0.
          QZ(I)=0.
          DIF(I)=0.
      ENDDO
      RETURN
C----------------------------------------------------------------------
CHAPTER  I  I  I  I  I  I  PHOSPHATE  I  I  I  I  I  I  I  I  I  I
   18 CONTINUE
      CALL PHOS(GRO,GRA,GRAX,WDEN,DIFP,GMDP,DDC,TOTDIFFP,
     1 TOTPDENI,TOTDDC5,TOTPGRO,TOTPGRA,TOTPGRX,TOTPGROM)
C     INCLUDE ADVECTION AGAIN
      DO I=1,N
          QINFL(I)=SAQIN(I)
          QOUTFL(I)=SAQOUT(I)
          QZ(I)=SAQZ(I)
      ENDDO
      RETURN
C----------------------------------------------------------------------
CHAPTER  J  J  J  J  J  J  OXYGEN  J  J  J  J  J  J  J  J  J  J  J
   19 CONTINUE
      CALL OXYGEN(GRO,GRA,GRAX,SMIN,NIT,GROM,DDC,
     1 TOTOGRO,TOTOGRA,TOTOGRX,TOTOGROM,TOTOSMIN,TOTDDC6,TOTONIT)
      RETURN
C----------------------------------------------------------------------
CHAPTER  K  K  K  K  K  K  BENTHIC PHOSPHORUS  K  K  K  K  K  K  K
   20 CONTINUE
      CALL BPHOS(SMIN,SDEN,DIFP,GMBP,TOTBPMIN,TOTBPDEN,
     1 TOTBPMAC,TOTBPDIF,SAREA)
C     EXCLUDE ADVECTION AND DIFFUSION
      DO I=1,N
          QINFL(I)=0.
          QOUTFL(I)=0.
          QZ(I)=0.
          DIF(I)=0.
       ENDDO
      RETURN
C----------------------------------------------------------------------
CHAPTER  L  L  L  L  L  L  BENTHIC AMMONIUM  L  L  L  L  L  L  L  L
   21 CONTINUE
      CALL BAMMO(SMIN,SDEN,NIT,DIFNH,GMBN,
     1 TOTBNMIN,TOTBNDEN,TOTNNIT,TOTBNMAC,TOTBNDIF)
C      EXCLUDE DIFFUSION, NEEDS TO BE DONE EVERY J
       DO I=1,N
          DIF(I)=0.
       ENDDO
      RETURN
C----------------------------------------------------------------------
CHAPTER  M  M  M  M  M  M  BENTHIC NITRATE  M  M  M  M  M  M  M  M
   22 CONTINUE
      CALL BNITR(SDEN,NIT,DIFNO,TOTINIT,TOTBIDEN,TOTBIDIF)
C      EXCLUDE DIFFUSION, NEEDS TO BE DONE EVERY J
       DO I=1,N
          DIF(I)=0.
       ENDDO
      RETURN
C----------------------------------------------------------------------
CHAPTER  N  N  N  N  N  N  PLANKTIVOROUS FISH  N  N  N  N  N  N  N
   23 CONTINUE
      CALL FISHA(PREZ,PREFI,TOTFAGR,TOTFAM)
      RETURN
C----------------------------------------------------------------------
CHAPTER  O  O  O  O  O  O  PISCIVOROUS FISH  O  O  O  O  O  O  O  O
   24 CONTINUE
```

```
      CALL PISCI(PREFI,FMORT,TOTFBGR,TOTFBM)
      RETURN
      END
```

5.2 The process rate subroutines

Each process is calculated in a subroutine. These are called from BIOLA the first time BIOLA is called each time step. The subroutines are gathered in the Fortran-file BIOPRO.FOR. The subroutines have the calculated process rates as output arguments. The parameters of the process equations are declared in the include-file biopar.inc. The process rates are gathered in Table 12 together with a description and a reference to the source and sink subroutine of the state variables that it is influencing. The process rate subroutines are described below in alphabetical order. In addition to the process rates, a subroutine calculating the adsorption of phosphate in the sediment are described (ADSCOEFF). Names of local variables in the subroutines are, when different from the names used in BIOSO.FOR, given in the describing text only. Most processes are compared with the minimum concentration of the variables it is a sink of directly in the process rate subroutine. More details about the process rates can be found in Pers (2002) and for some subroutines in Ahlkrona (2002).

The subroutine ADSCOEFF calculates the part of phosphate that is adsorbed in sediment. A fraction of the sediment phosphate is assumed adsorbed. The variable FAKTA is the quotient between adsorbed and dissolved sediment phosphate. The fraction depends on the model parameters ADSCAP, ADSCAPAN, ANACOEFF, OXYLIMIT, KAD, DANS and HAR (Table 3). More details about this subroutine can be found in Ahlkrona (2002).

```
C*******************************************************************
      SUBROUTINE ADSCOEFF(FAKTA)
C*******************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL FAKTA(NIM),PSC,D,A,K

      DO I=2,NM1
         IF (F(I,JOXY).LT.OXYLIMIT) THEN
            PSC=ADSCAPAN*(ANACOEFF**(F(I,JOXY)/OXYLIMIT))*HAR
         ELSE
            PSC=ADSCAP*HAR
         ENDIF
         K=KAD*HAR
         D=DANS*HAR
         DO
            A=PSC*D/(D+K)
            IF (ABS((A+D)-F(I,JBIP)).LT.(0.001*F(I,JBIP))) EXIT
            D=D*F(I,JBIP)/(D+A)
         ENDDO
         FAKTA(I)=A/D
      ENDDO
      RETURN
      END
```

Table 12. Process rates in BIOLA with corresponding state variable subroutine.

Process rate variable name	Description	State variable subroutines
AGRAZE	Zooplankton grazing (mg C L^{-1} s^{-1})	PHYPL
DDC	Detritus mineralisation (mg C L^{-1} s^{-1})	DETR, DISSNITR, PHOS, OXYGEN
DETFEK	Zooplankton grazing (leftovers) (mg C L^{-1} s^{-1})	DETR
DETGRZ	Zooplankton grazing on detritus (mg C L^{-1} s^{-1})	DETR
DIFNH	Sediment exchange of ammonium (g N m^{-2} s^{-1})	BAMMO, DISSNITR
DIFNO	Sediment exchange of nitrate (g N m^{-2} s^{-1})	BNITR, DISSNITR
DIFP	Sediment exchange of phosphate (g P m^{-2} s^{-1})	BPHOS, PHOS
ETPROP	Resuspension of sediments (s^{-1})	DETR, SEDORG
FMORT	Piscivorous fish mortality (mg C L^{-1} s^{-1})	SEDORG, PISCI
GRA	Cyanobacteria growth (mg C L^{-1} s^{-1})	CBACT, DISSNITR, PHOS, OXYGEN
GRAX	Cyanobacteria growth by nitrogen fixation (mg C L^{-1} s^{-1})	CBACT, PHOS, OXYGEN
GRO	Phytoplankton growth (mg C L^{-1} s^{-1})	PHYPL, DISSNITR, PHOS, OXYGEN
GROM	Macrophyte growth (mg C L^{-1} s^{-1})	MACKOL, OXYGEN
GMBN	Macrophyte growth, uptake of BNH (g N m^{-2} s^{-1})	BAMMO
GMBP	Macrophyte growth, uptake of BIP (g P m^{-2} s^{-1})	BPHOS
GMDN	Macrophyte growth, uptake of DIN (mg N L^{-1} s^{-1})	DISSNITR
GMDP	Macrophyte growth, uptake of PO4 (mg P L^{-1} s^{-1})	PHOS
MACM	Macrophyte mortality (mg C L^{-1} s^{-1})	MACKOL, SEDORG
MORTX	Mortality of cyanobacteria (mg C L^{-1} s^{-1})	CBACT, DETR
CBXGRZ	Zooplankton grazing (mg C L^{-1} s^{-1})	CBACT
NIT	Nitrification (g N m^{-2} s^{-1})	OXYGEN, BAMMO, BNITR
PREFI	Predation on planktivorous fish (mg C L^{-1} s^{-1})	DETR, FISHA, PISCI
PREZ	Predation on zooplankton (mg C L^{-1} s^{-1})	ZOOPL, DETR, FISHA
SDEN	Denitrification in sediment (g N m^{-2} s^{-1})	SEDORG, BPHOS, BAMMO, BNITR
SEDD	Sedimentation of detritus (g C m^{-2} s^{-1})	SEDORG
SEDP	Sedimentation of phytoplankton (g C m^{-2} s^{-1})	SEDORG
SEDX	Sedimentation of cyanobacteria (g C m^{-2} s^{-1})	SEDORG
SEQ	Sequestering of sediment organic matter (g C m^{-2} s^{-1})	SEDORG
SINKID	Sinking of detritus into cell (mg C L^{-1} s^{-1})	DETR
SINKIP	Sinking of phytoplankton into cell (mg C L^{-1} s^{-1})	PHYPL
SINKIX	Sinking of cyanobacteria into cell (mg C L^{-1} s^{-1})	CBACT
SINKOD	Sinking of detritus out of cell (mg C L^{-1} s^{-1})	DETR
SINKOP	Sinking of phytoplankton out of cell (mg C L^{-1} s^{-1})	PHYPL
SINKOX	Sinking of cyanobacteria out of cell (mg C L^{-1} s^{-1})	CBACT
SMIN	Mineralisation of sediment organic matter (g C m^{-2} s^{-1})	SEDORG, OXYGEN, BPHOS, BAMMO
WDEN	Denitrification in water column (mg N L^{-1} s^{-1})	DETR, DISSNITR, PHOS
ZGRAZE	Zooplankton grazing used for growth (mg C L^{-1} s^{-1})	ZOOPL
ZMORT	Zooplankton mortality rate (mg C L^{-1} s^{-1})	ZOOPL, DETR

The subroutine CBMORT calculates the mortality of cyanobacteria (MORTX). The process rate has the unit g C m^{-3} s^{-1} and uses the model parameter XMRT (Table 3).

```
C*********************************************************************
      SUBROUTINE CBMORT(MORTX)
C*********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL MORTX(NIM)

      DO I=NM1,2,-1
         MORTX(I)=XMRT/86400.*F(I,JANF)
         IF(F(I,JANF)-MORTX(I)*DT.LE.ANFMIN)MORTX(I)=0.
      ENDDO
      RETURN
      END
```

The mineralisation of detritus subroutine (DDCOMP) calculates the detritus mineralisation rate (DDC). It depends on the model parameters TREFG, TETAG and DDCC (Table 3) and uses the temperature function (MYTEMP). The rate has the unit g C m^{-3} s^{-1}.

```
C*********************************************************************
      SUBROUTINE DDCOMP(DDC)
C*********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL DDC(NIM),MYTEMP(NIM)

      CALL TEMPBER(TREFG,TETAG,MYTEMP)
      DO I=2,NM1
         DDC(I)=DDCC/86400.*MYTEMP(I)*F(I,JDET)
         IF(F(I,JDET)-DDC(I)*DT.LE.DETMIN) DDC(I)=0.
      ENDDO
      RETURN
      END
```

In the subroutine DENITR the denitrification rate in the water (WDEN) and in the sediment (SDEN) is calculated. Denitrification is temperature dependent and calls the TEMPBER subroutine. The oxygen concentration in the water determines where the denitrification takes place, in the water ($O2 \leq 0$) or in the sediment ($O2>0$). The denitrification depends on the model parameters ADEN, BDEN, TDEN, TETADEN and KBNO (Table 3). The unit is g N m^{-3} s^{-1} for denitrification rate in the water and g N m^{-2} s^{-1} for denitrification rate in the sediment.

```
C*********************************************************************
      SUBROUTINE DENITR(SDEN,WDEN)
C*********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL POSDEN,MYTEMP(NIM),SDEN(NIM),WDEN(NIM)

      CALL TEMPBER(TDEN,TETADEN,MYTEMP)
      DO I=2,NM1
         IF(F(I,JOXY).GT.0.0)THEN
            POSDEN=F(I,JBNO)/(KBNO*DSED+F(I,JBNO))
            SDEN(I)=ADEN/86400.*MYTEMP(I)*POSDEN
            WDEN(I)=0.0
         ELSE
```

```
            POSDEN=F(I,JDIN)/(KBNO+F(I,JDIN))
            SDEN(I)=0.0
            WDEN(I)=BDEN/86400.*MYTEMP(I)*POSDEN
         ENDIF
         IF(F(I,JBNO)-SDEN(I)*DT.LT.BNOMIN) SDEN(I)=0.
         IF(F(I,JDIN)-WDEN(I)*DT.LT.DINMIN) WDEN(I)=0.

      ENDDO
      RETURN
      END
```

Resuspension of sediment organic matter (ETPROP) is calculated by the subroutine ETADVECTION. The process uses the model parameter RESP (Table 3) and has as additional input the z-coordinate of the border between transport and erosion bottoms (ETD) and wind velocity (WG). ETPROP is not the resuspension rate, but give the part of sediment organic matter that will be resuspended. The unit is (s^{-1}). More details about this subroutine can be found in Ahlkrona (2002).

```
C*********************************************************************
           SUBROUTINE ETADVECTION(ETD,WG,ETPROP)
C*********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL ETD,WG,ETPROP(NIM)

      DO I = 2, N-1
         IF (Z(I).LT.ETD) THEN
            ETPROP(I) = 0.0
         ELSE
            ETPROP(I)=RESP*9.5866E-7*WG**2/600.
         ENDIF
      ENDDO
      RETURN
      END
```

The mortality of fish subroutine (FISHMORT) calculates the mortality rate of piscivorous fish (FMORT=FISHMRT). The rate has the unit g C m^{-3} s^{-1} and uses the model parameter FMRT (Table 3).

```
C*********************************************************************
      SUBROUTINE FISHMORT(FISHMRT)
C*********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL FISHMRT(NIM)

      DO I=2,NM1
         FISHMRT(I)=FMRT/86400.*F(I,JFIB)
         IF((F(I,JFIB)-FISHMRT(I)*DT).LE.FIBMIN) FISHMRT(I)=0.
      ENDDO
      RETURN
      END
```

In the zooplankton grazing subroutine (GRAZER) zooplankton grazing (ZGRAZE=ZZ) is calculated in addition to grazing on phytoplankton (AGRAZE=PP), on cyanobacteria (CBXGRZ=AA) and on detritus (DETGRZ=DD) respectively. All rates are in unit g C m^{-3} s^{-1}. Also the leftovers (DETFEK=D2) is calculated. Model parameters in the zooplankton grazing process are GAMMA, P1, P2, P3, KG, and EFFI (Table 3).

```
C**********************************************************************
      SUBROUTINE GRAZER(ZZ,GG,DD,AA,D2)
C**********************************************************************
       INCLUDE 'comp97.inc'
       INCLUDE 'biopar.inc'
       INTEGER I
       REAL PP,PD,PA
       REAL GG(NIM),DD(NIM),ZZ(NIM),AA(NIM),D2(NIM)

       DO I=2,NM1
          PP=P1*F(I,JPHY)/(P1*F(I,JPHY)+P2*F(I,JDET)+P3*F(I,JANF)+KG)*
     1      GAMMA/86400.*F(I,JZOO)
          PD=P2*F(I,JDET)/(P1*F(I,JPHY)+P2*F(I,JDET)+P3*F(I,JANF)+KG)*
     1      GAMMA/86400.*F(I,JZOO)
          PA=P3*F(I,JANF)/(P1*F(I,JPHY)+P2*F(I,JDET)+P3*F(I,JANF)+KG)*
     1      GAMMA/86400.*F(I,JZOO)
          IF(F(I,JPHY)-PP*DT.LE.PHYMIN)PP=0.
          IF(F(I,JDET)-PD*DT.LE.DETMIN)PD=0.
          IF(F(I,JANF)-PA*DT.LE.ANFMIN)PA=0.
          ZZ(I)=EFFI*(PP+PD+PA)
          GG(I)=PP
          DD(I)=PD
          AA(I)=PA
          D2(I)=(1-EFFI)/EFFI*ZZ(I)
       ENDDO
       RETURN
       END
```

The phytoplankton growth is calculated by the GROWTH subroutine. The phytoplankton growth rate (GRO=GR) depends on the LIGHT and TEMPBER subroutines, and on the variable telling whether macrophytes are modelled or not (MACOFF). The rate has the unit g C m^{-3} s^{-1} and uses the model parameters MYMAX, KDIN, KPO4, IOPTA, TPHY and TETAPHY (Table 3). The process has solar radiation as additional input (I0).

```
C**********************************************************************
      SUBROUTINE GROWTH(I0,GR,MACOFF)
C**********************************************************************
       INCLUDE 'comp97.inc'
       INCLUDE 'biopar.inc'
       INTEGER I,IH
       REAL X2(NIM),X3(NIM),GR(NIM),I0,LGHW
       REAL MYLGTH(NIM,NIM),MYTEMP(NIM)

       CALL LIGHT(I0,IOPTA,MYLGTH,MACOFF)
       CALL TEMPBER(TPHY,TETAPHY,MYTEMP)

       DO I=2,NM1
C-----NITROGEN/PHOSPHORUS LIMITATION
           X2(I)=F(I,JDIN)/(KDIN+F(I,JDIN))
           X3(I)=F(I,JPO4)/(KPO4+F(I,JPO4))
C-----PHYTOPLANKTON GROWTH
           IF (X2(I).LE.X3(I)) THEN
              GR(I)=MYMAX/86400.*MYTEMP(I)*X2(I)*F(I,JPHY)
           ELSE
              GR(I)=MYMAX/86400.*MYTEMP(I)*X3(I)*F(I,JPHY)
           ENDIF
           LGHW=0.0
           DO IH=2,I
              LGHW=LGHW+MYLGTH(I,IH)*(AREA(IH)-AREA(IH-1))/AREA(I)
           ENDDO
           GR(I)=GR(I)*LGHW
       ENDDO
```

```
      RETURN
      END
```

The subroutine MACMORT calculates the mortality of macrophytes (MACM=MMORT) if macrophytes is included in the simulation (controled by variable MACOFF). The rate has the unit g C m^{-3} s^{-1} and uses the model parameter MMRT (Table 3).

```
C**********************************************************************
      SUBROUTINE MACMORT(MMORT,MACOFF)
C**********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL MMORT(NIM)
      LOGICAL MACOFF

      IF (MACOFF) GOTO 50
      DO I=2,NM1
         MMORT(I)=MMRT/86400.*F(I,JMAC)
         IF (F(I,JMAC)-MMORT(I)*DT.LE.MACMIN) MMORT(I)=0.
      ENDDO
      RETURN

50    DO I=2,NM1
         MMORT(I)=0.
      ENDDO
      RETURN
      END
```

The macrophyte growth is calculated by the MACROF subroutine. Macrophytes can grow on nutrients from the water and on nutrients from the sediment. Therefore four different variables give the uptake of nutrients (GMBN for sediment ammonium, GMBP for sediment phosphate, GMDN for DIN, and GMDP for phosphate), while growth rate is given by GROM (=GRM). The growth depends on the LIGHT and TEMPBER subroutines, and on the variable telling whether macrophytes are modelled or not (MACOFF). GMBN and GMBP have unit g N/P m^{-2} s^{-1}, while GMDN, GMDP and GRM have unit g N/P/C m^{-3} s^{-1}. The process depends on the model parameters MYMAXM, IOPTM, TMAC, TETAMAC, KMIN, KMIP (Table 3). The process has solar radiation as additional input (I0).

```
C**********************************************************************
      SUBROUTINE MACROF(I0,GRM,GMBN,GMDN,GMBP,GMDP,MACOFF)
C**********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I,ID
      REAL X2(NIM),X3(NIM),GRM(NIM),I0,LGHW
      REAL MYLGTH(NIM,NIM),MYTEMP(NIM)
      REAL GMBP(NIM),GMDP(NIM),GMBN(NIM),GMDN(NIM)
      REAL X4(NIM),X5(NIM)
      LOGICAL MACOFF
C
      IF (MACOFF) GOTO 50
      CALL LIGHT(I0,IOPTM,MYLGTH,MACOFF)
      CALL TEMPBER(TMAC,TETAMAC,MYTEMP)

      DO I=2,NM1
C-----NITROGEN/PHOSPHORUS LIMITATION
         X2(I)=(F(I,JBNH)/DSED2+F(I,JDIN))/
     1           (KMIN+F(I,JBNH)/DSED2+F(I,JDIN))
         X3(I)=(F(I,JBIP)/DSED2+F(I,JPO4))/
     1           (KMIP+F(I,JBIP)/DSED2+F(I,JPO4))
```

```
      IF (X2(I).LE.X3(I)) THEN
         GRM(I)=MYMAXM/86400.*MYTEMP(I)*X2(I)*F(I,JMAC)
      ELSE
         GRM(I)=MYMAXM/86400.*MYTEMP(I)*X3(I)*F(I,JMAC)
      ENDIF
      LGHW=0.0
      DO ID=I,NM1
         LGHW=LGHW+MYLGTH(ID,I)*DZCELL(ID)/(ZDIM-ZBOUND(I-1))
      ENDDO
      GRM(I)=GRM(I)*LGHW
      X4(I)=CNC*GRM(I)*(ZDIM-ZBOUND(I-1))
      X5(I)=CPC*GRM(I)*(ZDIM-ZBOUND(I-1))
      IF (X4(I)*DT.LT.(F(I,JBNH)-BNHMIN)) THEN
         GMBN(I)=X4(I)
         GMDN(I)=0.0
      ELSEIF(F(I,JBNH).GT.BNHMIN)THEN
         GMBN(I)=F(I,JBNH)-BNHMIN
         GMDN(I)=X4(I)/DZCELL(I)-GMBN(I)/DZCELL(I)
         IF (F(I,JDIN)-GMDN(I)*DT.LT.DINMIN) GOTO 49
      ELSE
         GMBN(I)=0.
         GMDN(I)=X4(I)/DZCELL(I)
         IF (F(I,JDIN)-GMDN(I)*DT.LT.DINMIN) GOTO 49
      ENDIF
      IF (X5(I)*DT.LT.(F(I,JBIP)-BIPMIN)) THEN
         GMBP(I)=X5(I)
         GMDP(I)=0.0
      ELSEIF (F(I,JBIP).GT.BIPMIN)THEN
         GMBP(I)=F(I,JBIP)-BIPMIN
         GMDP(I)=X5(I)/DZCELL(I)-GMBP(I)/DZCELL(I)
         IF (F(I,JPO4)-GMDP(I)*DT.LT.PO4MIN) GOTO 49
      ELSE
         GMBP(I)=0.
         GMDP(I)=X5(I)/DZCELL(I)
         IF (F(I,JPO4)-GMDP(I)*DT.LT.PO4MIN) GOTO 49
      ENDIF
      RETURN

49    GRM(I)=0.
      GMBP(I)=0.
      GMDP(I)=0.
      GMBN(I)=0.
      GMDN(I)=0.
   ENDDO
   RETURN
50 DO I=2,NM1
      GRM(I)=0.
      GMBP(I)=0.
      GMDP(I)=0.
      GMBN(I)=0.
      GMDN(I)=0.
   ENDDO
   RETURN
   END
```

The cyanobacteria growth is calculated by the `CBGROWTH` subroutine. The cyanobacteria growth depends on the `LIGHT` and `TEMPBER` subroutines, and on the variable telling whether macrophytes are modelled or not (`MACOFF`). The cyanobacteria growth rate is divided in two, one for growth using DIN (`GRA`) and one for nitrogen fixation (`GRAX`). The rates have the unit g C m^{-3} s^{-1} and use the model parameters `MYMAXA`, `MYMAXB`, `KDIN`, `KPO4`, `IOPTA`, `TPHY` and `TETAPHY` (Table 3). The process has solar radiation as additional input (`I0`).

```
C***********************************************************************
      SUBROUTINE CBGROWTH(I0,GRA,GRAX,MACOFF)
C***********************************************************************
       INCLUDE 'comp97.inc'
       INCLUDE 'biopar.inc'
       INTEGER I,IH
       REAL X2(NIM),X3(NIM),GRA(NIM),GRAX(NIM),I0,LGHW
       REAL MYLGTH(NIM,NIM),MYTEMP(NIM)

       CALL LIGHT(I0,IOPTA,MYLGTH,MACOFF)
       CALL TEMPBER(TPHY,TETAPHY,MYTEMP)

      DO I=2,NM1
C-----NITROGEN/PHOSPHORUS LIMITATION
          X2(I)=F(I,JDIN)/(KDIN+F(I,JDIN))
          X3(I)=F(I,JPO4)/(KPO4+F(I,JPO4))
C-----CYANOBACTERIA GROWTH
          IF (X2(I).LE.X3(I)) THEN
              GRAX(I)=MYMAXA/86400.*MYTEMP(I)*
     1          X3(I)/(1+(F(I,JDIN)/F(I,JPO4)/KNFIX)**SNF4)*F(I,JANF)
              GRA(I)=0.
          ELSE
              GRAX(I)=0.
              GRA(I)=MYMAXB/86400.*MYTEMP(I)*X3(I)*F(I,JANF)
          ENDIF
          LGHW=0.0
          DO IH=2,I
              LGHW=LGHW+MYLGTH(I,IH)*(AREA(IH)-AREA(IH-1))/AREA(I)
          ENDDO
          GRA(I)=GRA(I)*LGHW
          GRAX(I)=GRAX(I)*LGHW
      ENDDO
      RETURN
      END
```

Nitrification in the sediment is calculated by the `NITFIC` subroutine. The nitrification rate (`NIT`) depends on the temperature function (subroutine `TEMPBER`). Nitrification depends on the model parameters `TSC`, `TNFIC` and `TETANFIC` (Table 3), and is given in unit g C m^{-2} s^{-1}.

```
C***********************************************************************
      SUBROUTINE NITFIC(NIT)
C***********************************************************************
       INCLUDE 'comp97.inc'
       INCLUDE 'biopar.inc'
       INTEGER I
       REAL NIT(NIM),MYTEMP(NIM)

       CALL TEMPBER(TNFIC,TETANFIC,MYTEMP)

      DO I=2,NM1
          NIT(I)=TSC/86400.*MYTEMP(I)*
     1    MAX(MIN(F(I,JBNH),F(I,JOXY)/CONNITR/DSED2),0.)
          IF ((F(I,JBNH)-NIT(I)*DT.LE.BNHMIN).OR.
     1       (F(I,JOXY).LE.NIT(I)*DT *CONNITR*
     1       (AREA(I)-AREA(I-1))/AREA(I)/DZCELL(I))) THEN
              NIT(I)=0.
          ENDIF
      ENDDO
      RETURN
      END
```

The predation on fish subroutine (`PREDATF`) calculates the piscivorous fish predation on planktivorous fish (`PREFI`). The predation causes growth of piscivorous fish given by the efficiency parameter OMEGA (Table 3). The growth and leftovers are calculated directly in the source subroutines (FISHA, FISHB). The rate has the unit g C m^{-3} s^{-1} and uses the model parameter PREDF, TETAG, TREFG and KFIA (Table 3).

```
C*********************************************************************
      SUBROUTINE PREDATF(PREFI)
C*********************************************************************
       INCLUDE 'comp97.inc'
       INCLUDE 'biopar.inc'
       INTEGER I
       REAL PREFI(NIM)

      DO I=2,NM1
         PREFI(I)=PREDF/86400.*EXP(LOG(TETAG)*(F(I,JTE)-TREFG))*
     1      F(I,JFIA)**2/(KFIA**2+F(I,JFIA)**2)*F(I,JFIB)
         IF((F(I,JFIA)-PREFI(I)*DT).LE.FIAMIN) PREFI(I)=0.
      ENDDO
      RETURN
      END
```

The predation on zooplankton subroutine (PREDATZ) calculates the loss of zooplankton (PREZ). The rate has the unit g C m^{-3} s^{-1}. The predation causes growth of planktivorous fish with the same efficiency as for predation on fish (OMEGA). Growth and leftover are calculated directly in the source subroutines (DETR, FISHA). The predation subroutine used the model parameters PREDZ, TETAG, TREFG, KZO (Table 3).

```
C*********************************************************************
      SUBROUTINE PREDATZ(PREZ)
C*********************************************************************
       INCLUDE 'comp97.inc'
       INCLUDE 'biopar.inc'
       INTEGER I
       REAL PREZ(NIM)

      DO I=2,NM1
         PREZ(I)=PREDZ/86400.*EXP(LOG(TETAG)*(F(I,JTE)-TREFG))*
     1      F(I,JZOO)**2/(KZO**2+F(I,JZOO)**2)*F(I,JFIA)
         IF((F(I,JZOO)-PREZ(I)*DT).LE.ZOOMIN) PREZ(I)=0.
      ENDDO
      RETURN
      END
```

The exchange of nutrients between the water and the sediment are calculated by the subroutine SEDDIFF. The variable FAKTA is an input argument. It is the quotient between sediment phosphorus that is absorbed to particles and sediment phosphorus that is dissolved. Only the dissolved sediment phosphorus is included in the exchange. The exchange rates are DIFP (=RP), DIFNH (=RNH) and DIFNO (=RNO) for phosphorus, ammonium and nitrate respective. The sediment diffusion depends on the model parameters KDIFF and HAR (Table 3) and the unit is g N/P m^{-2} s^{-1}.

```
C*********************************************************************
      SUBROUTINE SEDDIFF(RP,RNO,RNH,FAKTA)
C*********************************************************************
       INCLUDE 'comp97.inc'
       INCLUDE 'biopar.inc'
       INTEGER I
       REAL RP(NIM),RNH(NIM),RNO(NIM),FAKTA(NIM)

       DO I=2,NM1
         RP(I)=KDIFF*(F(I,JBIP)/(1+FAKTA(I))/HAR-F(I,JPO4))/HAR
```

```
   RNH(I)=KDIFF*F(I,JBNH)/HAR/HAR
   RNO(I)=KDIFF*(F(I,JBNO)-F(I,JDIN)*DZCELL(I))/HAR/HAR
ENDDO
RETURN
END
```

In the subroutine SEDMINE is mineralisation of sediment organic matter (SMIN) calculated. Sediment denitrification is an input argument, as the sediment mineralisation formula calculates the total mineralisation from which denitrification is subtracted before the variable SMIN is assigned. Sediment mineralisation is temperature dependent and calls the TEMPBER subroutine. The rate has the unit g C m^{-2} s^{-1} and uses the model parameters MINMAX, TSMIN and TETASMIN (Table 3).

```
C***********************************************************************
      SUBROUTINE SEDMINE(SMIN,SDEN)
C***********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL POSDEN,MYTEMP(NIM),SMIN(NIM),SDEN(NIM)

      CALL TEMPBER(TSMIN,TETASMIN,MYTEMP)
      DO I=2,NM1
         SMIN(I)=MAX(MINMAX*MYTEMP(I)*F(I,JSED)-SDEN(I)*CCNDENIT,0.)
      ENDDO
      RETURN
      END
```

In the subroutine SEQUEST sequestering of sediment organic matter (SEQ) is calculated. The rate has the unit g C m^{-2} s^{-1} and uses the model parameter SEQCON (Table 3).

```
C***********************************************************************
      SUBROUTINE SEQUEST(SEQ)
C***********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL SEQ(NIM)

      DO I=2,NM1
         SEQ(I)=SEQCON*F(I,JSED)
         IF(F(I,JSED)-SEQ(I)*DT.LT.SEDMIN) SEQ(I)=0.
      ENDDO
      RETURN
      END
```

The subroutines SINKPHY, SINKDET and SINKCB calculate the sinking and settling of phytoplankton, detritus and cyanobacteria respectively. The subroutines are similar with output arguments SINKOP/SINKOD/SINKOX (=SINKO), SINKIP/SINKID/SINKIX (=SINKI) and SEDP/SEDD/SEDX (=SED), which denotes material sinking out of the cell (g C m^{-3} s^{-1}), sinking into the next cell (g C m^{-3} s^{-1}), and settling on the bottom of the cell (g C m^{-2} s^{-1}). The model parameters SINKP, SINKD and SINKAN are used.

```
C***********************************************************************
      SUBROUTINE SINKCB(SINKO,SINKI,SED)
C***********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL SINKO(NIM),SINKI(NIM),SED(NIM)
```

```
C-----SINKING OF CYANOBACTERIA
      DO I=NM1,2,-1
         SINKO(I)=SINKAN/86400./DZCELL(I)*F(I,JANF)
         IF(F(I,JANF)-SINKO(I)*DT.LE.ANFMIN)SINKO(I)=0.
         SINKI(I-1)=SINKO(I)*DZCELL(I)/DZCELL(I-1)
         SED(I)=SINKO(I)*DZCELL(I)
      ENDDO
      SINKI(NM1)=0.
      SINKI(1)=0.
      RETURN
      END

C*********************************************************************
      SUBROUTINE SINKDET(SINKO,SINKI,SED)
C*********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL SINKO(NIM),SINKI(NIM),SED(NIM)

C-----DETRITUS SINKING
      DO I=NM1,2,-1
         SINKO(I)=SINKD/86400./DZCELL(I)*F(I,JDET)
         IF(F(I,JDET)-SINKO(I)*DT.LE.DETMIN)SINKO(I)=0.
         SINKI(I-1)=SINKO(I)*DZCELL(I)/DZCELL(I-1)
         SED(I)=SINKO(I)*DZCELL(I)
      ENDDO
      SINKI(NM1)=0.
      SINKI(1)=0.
      RETURN
      END

C*********************************************************************
      SUBROUTINE SINKPHY(SINKO,SINKI,SED)
C*********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL SINKO(NIM),SINKI(NIM),SED(NIM)

C-----PHYTOPLANKTON SINKING (POSITIVE DOWNWARD)
      DO I=NM1,2,-1
         SINKO(I)=SINKP/86400./DZCELL(I)*F(I,JPHY)
         IF(F(I,JPHY)-SINKO(I)*DT.LE.PHYMIN)SINKO(I)=0.
         SINKI(I-1)=SINKO(I)*DZCELL(I)/DZCELL(I-1)
         SED(I)=SINKO(I)*DZCELL(I)
      ENDDO
      SINKI(NM1)=0.
      SINKI(1)=0.
      RETURN
      END
```

The mortality of zooplankton subroutine (ZOOMORT) calculates the mortality rate (ZMORT=ZNMRT). The rate has the unit g C m^{-3} s^{-1} and uses the model parameter ZMRT (Table 3).

```
C*********************************************************************
      SUBROUTINE ZOOMORT(ZNMRT)
C*********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL ZNMRT(NIM)
```

```
      DO I=2,NM1
         ZNMRT(I)=ZMRT/86400.*F(I,JZOO)
         IF((F(I,JZOO)-ZNMRT(I)*DT).LE.ZOOMIN) ZNMRT(I)=0.
      ENDDO
      RETURN
      END
```

Here follow two subroutines called by several of the process rate subroutines above.

The subroutine `LIGHT` calculates the light extinction and light dependence function (`MYLGHT`) of algal growth. The subroutine is called from the subroutines `GROWTH`, `CBGROWTH` and `MACROF`. Input arguments are the current light (`I0`), the optimum light for the alga specie (`IOPT`), and the variable telling if macrophytes are included in the model (`MACOFF`). The model parameters `KPKONST`, `KDKONST`, `KMKONST` and `IZKONST` are used (Table 3).

```
C**********************************************************************
      SUBROUTINE LIGHT(I0,IOPT,MYLGHT,MACOFF)
C**********************************************************************
       INCLUDE 'comp97.inc'
       INCLUDE 'biopar.inc'
       INTEGER ID,IH,I
       LOGICAL MACOFF
       REAL I0,K(NIM,NIM),K2(NIM,NIM),IZ(NIM,NIM),IOPT
       REAL KP,KD,KM
       REAL MYLGHT(NIM,NIM)

      DO ID=NM1,2,-1
          DO IH=ID,2,-1
C-----EXTINCTION COEFFICIENT K
C-----------SELF SHADING OF PHYTOPLANKTON
             KP=KPKONST*(F(ID,JPHY)+F(ID,JANF))
C-----------SHADING BY DETRITUS
             KD=KDKONST*F(ID,JDET)
C-----------SHADING BY MACROPHYTES
             IF(MACOFF)THEN
                KM=0.
             ELSE
                KM=KMKONST*F(IH,JMAC)
             ENDIF
             K(ID,IH)=KW+KP+KD+KM
             K2(ID,IH)=0.0
          ENDDO
       ENDDO
       DO ID=NM1,2,-1
         DO IH=ID,2,-1
           DO I=ID,NM1
C-----------THE EXPONENT OF THE LIGHT FUNCTION
             K2(ID,IH)=K2(ID,IH)+K(I,IH)*(Z(I+1)-Z(I))
           ENDDO
           IZ(ID,IH)=IZKONST*I0*EXP(-K2(ID,IH))
           MYLGHT(ID,IH)=(IZ(ID,IH)/IOPT)*EXP(1.-IZ(ID,IH)/IOPT)
         ENDDO
       ENDDO
      RETURN
      END
```

The subroutine `TEMPBER` calculates the temperature dependence function (`MYTEMP`) from the coefficients `TETA` and `TKOEFF`. The coefficients of the temperature dependence are different parameters for different processes. The temperature dependence function is used in subroutines `DDCOMP`, `GROWTH`, `CBGROWTH`, `MACROF`, `NITFIC`, `DENITR` and `SEDMINE`.

```
C*********************************************************************
      SUBROUTINE TEMPBER(TKOEFF,TETA,MYTEMP)
C*********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL MYTEMP(NIM)
      REAL TKOEFF,TETA
      DO I=2,NM1
         MYTEMP(I)=EXP(LOG(TETA)*(F(I,JTE)-TKOEFF))
      ENDDO
      RETURN
      END
```

5.3 The source and sink subroutines

There is one source and sink subroutine for each state variable in BIOLA. Input arguments to the subroutine are the process rates, while the accumulated process flows are output arguments. The definitions of the accumulated process flows can be found in Table 11, while the definitions of the process rates are gathered in Table 12. The main purpose of a source and sink subroutine is to set the `SI` and `SIP` arrays. These are `PROBE`'s variables for two different formulations of sources. `SI` corresponds directly to Φ in eq. (1-2), while `SIP` correspond to Φ/C. Traditionally sources are put in `SI` and sinks in `SIP` when programming with PROBE to get a stable equation to solve. Last in the subroutine the process flow variables are accumulated.

In the zooplankton subroutine (`ZOOPL`), three processes are included; zooplankton grazing (`ZGRAZE`), predation on zooplankton (`PREZ`) and zooplankton mortality (`ZMORT`). Predation on zooplankton and zooplankton mortality are sinks to zooplankton and therefore put in SIP. The accumulated process flows are integrated over the vertical and time.

```
C*********************************************************************
      SUBROUTINE ZOOPL(ZGRAZE,PREZ,ZMORT,TOTGRZ,TOTPRED,TOTZMRT)
C*********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL ZGRAZE(NIM),PREZ(NIM),ZMORT(NIM),TOTGRZ,TOTPRED,TOTZMRT

C-----ASSIGN VALUES TO SI AND SIP
      DO I=2,NM1
         SI(I)=ZGRAZE(I)
         SIP(I)=-PREZ(I)/F(I,JZOO)-ZMORT(I)/F(I,JZOO)
C-----VERTICAL INTEGRATION OF PROCESSES
         TOTGRZ=TOTGRZ+ZGRAZE(I)*DZCELL(I)*AREA(I)*DT
         TOTPRED=TOTPRED+PREZ(I)*DZCELL(I)*AREA(I)*DT
         TOTZMRT=TOTZMRT+ZMORT(I)*DZCELL(I)*AREA(I)*DT
      ENDDO
      RETURN
      END
```

In the phytoplankton subroutine (`PHYPL`), three processes are included; phytoplankton growth (`GRO`), zooplankton grazing (`AGRAZE`), and sinking. Phytoplankton sinking out of a cell (`SINKOP`) is separated into two parts one settling on the bottom of the cell, and one sinking into the next cell (`SINKIP`).

```
C*********************************************************************
      SUBROUTINE PHYPL(GRO,AGRAZE,SINKOP,SINKIP,
     1 TOTASINKI,TOTGRW,TOTASINKO,TOTAGRZ)
C*********************************************************************
       INCLUDE 'comp97.inc'
       INCLUDE 'biopar.inc'
       INTEGER I
       REAL GRO(NIM),AGRAZE(NIM),TOTASINKI,TOTGRW,TOTASINKO,TOTAGRZ
       REAL SINKOP(NIM),SINKIP(NIM)

C-----ASSIGN VALUES TO SI AND SIP
      DO I=2,NM1
          SI(I)=SINKIP(I)+GRO(I)
          SIP(I)=-SINKOP(I)/F(I,JPHY)-AGRAZE(I)/F(I,JPHY)
C-----VERTICAL INTEGRATION OF PROCESSES
          TOTASINKI=TOTASINKI+SINKIP(I)*DZCELL(I)*DT*AREA(I)
          TOTGRW=TOTGRW+GRO(I)*DZCELL(I)*AREA(I)*DT
          TOTASINKO=TOTASINKO+SINKOP(I)*DZCELL(I)*DT*AREA(I)
          TOTAGRZ=TOTAGRZ+AGRAZE(I)*DZCELL(I)*AREA(I)*DT
      ENDDO
      RETURN
      END
```

The detritus subroutine (DETR) depends on a lot of processes. Detritus sinking out of a cell (SINKOD) is separated into two parts, one settling on the bottom of the cell and one sinking into the next cell (SINKID). Detritus may be grazed by zooplankton (DETGRZ). During grazing zooplankton does not use all food for growth but there is leftovers (DETFEK). Also during planktivorous fish predation on zooplankton (PREZ) and piscivorous fish predation (PREFI), there are leftovers. Only OMEGA part of the predation is used for growth. Zooplankton mortality (ZMORT), cyanobacteria mortality (XMORT) and resuspension (ETPROP) are other sources of detritus. Resuspension rate is given per unit time and is therefore multiplied with SED. Only the bottom of the cell (AREA(I)-AREA(I-1)) releases sediment organic matter, but the resulting detritus is spread in the cell volume. Mineralisation (DDC) is a sink. Denitrification (WDEN) is a sink of detritus, and the ratio between denitrified nitrogen (WDEN) and degraded detritus is constant (CCNDENIT).

```
C*********************************************************************
      SUBROUTINE DETR(DETFEK,DETGRZ,PREZ,PREFI,ZMORT,WDEN,DDC,
     1 SINKOD,SINKID,MORTX,ETPROP,TOTDSINKI,TOTDPRED,TOTDFEK,TOTDFI,
     1 TOTDDEN,TOTDSINKO,TOTDECOM,TOTDGRZ,TOTDZM,TOTDMRTX,TOTDETP)
C*********************************************************************
       INCLUDE 'comp97.inc'
       INCLUDE 'biopar.inc'
       INTEGER I
       REAL SINKID(NIM),SINKOD(NIM)
       REAL DDC(NIM),WDEN(NIM),ZMORT(NIM),MORTX(NIM),ETPROP(NIM)
       REAL DETFEK(NIM),DETGRZ(NIM),PREZ(NIM),PREFI(NIM)
       REAL TOTDSINKI,TOTDPRED,TOTDFEK,TOTDFI,TOTDDEN
       REAL TOTDSINKO,TOTDECOM,TOTDGRZ,TOTDZM,TOTDMRTX,TOTDETP

C-----ASSIGN VALUES TO SI AND SIP
      DO I=2,NM1
          SI(I)=SINKID(I)+DETFEK(I)+(1-OMEGA)*PREZ(I)+MORTX(I)+
     &         (1-OMEGA)*PREFI(I)+ZMORT(I)-DDC(I)-DETGRZ(I)
          SIP(I)=-SINKOD(I)/F(I,JDET)+ETPROP(I)*F(I,JSED)
     &        /F(I,JDET)*(AREA(I)-AREA(I-1))/(AREA(I)*DZCELL(I))
     &        -WDEN(I)*CCNDENIT/F(I,JDET)
C-----VERTICAL INTEGRATION OF PROCESSES
          TOTDSINKI=TOTDSINKI+SINKID(I)*DZCELL(I)*DT*AREA(I)
          TOTDPRED=TOTDPRED+(1-OMEGA)*PREZ(I)*DZCELL(I)*DT*AREA(I)
          TOTDFEK=TOTDFEK+DETFEK(I)*DZCELL(I)*DT*AREA(I)
```

```
        TOTDFI=TOTDFI+(1-OMEGA)*PREFI(I)*DZCELL(I)*DT*AREA(I)
        TOTDDEN=TOTDDEN+WDEN(I)*CCNDENIT*DZCELL(I)*DT*AREA(I)
        TOTDSINKO=TOTDSINKO+SINKOD(I)*DZCELL(I)*DT*AREA(I)
        TOTDECOM=TOTDECOM+DDC(I)*DZCELL(I)*DT*AREA(I)
        TOTDGRZ=TOTDGRZ+DETGRZ(I)*DZCELL(I)*DT*AREA(I)
        TOTDZM=TOTDZM+ZMORT(I)*DZCELL(I)*DT*AREA(I)
        TOTDMRTX=TOTDMRTX+MORTX(I)*DZCELL(I)*DT*AREA(I)
        TOTDETP=TOTDETP+ETPROP(I)*F(I,JSED)*(AREA(I)-AREA(I-1))*DT
      ENDDO
      RETURN
      END
```

The sediment organic carbon subroutine (SEDORG) is influenced by the following processes: detritus, phytoplankton and cyanobacteria settling on the bottom (SEDD, SEDP, SEDX), sequestering of sediment organic carbon (SEQ), macrophyte mortality (MACM), piscivorous fish mortality (FMORT), resuspension (ETPROP), and mineralisation (SMIN). Macrophytes and fish are water variables and thus defined per volume. These have to be recalculated when reaching the bottom since sediment organic matter is defined per bottom area. The macrophyte variable denotes the concentration in a column above the bottom of cell I, so its mortality is integrated over the column height. Fish mortality is integrated over the cell volume and spread over the bottom area. Denitrification (SDEN) is proportional to the denitrified nitrogen (CCNDENIT).

```
C*******************************************************************
      SUBROUTINE SEDORG(SDEN,SMIN,MACM,SEQ,SEDP,SEDD,SEDX,ETPROP,
     1 FMORT,TOTSSINKP,TOTSSINKD,TOTSSINKN,TOTSMACM,TOTSDENI,
     1 TOTSMIN,TOTSMFI,TOTETP,TOTSEQ)
C*******************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL SDEN(NIM),SMIN(NIM),MACM(NIM),SEQ(NIM),
     & SEDP(NIM),SEDD(NIM),SEDX(NIM),FMORT(NIM),ETPROP(NIM)
      REAL TOTSSINKP,TOTSSINKD, TOTSSINKN, TOTSMACM,
     & TOTSDENI,TOTSMIN,TOTSMFI,TOTETP,TOTSEQ

C-----ASSIGN VALUES TO SI AND SIP
      DO I=2,NM1
        SI(I)=SEDP(I)+SEDD(I)+SEDX(I)+MACM(I)*(ZDIM-ZBOUND(I-1))
     &       +FMORT(I)*DZCELL(I)*AREA(I)/(AREA(I)-AREA(I-1))
        SIP(I)=-SMIN(I)/F(I,JSED)-ETPROP(I)
     &       -SEQ(I)/F(I,JSED)
     &       -SDEN(I)*CCNDENIT/F(I,JSED)
C-----VERTICAL INTEGRATION OF PROCESSES
        TOTSSINKP=TOTSSINKP+SEDP(I)*(AREA(I)-AREA(I-1))*DT
        TOTSSINKD=TOTSSINKD+SEDD(I)*(AREA(I)-AREA(I-1))*DT
        TOTSSINKN=TOTSSINKN+SEDX(I)*(AREA(I)-AREA(I-1))*DT
        TOTSMACM=TOTSMACM+MACM(I)*(ZDIM-ZBOUND(I-1))*
     &          (AREA(I)-AREA(I-1))*DT
        TOTSDENI=TOTSDENI+SDEN(I)*CCNDENIT*(AREA(I)-AREA(I-1))*DT
        TOTSMIN=TOTSMIN+SMIN(I)*(AREA(I)-AREA(I-1))*DT
        TOTSMFI=TOTSMFI+FMORT(I)*DZCELL(I)*DT*AREA(I)
        TOTETP=TOTETP+ETPROP(I)*F(I,JSED)*(AREA(I)-AREA(I-1))*DT
        TOTSEQ=TOTSEQ+SEQ(I)*(AREA(I)-AREA(I-1))*DT
      ENDDO
      RETURN
      END
```

In the dissolved inorganic nitrogen subroutine (DISSNITR) the following processes are included: denitrification in the water column (WDEN), uptake for growth of phytoplankton, cyanobacteria and macrophytes (GRO, GRA, GMDN), exchange with nutrients in the sediment

(DIFNO, DIFNH) and mineralisation of detritus (DDC). Mineralisation of detritus is calculated as rate of detritus degraded. Corresponding amount of nitrogen released is given by the constant CNC. Denitrification is a sink of DIN, but as the process is combined with degradation of detritus it is also a small source. The nitrogen bound in detritus is released as the detritus is degraded. Exchange of nutrients between the water and the sediment is calculated per unit bottom area, so the process rates have to be recalculated to be included in this subroutine. It is added to SI or SIP depending on direction of the flow.

```
C*********************************************************************
      SUBROUTINE DISSNITR(WDEN,GRO,GRA,DIFNO,DIFNH,GMDN,DDC,
     1 TOTDIFFN,TOTWDENI,TOTDDC4,TOTGRO,TOTGRA,TOTGROM)
C*********************************************************************
       INCLUDE 'comp97.inc'
       INCLUDE 'biopar.inc'
       INTEGER I
       REAL WDEN(NIM),GRO(NIM),GRA(NIM)
       REAL DIFNO(NIM),DIFNH(NIM),GMDN(NIM)
       REAL DDC(NIM)
       REAL TOTDIFFN,TOTWDENI,TOTDDC4,TOTGRO,TOTGRA,TOTGROM

C-----ASSIGN VALUES TO SI AND SIP
       DO I=2,NM1
          SI(I)=DDC(I)*CNC+
     &         WDEN(I)*CCNDENIT*CNC
          SIP(I)=-GRO(I)*CNC/F(I,JDIN)-WDEN(I)/F(I,JDIN)
     &          -GRA(I)*CNC/F(I,JDIN)-GMDN(I)/F(I,JDIN)
          IF(DIFNO(I)+DIFNH(I).GT.0.)THEN
             SI(I)=SI(I)+(DIFNO(I)+DIFNH(I))*(AREA(I)-AREA(I-1))/
     &         AREA(I)/DZCELL(I)
          ELSE
             SIP(I)=SIP(I)+(DIFNO(I)+DIFNH(I))*(AREA(I)-AREA(I-1))/
     &         AREA(I)/DZCELL(I)/F(I,JDIN)
          ENDIF
C-----VERTICAL INTEGRATION OF PROCESSES
          TOTDIFFN=TOTDIFFN+(DIFNO(I)+DIFNH(I))*
     1          (AREA(I)-AREA(I-1))*DT
          TOTWDENI=TOTWDENI+WDEN(I)*CCNDENIT*CNC*DZCELL(I)*
     1          DT*AREA(I)-WDEN(I)*DZCELL(I)*DT*AREA(I)
          TOTDDC4=TOTDDC4+DDC(I)*CNC*DZCELL(I)*DT*AREA(I)
          TOTGRO=TOTGRO+GRO(I)*CNC*DZCELL(I)*DT*AREA(I)
          TOTGRA=TOTGRA+GRA(I)*CNC*DZCELL(I)*DT*AREA(I)
          TOTGROM=TOTGROM+GMDN(I)*DZCELL(I)*DT*AREA(I)
       ENDDO
       RETURN
       END
```

The subroutine for cyanobacteria (CBACT) is influenced by the following processes: growth of cyanobacteria using DIN and N_2 (GRA, GRAX), cyanobacteria mortality (MORTX), grazing by zooplankton (CBXGRZ) and cyanobacteria sinking out of the cell (SINKOX) and into the cell from above (SINKIX).

```
C*********************************************************************
      SUBROUTINE CBACT(GRA,GRAX,CBXGRZ,SINKOX,SINKIX,MORTX,
     1 TOTNSINKI,TOTNSINKO,TOTNGRA,TOTNGRAX,TOTXGRZ,TOTXMRT)
C*********************************************************************
       INCLUDE 'comp97.inc'
       INCLUDE 'biopar.inc'
       INTEGER I
       REAL SINKIX(NIM),SINKOX(NIM),MORTX(NIM)
       REAL GRA(NIM),CBXGRZ(NIM),GRAX(NIM)
       REAL TOTNSINKI,TOTNSINKO,TOTNGRA,TOTNGRAX,TOTXGRZ,TOTXMRT
```

```
C-----ASSIGN VALUES TO SI AND SIP
      DO I=2,NM1
          SI(I)=SINKIX(I)+GRA(I)+GRAX(I)
          SIP(I)=-SINKOX(I)/F(I,JANF)-CBXGRZ(I)/F(I,JANF)
     1    -MORTX(I)/F(I,JANF)
C-----VERTICAL INTEGRATION OF PROCESSES
          TOTNSINKI=TOTNSINKI+SINKIX(I)*DZCELL(I)*DT*AREA(I)
          TOTNSINKO=TOTNSINKO+SINKOX(I)*DZCELL(I)*DT*AREA(I)
          TOTNGRA=TOTNGRA+GRA(I)*DZCELL(I)*DT*AREA(I)
          TOTNGRAX=TOTNGRAX+GRAX(I)*DZCELL(I)*DT*AREA(I)
          TOTXGRZ=TOTXGRZ+CBXGRZ(I)*DZCELL(I)*DT*AREA(I)
          TOTXMRT=TOTXMRT+MORTX(I)*DZCELL(I)*DT*AREA(I)
       ENDDO
      RETURN
      END
```

The subroutine for macrophytes (`MACKOL`) includes only two processes: growth (`GROM`) and mortality (`MACM`). `MACOFF` is a logical variable for turning off the macrophyte variable.

```
C*********************************************************************
      SUBROUTINE MACKOL(GROM,MACM,TOTMGROM,TOTMACM,MACOFF)
C*********************************************************************
       INCLUDE 'comp97.inc'
       INCLUDE 'biopar.inc'
       INTEGER I
       REAL GROM(NIM),MACM(NIM),TOTMGROM,TOTMACM
       LOGICAL MACOFF

C-----ASSIGN VALUES TO SI AND SIP
      IF(MACOFF) GOTO 50
      DO I=2,NM1
          SI(I)=GROM(I)
          SIP(I)=-MACM(I)/F(I,JMAC)
C-----VERTICAL INTEGRATION OF PROCESSES
          TOTMGROM=TOTMGROM+GROM(I)*DZCELL(I)*DT*AREA(I)
          TOTMACM=TOTMACM+MACM(I)*DZCELL(I)*DT*AREA(I)
      ENDDO
      RETURN
50    DO I=2,NM1
          SI(I)=0.
          SIP(I)=0.
       ENDDO
       TOTMGROM=0.
       TOTMACM=0.
       RETURN
       END
```

In the phosphate subroutine (`PHOS`) the following processes are included: uptake by growth of phytoplankton, cyanobacteria (both with DIN and N_2) and macrophytes (`GRO`, `GRA`, `GRAX`, `GMDP`), exchange with phosphate in the sediment (`DIFP`) and mineralisation of detritus (`DDC`). Growth of phytoplankton, growth of cyanobacteria and detritus mineralisation are converted to phosphorus rates by the constant (`CPC`). These processes are calculated as organic carbon rates. Exchange of phosphorus with the sediment is calculated per unit bottom area, so the process rate has to be converted to phosphate per volume in this routine. Phosphate is released during denitrification in the water column (`WDEN`) since the process is combined with detritus degradation. Exchange of phosphate with the sediment is added to `SI` or `SIP` depending on direction of the flow.

```
C*********************************************************************
      SUBROUTINE PHOS(GRO,GRA,GRAX,WDEN,DIFP,GMDP,DDC,
     1 TOTDIFFP,TOTPDENI,TOTDDC5,TOTPGRO,TOTPGRA,TOTPGRX,TOTPGROM)
C*********************************************************************
       INCLUDE 'comp97.inc'
       INCLUDE 'biopar.inc'
       INTEGER I
       REAL DDC(NIM),GRAX(NIM)
       REAL GRO(NIM),GRA(NIM),WDEN(NIM),DIFP(NIM),GMDP(NIM)
       REAL TOTDIFFP,TOTPDENI,TOTDDC5,TOTPGRO,TOTPGRA,TOTPGRX
       REAL TOTPGROM

C-----ASSIGN VALUES TO SI AND SIP
      DO I=2,NM1
          SI(I)=DDC(I)*CPC+
     &          WDEN(I)*CCNDENIT*CPC
          SIP(I)=-GRO(I)*CPC/F(I,JPO4)  -GRA(I)*CPC/F(I,JPO4)-
     &          GRAX(I)*CPC/F(I,JPO4)-GMDP(I)/F(I,JPO4)
          IF(DIFP(I).GT.0.)THEN
              SI(I)=SI(I)+DIFP(I)*(AREA(I)-AREA(I-1))/
     &          AREA(I)/DZCELL(I)
          ELSE
              SIP(I)=SIP(I)+DIFP(I)*(AREA(I)-AREA(I-1))/AREA(I)/
     &          DZCELL(I)/F(I,JPO4)
          ENDIF
C-----VERTICAL INTEGRATION OF PROCESSES
          TOTDIFFP=TOTDIFFP+DIFP(I)*(AREA(I)-AREA(I-1))*DT
          TOTPDENI=TOTPDENI+WDEN(I)*CCNDENIT*CPC*DZCELL(I)*
     &          DT*AREA(I)
          TOTDDC5=TOTDDC5+DDC(I)*CPC*DZCELL(I)*DT*AREA(I)
          TOTPGRO=TOTPGRO+GRO(I)*CPC*DZCELL(I)*DT*AREA(I)
          TOTPGRA=TOTPGRA+GRA(I)*CPC*DZCELL(I)*DT*AREA(I)
          TOTPGRX=TOTPGRX+GRAX(I)*CPC*DZCELL(I)*DT*AREA(I)
          TOTPGROM=TOTPGROM+GMDP(I)*DZCELL(I)*DT*AREA(I)
      ENDDO
      RETURN
      END
```

In the oxygen subroutine (OXYGEN) the following processes are included: growth of phytoplankton, cyanobacteria (both with DIN and N_2) and macrophytes (GRO, GRA, GRAX, GROM), nitrification in sediment (NIT) and mineralisation of sediment organic matter and detritus (SMIN, DDC). All process rates are calculated in organic carbon, except nitrification, and have to be converted to corresponding oxygen rate by the constant COC. Nitrification uses a relation between nitrate produced and oxygen consumed (CONNITR). Sediment mineralisation and nitrification both are rates of the bottom area and consequently transformed for inclusion in the oxygen equation. All sinks for oxygen are included in the SI array to cope with negative oxygen concentrations.

```
C*********************************************************************
      SUBROUTINE OXYGEN(GRO,GRA,GRAX,SMIN,NIT,GROM,DDC,
     1 TOTOGRO,TOTOGRA,TOTOGRX,TOTOGROM,TOTOSMIN,TOTDDC6,TOTONIT)
C*********************************************************************
       INCLUDE 'comp97.inc'
       INCLUDE 'biopar.inc'
       INTEGER I
       REAL GRO(NIM),GRA(NIM),GRAX(NIM),SMIN(NIM)
       REAL DDC(NIM),GROM(NIM),NIT(NIM)
       REAL TOTOGRO,TOTOGRA,TOTOGRX,TOTOGROM,TOTOSMIN,TOTDDC6,TOTONIT

C-----ASSIGN VALUES TO SI AND SIP
C     NOTE THAT SINKS ARE INCLUDED IN THE SI-TERM
```

```
C      THIS MAKES IT POSSIBLE FOR JOXY TO BECOME NEGATIVE!
       DO I=2,NM1
           SI(I)=GRO(I)*COC-SMIN(I)*COC*(AREA(I)-AREA(I-1))/
     &       AREA(I)/DZCELL(I)+GRA(I)*COC+GRAX(I)*COC+GROM(I)*COC-
     &       DDC(I)*COC-NIT(I)*CONNITR*(AREA(I)-AREA(I-1))/
     &       AREA(I)/DZCELL(I)
           SIP(I)=0.
C-----VERTICAL INTEGRATION OF PROCESSES
           TOTOGRO=TOTOGRO+GRO(I)*COC*DZCELL(I)*DT*AREA(I)
           TOTOGRA=TOTOGRA+GRA(I)*COC*DZCELL(I)*DT*AREA(I)
           TOTOGRX=TOTOGRX+GRAX(I)*COC*DZCELL(I)*DT*AREA(I)
           TOTOGROM=TOTOGROM+GROM(I)*COC*DZCELL(I)*DT*AREA(I)
           TOTOSMIN=TOTOSMIN+SMIN(I)*COC*(AREA(I)-AREA(I-1))*DT
           TOTDDC6=TOTDDC6+DDC(I)*COC*DZCELL(I)*DT*AREA(I)
           TOTONIT=TOTONIT+NIT(I)*CONNITR*(AREA(I)-AREA(I-1))*DT
       ENDDO
       RETURN
       END
```

In the sediment phosphorus subroutine (BPHOS) the following processes are included: growth of macrophytes (GMBP), exchange of phosphate with the water (DIFP) and mineralisation and denitrification of sediment organic matter (SMIN, SEDN). In addition point sources of phosphorus are added through the parameter POINTBIP (Table 10). This is a common variable declared in biopar.inc. It is given in kg per year and therfore transformed to g m^{-2} here. Only the part corresponding to the bottom area (AREA(I)-AREA(I-1)) is added. The constant CPC converts sediment mineralisation and denitrification to phosphorus. Exchange of phosphate with water is added to SI or SIP depending on the direction of the flow.

```
C*******************************************************************
      SUBROUTINE BPHOS(SMIN,SDEN,DIFP,GMBP,TOTBPMIN,TOTBPDEN,
     1 TOTBPMAC,TOTBPDIF,SAREA)
C*******************************************************************
       INCLUDE 'comp97.inc'
       INCLUDE 'biopar.inc'
       INTEGER I
       REAL SMIN(NIM),SDEN(NIM),DIFP(NIM),GMBP(NIM)
       REAL SAREA
       REAL TOTBPMIN,TOTBPDEN,TOTBPMAC,TOTBPDI

C-----ASSIGN VALUES TO SI AND SIP
       DO I=2,NM1
           SI(I)=SMIN(I)*CPC+
     &         POINTBIP/SAREA/1000./365./86400.*
     &         (AREA(I)-AREA(I-1))/AREA(N)
     &         +SDEN(I)*CPC*CCNDENIT
           SIP(I)=-GMBP(I)/F(I,JBIP)
           IF(DIFP(I).GT.0.)THEN
               SIP(I)=SIP(I)-DIFP(I)/F(I,JBIP)
           ELSE
               SI(I)=SI(I)-DIFP(I)
           ENDIF
C-----VERTICAL INTEGRATION OF PROCESSES
           TOTBPMIN=TOTBPMIN+SMIN(I)*CPC*(AREA(I)-AREA(I-1))*DT
           TOTBPDEN=TOTBPDEN+SDEN(I)*CPC*CCNDENIT*
     &          (AREA(I)-AREA(I-1))*DT
           TOTBPMAC=TOTBPMAC+GMBP(I)*(AREA(I)-AREA(I-1))*DT
           TOTBPDIF=TOTBPDIF+DIFP(I)*(AREA(I)-AREA(I-1))*DT
       ENDDO
       RETURN
       END
```

The sources and sinks of ammonium in the sediment are added in subroutine (BAMMO). The processes influencing ammonium are growth of macrophytes (GMBN), exchange of nitrogen with the water (DIFNH), nitrification (NIT) and mineralisation and denitrification of sediment organic matter (SMIN, SEDN). Mineralisation and denitrification have to be converted to nitrogen rates by constants (CNC, CCNDENIT). Exchange of nitrogen with water is added to SI or SIP depending on the direction of the flow.

```
C*********************************************************************
      SUBROUTINE BAMMO(SMIN,SDEN,NIT,DIFNH,GMBN,
     1 TOTBNMIN,TOTBNDEN,TOTNNIT,TOTBNMAC,TOTBNDIF)
C*********************************************************************
       INCLUDE 'comp97.inc'
       INCLUDE 'biopar.inc'
       INTEGER I
       REAL SMIN(NIM),SDEN(NIM),NIT(NIM),DIFNH(NIM),GMBN(NIM)
       REAL TOTBNMIN,TOTBNDEN,TOTNNIT,TOTBNMAC,TOTBNDIF

C-----ASSIGN VALUES TO SI AND SIP
      DO I=2,NM1
              SI(I)=SMIN(I)*CNC
     &           +SDEN(I)*CNC*CCNDENIT
              SIP(I)=-NIT(I)/F(I,JBNH)-
     &             GMBN(I)/F(I,JBNH)
          IF(DIFNH(I).GT.0.)THEN
              SIP(I)=SIP(I)-DIFNH(I)/F(I,JBNH)
          ELSE
              SI(I)=SI(I)-DIFNH(I)
          ENDIF
C-----VERTICAL INTEGRATION OF PROCESSES
          TOTBNMIN=TOTBNMIN+SMIN(I)*CNC*(AREA(I)-AREA(I-1))*DT
          TOTBNDEN=TOTBNDEN+SDEN(I)*CNC*CCNDENIT*
     &          (AREA(I)-AREA(I-1))*DT
          TOTNNIT=TOTNNIT+NIT(I)*(AREA(I)-AREA(I-1))*DT
          TOTBNMAC=TOTBNMAC+GMBN(I)*(AREA(I)-AREA(I-1))*DT
          TOTBNDIF=TOTBNDIF+DIFNH(I)*(AREA(I)-AREA(I-1))*DT
       ENDDO
       RETURN
       END
```

The sediment nitrate, in subroutine BNITR, is influenced by the following processes; exchange of nitrogen with the water (DIFNO), nitrification (NIT) and denitrification (SEDN). Exchange of nitrogen with water is added to SI or SIP depending on the direction of the flow.

```
C*********************************************************************
      SUBROUTINE BNITR(SDEN,NIT,DIFNO,TOTINIT,TOTBIDEN,TOTBIDIF)
C*********************************************************************
       INCLUDE 'comp97.inc'
       INCLUDE 'biopar.inc'
       INTEGER I
       REAL SDEN(NIM),NIT(NIM),DIFNO(NIM)
       REAL TOTINIT,TOTBIDEN,TOTBIDIF

C-----ASSIGN VALUES TO SI AND SIP
      DO I=2,NM1
          SI(I)=NIT(I)
          SIP(I)=-SDEN(I)/F(I,JBNO)
          IF(DIFNO(I).GT.0.)THEN
              SIP(I)=SIP(I)-DIFNO(I)/F(I,JBNO)
          ELSE
              SI(I)=SI(I)-DIFNO(I)
```

```
          ENDIF
C-----VERTICAL INTEGRATION OF PROCESSES
          TOTINIT=TOTINIT+NIT(I)*(AREA(I)-AREA(I-1))*DT
          TOTBIDEN=TOTBIDEN+SDEN(I)*(AREA(I)-AREA(I-1))*DT
          TOTBIDIF=TOTBIDIF+DIFNO(I)*(AREA(I)-AREA(I-1))*DT
      ENDDO
      RETURN
      END
```

The subroutine for planktivorous fish (FISHA) include two processes: growth of planktivorous fish (predation on zooplankton) (PREZ) and predation by piscivorous fish (PREFI). Only part of the predation (OMEGA) is used for growth.

```
C*********************************************************************
      SUBROUTINE FISHA(PREZ,PREFI,TOTFAGR,TOTFAM)
C*********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL PREZ(NIM),PREFI(NIM),TOTFAGR,TOTFAM

C-----ASSIGN VALUES TO SI AND SIP
      DO I=2,NM1
          SI(I)=OMEGA*PREZ(I)
          SIP(I)=-PREFI(I)/F(I,JFIA)
C-----VERTICAL INTEGRATION OF PROCESSES
          TOTFAGR=TOTFAGR+OMEGA*PREZ(I)*DZCELL(I)*DT*AREA(I)
          TOTFAM=TOTFAM+PREFI(I)*DZCELL(I)*DT*AREA(I)
      ENDDO
      RETURN
      END
```

The subroutine for piscivorous fish (PISCI) include two processes: growth of piscivorous fish (predation on planktivorous fish) (PREFI) and mortality (FMORT). Only part of the predation (OMEGA) is used for growth.

```
C*********************************************************************
      SUBROUTINE PISCI(PREFI,FMORT,TOTFBGR,TOTFBM)
C*********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL PREFI(NIM),FMORT(NIM),TOTFBGR,TOTFBM

C-----ASSIGN VALUES TO SI AND SIP
      DO I=2,NM1
          SI(I)=OMEGA*PREFI(I)
          SIP(I)=-FMORT(I)/F(I,JFIB)
C-----VERTICAL INTEGRATION OF PROCESSES
          TOTFBGR=TOTFBGR+OMEGA*PREFI(I)*DZCELL(I)*DT*AREA(I)
          TOTFBM=TOTFBM+FMORT(I)*DZCELL(I)*DT*AREA(I)
      ENDDO
      RETURN
      END
```

5.4 The BIOPAR subroutine

The `BIOPAR` subroutine reads the model parameters from a file with the number of the unit connected to the file given as argument (`fileunit`). An example of such file is shown in Section 6.7. The parameters not found in the file are given default values in the `BIOPAR` subroutine. The model parameter `ANACOEFF` is calculated after the file is read.

```
C**********************************************************************
      SUBROUTINE BIOPAR(fileunit)
C**********************************************************************
      INTEGER fileunit
      REAL value
      LOGICAL equal
      CHARACTER(LEN=80) line
      CHARACTER(LEN=12) str

      INCLUDE 'biopar.inc'

C     Default values with reference
      P1=1.               !RH16
      P2=0.4              !RH16
      P3=0.2              !RH16
      SINKP=0.8           !RH16
      SINKD=1.2           !RH16
      SINKAN=0.           !RH16
      OMEGA=1.            !RH16, not used there
      ZMRT=0.005          !RH16
      KPKONST=0.3         !RH16
      KW=0.04             !RH16
      KDKONST=0.2         !RH16
      IZKONST=1.0         !RH16, not used there
      IOPTA=50.           !RH16
      MYMAX=2.            !RH16
      MYMAXA=1.           !RH16
      MYMAXB=1.           !RH16, same as MYMAXA
      KMKONST=0.16        !RH16
      KDIN=0.02           !RH16
      KPO4=0.01           !RH16
      KBNO=3.             !RH16
      KMIP=0.005          !RH16
      KMIN=0.01           !RH16
      TMAC=20.            !RH16, same as TPHY
      TETAMAC=1.04        !RH16
      MYMAXM=0.5          !RH16
      IOPTM=75.           !RH16
      MMRT=0.05           !RH16
      XMRT=0.             !RH16
      SNF4=4.             !RH16
      KNFIX=7.2           !RH16
      TSC=0.08            !RH16,depend on DSED2
      TNFIC=21.           !RH16
      TETANFIC=1.1        !RH16
      TETAG=1.07          !RH16, for fishpred, same as TETAPHY
      TREFG=20.           !RH16, for fishpred, same as TPHY
      TPHY=20.            !RH16
      TETAPHY=1.07        !RH16
      TSMIN=20.           !RH16, same as TPHY
      TETASMIN=1.07       !RH16, same as TETAPHY
      TDEN=20.            !RH16, same as TPHY
      TETADEN=1.07        !RH16, same as TETAPHY
      GAMMA=0.8           !RH16
      KG=0.5              !RH16
```

```
 EFFI=0.4          !RH16
 PREDZ=0.1         !RH16
 KZO=1.5           !RH16
 PREDF=0.1         !RH16
 KFIA=0.2          !RH16
 FMRT=0.0008       !RH16
 DDCC=0.02         !RH16
 KDIFF=1.E-9       !RH16
 HAR=0.03          !RH17
 OXYLIMIT=1.       !RH17
 ADSCAP=129.883    !RH17
 ADSCAPAN=51.953   !RH17
 KAD=0.652         !RH17
 DANS=4.           !RH17
 ADEN=0.16         !RH16
 BDEN=0.03         !RH16
 DSED=0.1          !RH16, for denitrification
 MINMAX=0.002      !RH16
 DSED2=0.1         !RH16, for macrophytes and nitrification, same
                          !as DSED
 SEQCON=0.         !RH16
 RESP=1.           !RH17
 NDEPPAR=1500.     !RH16   mg/m2/yr
 PDEPPAR=7.        !RH16
 POINTDIN=0.
 POINTDET=0.
 POINTPO4=0.
 POINTBIP=0.

DO
    READ(fileunit,600,END=300) line
    IF(line(1:1).EQ.'!') CYCLE
    i = 1
    DO
        equal = (line(i: i) .EQ. '=')
        IF(equal) THEN
          EXIT
        ELSE
          i=i+1
        ENDIF
    ENDDO
    str  = line(1:i-1)
    line = line(i+1:80)
    READ(line,*) value

    IF(str.EQ.'P1')         P1=value
    IF(str.EQ.'P2')         P2=value
    IF(str.EQ.'P3')         P3=value
    IF(str.EQ.'SINKP')      SINKP=value
    IF(str.EQ.'SINKD')      SINKD=value
    IF(str.EQ.'SINKAN')     SINKAN=value
    IF(str.EQ.'OMEGA')      OMEGA=value
    IF(str.EQ.'ZMRT')       ZMRT=value
    IF(str.EQ.'KPKONST')    KPKONST=value
    IF(str.EQ.'KW')         KW=value
    IF(str.EQ.'KDKONST')    KDKONST=value
    IF(str.EQ.'IZKONST')    IZKONST=value
    IF(str.EQ.'IOPTA')      IOPTA=value
    IF(str.EQ.'MYMAX')      MYMAX=value
    IF(str.EQ.'MYMAXA')     MYMAXA=value
    IF(str.EQ.'MYMAXB')     MYMAXB=value
    IF(str.EQ.'KMKONST')    KMKONST=value
    IF(str.EQ.'KDIN')       KDIN=value
```

```
         IF(str.EQ.'KPO4')        KPO4=value
         IF(str.EQ.'KBNO')        KBNO=value
         IF(str.EQ.'KMIP')        KMIP=value
         IF(str.EQ.'KMIN')        KMIN=value
         IF(str.EQ.'TMAC')        TMAC=value
         IF(str.EQ.'TETAMAC')     TETAMAC=value
         IF(str.EQ.'MYMAXM')      MYMAXM=value
         IF(str.EQ.'IOPTM')       IOPTM=value
         IF(str.EQ.'MMRT')        MMRT=value
         IF(str.EQ.'XMRT')        XMRT=value
         IF(str.EQ.'SNF4')        SNF4=value
         IF(str.EQ.'KNFIX')       KNFIX=value
         IF(str.EQ.'TSC')         TSC=value
         IF(str.EQ.'TNFIC')       TNFIC=value
         IF(str.EQ.'TETANFIC')    TETANFIC=value
         IF(str.EQ.'TETAG')       TETAG=value
         IF(str.EQ.'TREFG')       TREFG=value
         IF(str.EQ.'TPHY')        TPHY=value
         IF(str.EQ.'TETAPHY')     TETAPHY=value
         IF(str.EQ.'TSMIN')       TSMIN=value
         IF(str.EQ.'TETASMIN')    TETASMIN=value
         IF(str.EQ.'TDEN')        TDEN=value
         IF(str.EQ.'TETADEN')     TETADEN=value
         IF(str.EQ.'GAMMA')       GAMMA=value
         IF(str.EQ.'KG')          KG=value
         IF(str.EQ.'EFFI')        EFFI=value
         IF(str.EQ.'PREDZ')       PREDZ=value
         IF(str.EQ.'KZO')         KZO=value
         IF(str.EQ.'PREDF')       PREDF=value
         IF(str.EQ.'KFIA')        KFIA=value
         IF(str.EQ.'FMRT')        FMRT=value
         IF(str.EQ.'DDCC')        DDCC=value
         IF(str.EQ.'KDIFF')       KDIFF=value
         IF(str.EQ.'HAR')         HAR=value
         IF(str.EQ.'OXYLIMIT')    OXYLIMIT=value
         IF(str.EQ.'ADSCAP')      ADSCAP=value
         IF(str.EQ.'ADSCAPAN')    ADSCAPAN=value
         IF(str.EQ.'KAD')         KAD=value
         IF(str.EQ.'DANS')        DANS=value
         IF(str.EQ.'ADEN')        ADEN=value
         IF(str.EQ.'BDEN')        BDEN=value
         IF(str.EQ.'DSED')        DSED=value
         IF(str.EQ.'MINMAX')      MINMAX=value
         IF(str.EQ.'DSED2')       DSED2=value
         IF(str.EQ.'SEQCON')      SEQCON=value
         IF(str.EQ.'RESP')        RESP=value
         IF(str.EQ.'NDEPPAR')     NDEPPAR=value
         IF(str.EQ.'PDEPPAR')     PDEPPAR=value
         IF(str.EQ.'POINTDIN')    POINTDIN=value
         IF(str.EQ.'POINTDET')    POINTDET=value
         IF(str.EQ.'POINTPO4')    POINTPO4=value
         IF(str.EQ.'POINTBIP')    POINTBIP=value
      ENDDO
300   CONTINUE

      ANACOEFF = ADSCAP / ADSCAPAN

600   FORMAT(A80)
      END
```

5.5 The CHKMIN subroutine

The subroutine CHKMIN is designed to avoid zero concentrations of the state variables and possible problems caused by that. All processes are included in the testing (Table 12) and are arguments of the subroutine CHKMIN. The procedure during checking begins with removing all sinks from the concentration. If the new concentration is less than the minimum value, the sinks are zeroed in turn until the concentration is above the minimum value. This is done for every variable and cell.

```
C********************************************************************
      SUBROUTINE CHKMIN(GRO,ZGRAZE,AGRAZE,DETGRZ,CBXGRZ,DETFEK,PREZ,
     1 PREFI,FMORT,GRA,GRAX,SDEN,WDEN,SMIN,NIT,DIFP,DIFNO,DIFNH,GROM
     1  ,GMBN,GMDN,GMBP,GMDP,MACM,DDC,SINKOD,SINKID,SEDD,SINKOP,
     1 SINKIP,SEDP,SEQ,SINKOX,SINKIX,SEDX,ZMORT,MORTX,ETPROP)
C********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'

       INTEGER I
       REAL TESTMIN
       REAL GRO(NIM),ZGRAZE(NIM),AGRAZE(NIM),DETGRZ(NIM),CBXGRZ(NIM)
       REAL DETFEK(NIM),PREZ(NIM),PREFI(NIM),FMORT(NIM),GRA(NIM)
       REAL GRAX(NIM),ZMORT(NIM),MORTX(NIM)
       REAL SDEN(NIM),WDEN(NIM),SMIN(NIM),NIT(NIM),DIFP(NIM)
       REAL DIFNO(NIM),DIFNH(NIM),GROM(NIM),GMBN(NIM),GMDN(NIM)
       REAL GMBP(NIM),GMDP(NIM),MACM(NIM),DDC(NIM)
       REAL SINKOD(NIM),SINKOP(NIM),SINKIP(NIM),SEDP(NIM)
       REAL SINKID(NIM),SEDD(NIM),SEQ(NIM),SINKOX(NIM),SINKIX(NIM)
       REAL SEDX(NIM),ETPROP(NIM)

C      CHECK EVERY VARIABLE FOR SINKS THAT REDUCES IT BELOW MINIMUM
C      VALUE
      DO I=2,NM1
C
C      ZOOPLANKTON
       TESTMIN=F(I,JZOO)-PREZ(I)*DT-ZMORT(I)*DT
       IF (TESTMIN.LT.ZOOMIN) THEN
         TESTMIN=TESTMIN+PREZ(I)*DT
         PREZ(I)=0.
         IF (TESTMIN.LT.ZOOMIN) THEN
           ZMORT(I)=0.
         ENDIF
       ENDIF
C
C      PHYTOPLANKTON
       TESTMIN=F(I,JPHY)-AGRAZE(I)*DT-SINKOP(I)*DT
       IF (TESTMIN.LT.PHYMIN) THEN
         TESTMIN=TESTMIN+AGRAZE(I)*DT
         ZGRAZE(I)=ZGRAZE(I)-EFFI*AGRAZE(I)
         DETFEK(I)=(1-EFFI)/EFFI*ZGRAZE(I)
         AGRAZE(I)=0.
         IF (TESTMIN.LT.PHYMIN) THEN
           SINKOP(I)=0.
           SINKIP(I-1)=0.
           SEDP(I)=0.
         ENDIF
       ENDIF
C
C      DETRITUS
       TESTMIN=F(I,JDET)-SINKOD(I)*DT-DDC(I)*DT
     & -DETGRZ(I)*DT-WDEN(I)*CCNDENIT*DT
       IF (TESTMIN.LT.DETMIN) THEN
```

```
          TESTMIN=TESTMIN+WDEN(I)*CCNDENIT*DT
          WDEN(I)=0.
          IF (TESTMIN.LT.DETMIN) THEN
            ZGRAZE(I)=ZGRAZE(I)-EFFI*DETGRZ(I)
            DETFEK(I)=(1-EFFI)/EFFI*ZGRAZE(I)
            TESTMIN=TESTMIN+DETGRZ(I)*DT
            DETGRZ(I)=0.
            IF (TESTMIN.LT.DETMIN) THEN
              TESTMIN=TESTMIN+DDC(I)*DT
              DDC(I)=0.
              IF (TESTMIN.LT.DETMIN) THEN
                SINKOD(I)=0.
                SINKID(I-1)=0.
                SEDD(I)=0.
              ENDIF
            ENDIF
          ENDIF
        ENDIF
C
C       SEDIMENT ORGANIC MATTER
        TESTMIN=F(I,JSED)-SDEN(I)*CCNDENIT*DT-SMIN(I)*DT-
     &  SEQ(I)*DT-ETPROP(I)*F(I,JSED)*DT
        IF (TESTMIN.LT.SEDMIN) THEN
          TESTMIN=TESTMIN+SMIN(I)*DT
          SMIN(I)=0.
          IF (TESTMIN.LT.SEDMIN) THEN
            TESTMIN=TESTMIN+ETPROP(I)*F(I,JSED)*DT
            ETPROP(I)=0.
            IF (TESTMIN.LT.SEDMIN) THEN
              TESTMIN=TESTMIN+SEQ(I)*DT
              SEQ(I)=0.
              IF (TESTMIN.LT.SEDMIN) THEN
                SDEN(I)=0.
              ENDIF
            ENDIF
          ENDIF
        ENDIF
C
C       DISSOLVED NITROGEN
        IF((DIFNO(I).LE.0.).AND.(DIFNH(I).LE.0.))THEN
          TESTMIN=F(I,JDIN)-GRO(I)*CNC*DT-WDEN(I)*DT-GRA(I)*CNC*DT-
     & GMDN(I)*DT+(DIFNO(I)+DIFNH(I))*
     & (AREA(I)-AREA(I-1))/AREA(I)/DZCELL(I)*DT
        ELSEIF((DIFNO(I).GT.0.).AND.(DIFNH(I).LT.0.))THEN
          TESTMIN=F(I,JDIN)-GRO(I)*CNC*DT-WDEN(I)*DT-GRA(I)*CNC*DT-
     & GMDN(I)*DT+DIFNH(I)*(AREA(I)-AREA(I-1))/AREA(I)/DZCELL(I)*DT
        ELSEIF((DIFNO(I).LT.0.).AND.(DIFNH(I).GT.0.))THEN
          TESTMIN=F(I,JDIN)-GRO(I)*CNC*DT-WDEN(I)*DT-GRA(I)*CNC*DT-
     & GMDN(I)*DT+DIFNO(I)*(AREA(I)-AREA(I-1))/AREA(I)/DZCELL(I)*DT
        ELSE
          TESTMIN=F(I,JDIN)-GRO(I)*CNC*DT-WDEN(I)*DT-GRA(I)*CNC*DT
     & -GMDN(I)*DT
        ENDIF
        IF (TESTMIN.LT.DINMIN) THEN
          TESTMIN=TESTMIN+WDEN(I)*DT
          WDEN(I)=0.
          IF (TESTMIN.LT.DINMIN) THEN
            IF (GMDN(I).GT.0.0) THEN
              TESTMIN=TESTMIN+GMDN(I)*DT
              GMDN(I)=0.
              GMBN(I)=0.
              GMDP(I)=0.
              GMBP(I)=0.
```

```
          GROM(I)=0.
        ENDIF
        IF (TESTMIN.LT.DINMIN) THEN
          TESTMIN=TESTMIN+GRA(I)*CNC*DT
          GRA(I)=0.
          IF (TESTMIN.LT.DINMIN) THEN
            TESTMIN=TESTMIN+GRO(I)*CNC*DT
            GRO(I)=0.
            IF (TESTMIN.LT.DINMIN) THEN
              DIFNO(I)=0.
              DIFNH(I)=0.
            ENDIF
          ENDIF
        ENDIF
      ENDIF
    ENDIF
C
C     NITROGEN FIXATING ALGAE
      TESTMIN=F(I,JANF)-SINKOX(I)*DT-CBXGRZ(I)*DT-MORTX(I)*DT
      IF (TESTMIN.LT.ANFMIN) THEN
        TESTMIN=TESTMIN+CBXGRZ(I)*DT
        ZGRAZE(I)=ZGRAZE(I)-EFFI*CBXGRZ(I)
        DETFEK(I)=(1-EFFI)/EFFI*ZGRAZE(I)
        CBXGRZ(I)=0.
        IF (TESTMIN.LT.ANFMIN) THEN
          TESTMIN=TESTMIN+MORTX(I)*DT
          MORTX(I)=0.
          IF (TESTMIN.LT.ANFMIN) THEN
            SINKOX(I)=0.
            SINKIX(I-1)=0.
            SEDX(I)=0.
          ENDIF
        ENDIF
      ENDIF
C
C     MACROPHYTES
C     HAVE ONLY ONE SINK, WHICH IS CHECKED WHEN CALCULATED
C
C     PHOSPHATE
      IF(DIFP(I).LT.0.)THEN
        TESTMIN=F(I,JPO4)-GRO(I)*CPC*DT+
     &    DIFP(I)*(AREA(I)-AREA(I-1))/AREA(I)/DZCELL(I)*DT
     &   -GRA(I)*CPC*DT-GRAX(I)*CPC*DT-GMDP(I)*DT
      ELSE
       TESTMIN=F(I,JPO4)-GRO(I)*CPC*DT
     &   -GRA(I)*CPC*DT-GRAX(I)*CPC*DT-GMDP(I)*DT
      ENDIF
      IF (TESTMIN.LT.PO4MIN) THEN
        IF (GMDP(I).GT.0.0) THEN
          TESTMIN=TESTMIN+GMDP(I)*DT
          GMDP(I)=0.
          GMBP(I)=0.
          GMDN(I)=0.
          GMBN(I)=0.
          GROM(I)=0.
        ENDIF
        IF (TESTMIN.LT.PO4MIN) THEN
          TESTMIN=TESTMIN+GRAX(I)*CPC*DT
          GRAX(I)=0.
          IF (TESTMIN.LT.PO4MIN) THEN
            TESTMIN=TESTMIN+GRA(I)*CPC*DT
            GRA(I)=0.
            IF (TESTMIN.LT.PO4MIN) THEN
```

```
              TESTMIN=TESTMIN+GRO(I)*CPC*DT
              GRO(I)=0.
              IF (TESTMIN.LT.PO4MIN) THEN
                DIFP(I)=0.
              ENDIF
            ENDIF
          ENDIF
        ENDIF
      ENDIF
C
C     OXYGEN
      TESTMIN=F(I,JOXY)-DDC(I)*COC*DT
     &   -SMIN(I)*COC*(AREA(I)-AREA(I-1))/AREA(I)/DZCELL(I)*DT-
     &    NIT(I)*CONNITR*(AREA(I)-AREA(I-1))/AREA(I)/DZCELL(I)*DT
      IF (TESTMIN.LT.OXYMIN) THEN
        TESTMIN=TESTMIN+NIT(I)*CONNITR*(AREA(I)-AREA(I-1))
     1  /AREA(I)/DZCELL(I)*DT
        NIT(I)=0.
        IF (TESTMIN.LT.OXYMIN) THEN
          TESTMIN=TESTMIN+SMIN(I)*COC*(AREA(I)-AREA(I-1))
     1    /AREA(I)/DZCELL(I)*DT
          SMIN(I)=0.
          IF (TESTMIN.LT.OXYMIN) THEN
            DDC(I)=0.
          ENDIF
        ENDIF
      ENDIF
C
C     BENTHIC PHOSPHATE
      IF(DIFP(I).GT.0.)THEN
        TESTMIN=F(I,JBIP)-DIFP(I)*DT-GMBP(I)*DT
      ELSE
        TESTMIN=F(I,JBIP)-GMBP(I)*DT
      ENDIF
      IF (TESTMIN.LT.BIPMIN) THEN
        IF (GMBP(I).GT.0.0) THEN
          TESTMIN=TESTMIN+GMBP(I)*DT
          GMBP(I)=0.
          GMDP(I)=0.
          GMBN(I)=0.
          GMDN(I)=0.
          GROM(I)=0.
        ENDIF
        IF (TESTMIN.LT.BIPMIN) THEN
          DIFP(I)=0.
        ENDIF
      ENDIF
C
C     BENTIC AMMONIUM
      IF(DIFNH(I).GT.0.)THEN
        TESTMIN=F(I,JBNH)-NIT(I)*DT-DIFNH(I)*DT-GMBN(I)*DT
      ELSE
        TESTMIN=F(I,JBNH)-NIT(I)*DT-GMBN(I)*DT
      ENDIF
      IF (TESTMIN.LT.BNHMIN) THEN
        IF (GMBN(I).GT.0.0) THEN
          TESTMIN=TESTMIN+GMBN(I)*DT
          GMBN(I)=0.
          GMDN(I)=0.
          GMBP(I)=0.
          GMDP(I)=0.
          GROM(I)=0.
```

```
          ENDIF
          IF (TESTMIN.LT.BNHMIN) THEN
            TESTMIN=TESTMIN+NIT(I)*DT
            NIT(I)=0.
            IF (TESTMIN.LT.BNHMIN) THEN
              DIFNH(I)=0.
            ENDIF
          ENDIF
        ENDIF
C
C       SEDIMENT NITRATE
        IF(DIFNO(I).GT.0.) THEN
          TESTMIN=F(I,JBNO)-SDEN(I)*DT-DIFNO(I)*DT
        ELSE
          TESTMIN=F(I,JBNO)-SDEN(I)*DT
        ENDIF
        IF (TESTMIN.LT.BNOMIN) THEN
          TESTMIN=TESTMIN+SDEN(I)*DT
          SDEN(I)=0.
          IF (TESTMIN.LT.BNOMIN) THEN
            DIFNO(I)=0.
          ENDIF
        ENDIF

      ENDDO
      RETURN
      END
```

5.6 The OXYFLX subroutine

The `OXYFLX` subroutine calculates the exchange of oxygen between the water and the air (`FOT`, mg O_2 L^{-1} s^{-1} upward). The subroutine is adapted from the coastal zone model SCOBI (Marmefelt et al., 2000; Stigebrandt, 1991). The `OXYFLX` subroutine has been adjusted, because SCOBI calculates oxygen concentration in mL L^{-1}, while BIOLA calculates oxygen concentration in mg L^{-1}. The exchange depends on wind (subroutine argument), and on temperature and oxygen concentration in the lake.

```
C*********************************************************************
      SUBROUTINE OXYFLX(WIND,FOT)
C*********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'BIOPAR.INC'

      REAL WIND,FOT,TEMPC,TEMPK,O2,A,B,O2S,SC,V
      DATA BUBBEL/0.025/

      TEMPC=F(NM1,JTE)
      TEMPK=F(NM1,JTE)+273.15

C     TRANSFORM FROM MG/L TO ML/L.
      O2=F(NM1,JOXY)*0.7

      IF (WIND.LE.3.6) THEN
        A=0.17
        B=0.
      ELSEIF (WIND.GT.3.6.AND.WIND.LE.13.) THEN
        A=2.85
        B=-9.65
      ELSE
```

```
      A=5.9
      B=-49.3
     ENDIF

     O2S=EXP(-173.4292+249.6339*100./TEMPK+
    &143.3483*LOG(TEMPK/100.)-21.8492*TEMPK/100.)
     SC=1450.+(-71.+1.1*TEMPC)*TEMPC
     V=5.9*(A*WIND+B)/SQRT(SC)/86400.
     FOT=V*(O2-O2S*(1.+BUBBEL))

C    FOT IS IN ML/L o S. TRANSFORM TO MG/L o S
     FOT=FOT/0.700

     RETURN
     END
```

5.7 The ETADEPTH subroutine

The ETADEPTH subroutine calculates the z-coordinate of the deepest erosion-transport bottom. It has the input arguments AREA, Z, DZCELL, NIM and N, and the output is ETD. ETD is the cell mid point. More details about this subroutine and its equation can be found in Ahlkrona (2002).

```
C*******************************************************************
      SUBROUTINE ETADEPTH(AREA,Z,DZCELL,NIM,N,ETD)
C*******************************************************************
      INTEGER NIM, N, DINDEX
      REAL AREA(NIM),Z(NIM),DZCELL(NIM)
      REAL VOL,MEANDEPTH,PART,A,ETD

      VOL = 0.0
      DO I = 2, N-1
         VOL = VOL + AREA(I) * DZCELL(I)
      ENDDO
      MEANDEPTH = VOL / AREA(N-1)
      PART = 25.0*SQRT(AREA(N-1)/1.0E6)/MEANDEPTH*41.0**(0.061*
     1         MEANDEPTH/SQRT(AREA(N-1)/1.0E6))
      A = AREA(N-1)*(100.0-PART)/100.0
      DO I = 2, N-1
         IF (AREA(I).LT.A) THEN
         ELSE
            DINDEX = I
            EXIT
         ENDIF
      ENDDO
      ETD = Z(DINDEX)
      RETURN
      END
```

6 Detailed description of how to use BIOLA

6.1 Use the ice subroutine

When applying the BIOLA model to a lake in Sweden the lake may be ice covered during winter. This is important for the biogeochemical model as the ice decreases the penetrating light and may delay the spring phytoplankton growth.

Ice influences the upper boundary. The ice calculation is therefore put in Chapter 2 (Boundary Conditions) of CASE. In the example below, a logical variable `ICEFOR` is used to signify an ice-covered lake. Ice is assumed to begin forming when the mean temperature of the upper boundary and the uppermost cell is below zero (`TICE`). If the lake is ice covered, new upper boundaries are set; the temperature of the upper boundary is set to zero (`IKBHZ(JH)=1` means a value is used for boundary condition), a small momentum is assumed for one direction, while the other is zero, the Stanton number for heat is assumed to be hundred times lower than for an open lake. The ice thickness is calculated with the subroutine `ICE` and the daily mean variables are set to zero for next day. Thereafter ice break-up is tested. If the thickness of ice is zero, ice is assumed gone (`ICEFOR=.FALSE.`) and the Stanton number is changed back to default value. Also the boundary condition of heat is returned to be a flux (`IKBHZ(JH)=2`), the upper boundary and the four uppermost cells are given the starting value of kinetic energy and dissipation (k, ε), and the accumulation variables of temperature `SUMTAM` is set to zero.

Below is an example of ice calculation in Chapter 2 of CASE. Note that the new variables need to be declared in CASE.

```
      LOGICAL ICEFOR
      INTEGER NR
      REAL HICE,FTICE,SUMTAM,WGM,TAM,TICE
      ..
C     CHAPTER 2 TRANSIENT BOUNDARY CONDITIONS
      ..
C---- FOR ICE THICKNESS CALCULATION
      TAM=TAM+TA
      WGM=WGM+WG
      FTICE=FTICE+FN-FSB+FST
      NR=NR+1

C  ******************************************************************
C   ICE  FORMATION / DESTRUCTION  TEST
C  ******************************************************************

      TICE=(F(N,JTE)+F(NM1,JTE))/2.
      IF(TICE.LE.0.)ICEFOR=.TRUE.
      IF(ICEFOR)THEN

C---- NEW BOUNDARY CONDITIONS ICE COVERED LAKE
         IKBHZ(JH)=1
         STANTN(JH)=5.E-4
         F(N,JH)=0.
         FLUXHZ(JRHOU)=-0.003
         FLUXHZ(JRHOV)=0.0

C---- DAILY ICE THICKNESS CALCULATION
         IF(MOD(ISTEP,TSDAY).EQ.0)THEN
            TAM=TAM/REAL(NR)
```

```
              WGM=WGM/REAL(NR)
              FTICE=FTICE/REAL(NR)
              CALL ICE(TAM,WGM,SUMTAM,FTICE,HICE)
              TAM=0.0
              WGM=0.0
              FTICE=0.0
              NR=0

C---   ICE MELTING/DESTRUCTION TEST
              IF(HICE.LE.0.)THEN
                 DO I=N-4,N
                    F(I,JK)=1.E-5
                    F(I,JD)=5.E-8
                 ENDDO

C---- NEW BOUNDARY CONDITIONS OPEN LAKE
                 IKBHZ(JH)=2
                 STANTN(JH)=5.E-2
                 ICEFOR=.FALSE.
                 SUMTAM=0.
                 FTICE=0.
              ENDIF
          ENDIF
      ENDIF

C *** END  OF  ICE  TEST  *******************************************
```

The ice subroutine calculates ice thickness (`HICE`) in meters from air temperature (`TAM`), wind (`WGM`) and net heat flux at the upper ice/snow surface (`FTICE`) (Sahlberg, 1988). The subroutine is called once a day and the daily mean temperature, wind and heat flux are used. These are in this example calculated from variables (`TA`, `WG`, `FN`, `FST`, `FSB`) calculated by the subroutine PROBEFLX (see Section 6.2). The formulation of the ice calculation is from Ashton (1983).

6.2 Use PROBEFLX for meteorological forcing

The `PROBEFLX` subroutine uses synop station observations of air temperature, wind, relative humidity and cloud coverage to calculate the upper boundary conditions of momentum and heat and the heat source of solar radiation penetrating into the water (Omstedt, 1990; Rutgersson et al., 2001). BIOLA has used the solar radiation of PROBEFLX for calculating algal growth.

PROBEFLX is called in Chapter 2 (Boundary Conditions) of CASE. In this example 3-hour interval observations are read from a file that has been opened in Chapter 1 with fileunit `LUNIT`. The `PROBEFLX` subroutine has a lot of input arguments besides the meteorological data (time (`IYMDH`), air temperature (`TA`), wind in two directions (`UG`,`VG`), relative humidity (`RH`), and cloud coverage (`CL`)). The subroutine needs latitude (`FLAT`) for calculating the suns rise, fall and zenith angle, the temperature of the surface water for heat exchange, time between observations (`DTSOL`; in seconds) the thickness of snow (`HSNOW`) and ice (`HICE`) for reducing the light penetration and changing the surface albedo, and the surface area (`AREA(N)`). For a sheltered lake the variables `IHS`, `ILS`, `IBRE`, `SHELTER`, `STRERED` are used to reduce the solar radiation and stress on the lake. `SHELTER` and `STRERED` are logical variables indicating if the surroundings shelters the sun or reduces the stress on the lake. `IHS` is the height of the surroundings, `ILS` the distance from the lake, and `IBRE` the horizontal extension. All three are arrays of eight elements denoting eight sectors around the lake. Output from the `PROBEFLX` subroutine is wind speed (`WG`), stress on the lake surface in x-and y-directions (STREX, STREY), sensible heat flux (FH), latent heat flux (FE), net long-wave radiation (FL), total heat flux (FN), solar (short-wave) radiation penetrating the upper ice/snow surface into the ice (FST), and solar radiation penetrating the ice into the water (FSB). Stress and heat flux are applied to the upper boundary variable FLUXHZ, while solar radiation is put in the variable FLXRAD. If FLXRAD is used for light in BIOLA, BIOLA's parameter IZKONST should be equal to the PROBE variable RADFRA. Note that the new variables need to be declared in CASE. This is not shown in the example below.

```
C       CHAPTER 1  MODIFY DEFAULT DATA
C       GROUP 7
        ..
        OPEN(UNIT=10,FILE='weather.dat',form='formatted',STATUS='old')
        LUNIT=10
        DTSOL=60.*60.*3.
        SHELTER=.FALSE.
        STRERED=.FALSE.
        ..

C       CHAPTER 2 TRANSIENT BOUNDARY CONDITIONS
        ..
C---- READ WEATHERDATA AND COMPUTE SURFACE FLUXES
        IF(MOD(ISTEP,TSDAY/8).EQ.0)THEN
           READ(LUNIT,607)IYMDH,TA,UG,VG,CL,RH
607        FORMAT(I12,5F10.3)
           CALL PROBEFLX(FLAT,DTSOL,F(NM1,JTE),HSNOW,HICE,
     1              AREA(N),IHS,ILS,IBRE,SHELTER,STRERED,IYMDH,TA,
     2              UG,VG,WG,CL,RH,STREX,STREY,FH,FE,FL,FN,FST,FSB)
C
           FLUXHZ(JRHOU)=-STREX
           FLUXHZ(JRHOV)=-STREY
           FLUXHZ(JH)=FN
           FLXRAD=FSB
        ENDIF
```

6.3 Use eddy lake

Sometimes the mixing of deep water in PROBE is too small. Therefore a subroutine to include deep water mixing has been written by Jörgen Sahlberg, SMHI. The subroutine is called EDDY_LAKE. The subroutine calculates the turbulent eddy viscosity (XMIX) to be used instead of the value calculated by the k-ε model (Hondzo and Stafan, 1993). Input to the subroutine is the cell number (I). Below, the subroutine is applied for days with ice covered lake. The subroutine is called in Chapter 2 (Boundary Conditions) of CASE. The variable DEEPMIX denotes if eddy lake will be applied or not. ITURBM controls the turbulence model to be used (ITURBM=4 means eddy viscosity (F(I,JEMU) is set in CASE), ITURBM=3 means use of k-ε model with buoyancy effects (Svensson, 1998), while IPRSC controls the Prandtl/Schmidt number (IPRSC=1 means constant value, IPRSC=2 means the value is affected by buoyancy). The new variables need to be declared in CASE.

```
C       CHAPTER 1  MODIFY DEFAULT DATA
C       GROUP 6
        DEEPMIX=.TRUE.
        IF(DEEPMIX)THEN
           ITURBM=4
           IPRSC=1
        ENDIF
        ..
```

```
C       CHAPTER 2 TRANSIENT BOUNDARY CONDITIONS
        ..
        IF(ICEFOR)THEN
           DEEPMIX=.TRUE.
           ..
           IF(HICE.LE.0.)THEN
              DEEPMIX=.FALSE.
              ..

C       SPECIAL BOTTOM MIXING
        IF(DEEPMIX)THEN
           ITURBM=4
           IPRSC=1
           DO I=2,NM1
              CALL EDDY_LAKE(I,XMIX)
              F(I,JEMU)=XMIX
           ENDDO
       ELSE
           ITURBM=3
           IPRSC=2
       ENDIF
```

6.4 Add sources at the upper boundary

Sources can be added to the model through the upper boundary condition, directly to the source term of the transport equation or through the inflow. The application of the first method is described here, while the last two methods are described in Section 6.5 and 6.6.

If the upper boundary condition is a flux, it is given by the PROBE variable FLUXHZ. The variable has the unit of the corresponding variable times meter per second, i.e. for DIN (g m^{-3}) the unit is g N m^{-2} s^{-1}. The boundary conditions are set in CASE Chapter 2. Note that the flux is positive upward. Below is an example with momentum and heat boundary conditions (STREX, STREY, FN) calculated by PROBEFLX (Section 6.2). The subroutine OXYFLX (Section 5.6) is used to calculate the exchange of oxygen with the air (FOXY), and the resulting flow set as a upper boundary of the oxygen variable (index JOXY). For BIOLA there are a couple of upper boundary fluxes that may be read from the BIOPAR file (Section 5.4). To use them the upper boundary must be set as in the below example, if atmospheric deposition (NDEPPAR and PDEPPAR) is given in kg km^{-2} yr^{-1}, and point sources (POINTDIN, POINTPO4, POINTDET) are given in kg yr^{-1}. Additional sources may be added directly in the code. The atmospheric deposition and point sources variables have been created to make it possible to change them without recompiling the code.

```
C       CHAPTER 2 TRANSIENT BOUNDARY CONDITIONS
        ..
        FLUXHZ(JRHOU)=-STREX
        FLUXHZ(JRHOV)=-STREY
        FLUXHZ(JH)=FH

        CALL OXYFLX(WG,FOXY)
        FLUXHZ(JOXY)=FOXY
        FLUXHZ(JDIN)=-NDEPPAR/1000./365./86400.
      1              -POINTDIN/SAREAKM2/1000./365./86400.
        FLUXHZ(JPO4)=-PDEPPAR/1000./365./86400.
      1              -POINTPO4/SAREAKM2/1000./365./86400.
        FLUXHZ(JDET)=-POINTDET/SAREAKM2/1000./365./86400.
```

6.5 Add sources through the source term

Sources can be added directly to the source term of the transport equation like the way BIOLA does with process flows. This is done by adding the source to `SI` in Chapter 3 of CASE. The unit of `SI` is the unit of the variable per second. Note that `SI` may have a value already that you do not want to lose. A point source of particulate phosphorus that is assumed to settle on the bottom of the lake is set through the BIOPAR parameter `POINTBIP` (kg yr^{-1}). Below a sewage treatment plant are assumed to have a outlet at the bottom of the lake (`I=2`) that causes an increase of $5*10^{-5}$ mg N L^{-1} s^{-1} in the volume of that cell.

```
C       CHAPTER 3   ADDITIONAL SOURCE TERMS
        ..
        IF(J.EQ.JDIN)THEN
            SI(2)=SI(2)+5.E-5
        ENDIF
        ..
```

6.6 Add substances through inflow

A source that may be significant is the substances that arrive with inflow to the lake. If inflow to the lake is used (`QINFL` and `QOUTFL`) they need a corresponding concentration if they are not to dilute the substances in the lake. The concentration of inflow is given by the PROBE variable `PHIIN`. It has the same unit as the variable itself. Note that once given, a `QINFL`, `QOUTFL` or `PHIIN` value will stay until it is zeroed. This is important to remember especially if the inflow point moves between the cells.

In the example below, daily water discharge, temperature of incoming water, DIN and phosphate concentrations is read from file. `PHIIN` of heat is calculated from the temperature (`TEMP`) and specific heat capacity (`CPHEAT`) and density of water (a reference density is used `RHOREF`). `CPHEAT` and `RHOREF` are PROBE variables.

```
C       GROUP 7
        OPEN(UNIT=8,FILE='FLOW.DAT',FORM='FORMATTED',STATUS='OLD')
        ..

C       CHAPTER 2   TRANSIENT BOUNDARY CONDITIONS
C----   IN AND OUTFLOWS
        IF(MOD(ISTEP,TSDAY).EQ.0) THEN
            READ(8,*) IYMD,QREC,TEMP,DIN,PO4
            QINFL(NM1)=QREC
            QOUTFL(NM1)=QREC
            PHIIN(NM1,JH)=TEMP*CPHEAT*RHOREF
            PHIIN(NM1,JDIN)=DIN
            PHIIN(NM1,JPO4)=PO4
        ENDIF
        ..
```

6.7 Change parameter values

All model parameters and the sources atmospheric deposition and point sources to the lake surface will be changed through the file (here called biopar.par) that is read by the `BIOPAR` subroutine. The parameters not found in the file are given default values in the `BIOPAR` subroutine. When parameters are changed in biopar.par the model need to be run to see the effect of the new parameters.

This is an example of a biopar.par file.

```
P2=0.
SINKP=0.16
SINKD=0.5
OMEGA=0.1
ZMRT=0.1
KPO4=0.04
XMRT=0.01
TREFG=15.
TPHY=15.
TSMIN=16.
TETASMIN=1.1
TDEN=15.
TETADEN=1.5
PREDZ=0.05
PREDF=0.2
FMRT=0.0005
DDCC=0.05
KDIFF=1.E-8
ADEN=0.12
MINMAX=0.01
SEQCON=0.0000001
NDEPPAR=1500.
PDEPPAR=7.5
```

6.8 Save results

Outputs from the model are decided in Chapter 4 of CASE. In this section some examples of common output for BIOLA will be given. Other possibilities exist by modifying the code further. The files to be written are most often opened in Chapter 1. It is possible to open result files in Chapter 4, if they are put in an IF statement so that they are opened only once.

```
C       CHAPTER 1  MODIFY DEFAULT DATA
C       GROUP 11
        OPEN(UNIT=15,FILE='RESULTF.DAT',STATUS='UNKNOWN')
        OPEN(UNIT=16,FILE='PHYSPRO.DAT',STATUS='UNKNOWN')
        OPEN(UNIT=40,FILE='FLOWZ.DAT',STATUS='UNKNOWN')
        ..
```

or

```
C       CHAPTER 4  ADDITIONAL OUTPUT
        IF(ISTEP.EQ.0)THEN
            OPEN(UNIT=15,FILE='RESULTF.DAT',STATUS='UNKNOWN')
            OPEN(UNIT=16,FILE='PHYSPRO.DAT',STATUS='UNKNOWN')
            OPEN(UNIT=40,FILE='FLOWZ.DAT',STATUS='UNKNOWN')
            RETURN
        ENDIF
```

Chapter 4 is called once before the simulation starts, and then once in the end of each time step. If results are wanted at another time interval then the time step of the model an IF statement is needed. In the example below TSDAY is the number of time steps per day, and ice thickness and surface temperature are written daily.

```
C       CHAPTER 4  ADDITIONAL OUTPUT
        ..
        IF(MOD(ISTEP,TSDAY).EQ.0)THEN
            WRITE(15,*)IYMD,HICE,F(NM1,JTE)
        ENDIF
        ..
```

With a DO-loop the whole profile of e.g. temperature (`F(I,JTE)`) can be written with corresponding cell level (`Z(I)`). Note that the cells are numbered 2 to NM1 from bottom up.

```
DO I=NM1,2,-1
    WRITE(16,*)IYMD,Z(I),F(I,JTE)
ENDDO
..
```

Daily mean values may be calculated. Here is an example of daily mean DIN (`DDIN(I)`) in lake. The variable will be declared in CASE (`DDIN(NIM)`).

```
DO I=NM1,2,-1
    DDIN(I)=DDIN(I)+F(I,JDIN)
ENDDO
IF(MOD(ISTEP,TSDAY).EQ.0)THEN
    DO I=NM1,2,-1
        DDIN(I)=DDIN(I)/REAL(TSDAY)
        WRITE(16,*)IYMD,Z(I),DDIN(I)
        DDIN(I)=0.0
    ENDDO
ENDIF
..
```

The mean of the whole lake may also be interesting. Here is an example of daily mean DIN (`DWDIN`) in the whole lake. The variables will be declared in CASE (`VOLUME`, `DWDIN`).

```
IF(ISTEP.EQ.0)THEN
    VOLUME=0.0
    DO I=NM1,2,-1
        VOLUME=VOLUME+AREA(I)*DZCELL(I)
    ENDDO
    RETURN
ENDIF
DO I=NM1,2,-1
    DWDIN=DWDIN+F(I,JDIN)*AREA(I)*DZCELL(I)
ENDDO
IF(MOD(ISTEP,TSDAY).EQ.0)THEN
    DWDIN=DWDIN/REAL(TSDAY)/VOLUME
    WRITE(16,*)IYMD,DWDIN
    DWDIN=0.0
ENDIF
..
```

The accumulated process flow variables (Table 11) are designed to be saved in files. They are studied to understand why the model react as it does and help with choosing parameter values. The example is for daily accumulated zooplankton process flows; grazing (`TOTGRZ`), predation by fish (`TOTPRED`) and mortality (`TOTZMRT`). After the flows are written to file they are initialised for accumulating the process flows until next day.

```
IF(MOD(ISTEP,TSDAY).EQ.0)THEN
    WRITE(40,*)IYMD,TOTGRZ,TOTPRED,TOTZMRT
    TOTGRZ=0.0
    TOTPRED=0.0
    TOTZMRT=0.0
ENDIF
```

6.9 User defined cell grid and hypsograph

The cell grid is controlled by the PROBE index `IGRID`. The variable defines the kind of grid to be used in the model (see PROBE Manual). Default is `IGRID=1`, a uniform grid. By setting IGRID=4 in CASE Chapter 1, Group 2 the model will use a grid defined in CASE. The grid, i.e. the cell thickness DZCELL, will be defined (in meters) for each cell I between 2 and N-1. Note that that the PROBE variable NM1 is not defined in Chapter 1 of CASE. In the example is a grid with twelve 1 meter thick cells in the bottom and twelve 0.5 m thick cells on top defined (N=26).

```
C       CHAPTER 1  MODIFY DEFAULT DATA
C       GROUP 2
        IGRID=4
        DO I=2,13
           DZCELL(I)=1.0
           DZCELL(I+12)=0.5
        ENDDO
        ..
```

The user may define the lake's hypsograph by setting the PROBE index INDARE. The variable defines the kind of area distribution to be used in the model (see PROBE Manual). By setting INDARE=4 in CASE Chapter 1, Group 2 the model will use a hypsograph defined in CASE. The hypsograph, AREA, will give the horizontal area of each cell I in CASE Chapter 4 before the simulation starts, i.e. at ISTEP=0. In the example below, linear segments define the hypsograph. Note that the area is given in square meter.

```
C       CHAPTER 1  MODIFY DEFAULT DATA
C       GROUP 2
        INDARE=4
        ..

C       CHAPTER 4  ADDITIONAL OUTPUT
C---- SPECIFY LAKE AREA IN M2
        IF(ISTEP.EQ.0)THEN
           DO I=2,N-1
              IF(Z(I).LE.2.5)THEN
                 AREA(I)=0.0136*Z(I)
              ELSEIF(Z(I).LE.7.5)THEN
                 AREA(I)=0.0276*Z(I)-0.035
              ELSE
                 AREA(I)=0.031*Z(I)-0.0605
              ENDIF
              AREA(I)=AREA(I)*1.E6
           ENDDO
           AREA(N)=AREA(NM1)
           AREA(1)=0.0
           RETURN
       ENDIF
```

6.10 Moving surface

It is possible to have a moving lake surface in PROBE. To use a moving surface the PROBE variable MOVE is set true. The uppermost cell will then vary in thickness, and it is also possible that the number of cells vary during the simulation. The variables N and ZDIM, given in CASE Chapter 1, will denote the maximum number of grid points and the maximum vertical extent of the lake. Another variable ZSSTRT will give the lake extent at the start of simulation. PROBE will then calculate how many cells ZSSTRT corresponds to, and set the current N.

```
C       GROUP 13
        MOVE=.TRUE.
        ZSSTRT=16.
```

In- and outflows that not are in balance cause a moving lake surface and varying volume. In addition precipitation and evaporation may contribute. In- and outflows cause advective flows between the cells to preserve the volume of each cell. Precipitation minus evaporation is added to the uppermost cell of the lake. It is given by the PROBE variable PREEVA. If it is constant it is set in Chapter 1 Group 13 with the moving surface, else it is calculated in Chapter 2 among the transient boundary conditions. PREEVA is given in unit m s^{-1} and positive in the z-direction (upward). In the example below precipitation is read from file.

```
C      CHAPTER 2  TRANSIENT BOUNDARY CONDITIONS
C---- IN AND OUTFLOWS, AND PRECIPITATION
       DO I=1,N
          QINFL(I)=0.0
          QOUTFL(I)=0.0
          PHIIN(I,JH)=0.0
       ENDDO
       IF(MOD(ISTEP,TSDAY).EQ.0)
     &    READ(8,*) IYMD,QINREC,QOUTREC,PREC
       QINFL(NM1)=QINREC
       QOUTFL(NM1)=QOUTREC
       PHIIN(NM1,JH)=F(NM1,JH)
       PREEVA=-PREC
```

6.11 Use several subbasins

A lake model can be set up with several "PROBEs". Each PROBE may simulate a subbasin of the lake, or different PROBEs may simulate the water and the sediments. The number of PROBEs used is set by the PROBE variable NPROBE in CASE Chapter 1, Group 12. PROBE will then loop through the PROBEs every time step. When using NPROBE>1 the CASE file has to be expanded so that a full CASE with all chapters is provided for every IPROBE (the current PROBE). This is done with IF-statements based on IPROBE. Here is an example of the structure of CASE with NPROBE=3. More on coupled PROBEs see the PROBE Manual and the CASE reports.

```
       SUBROUTINE CASE(ICHAP)
       IF(IPROBE.GT.1) GOTO 200
C      --- PROBE 1
       IF(ICHAP.GT.1) GOTO 120
C      CHAPTER 1  MODIFY DEFAULT DATA
       ..
C      GROUP 12
       NPROBE=3
       ..
       RETURN
120    CONTINUE
C      CHAPTER 2  TRANSIENT BOUNDARY CONDITIONS
       IF(ICHAP.GT.2)GOTO 130
       RETURN
130    CONTINUE
C      CHAPTER 3  ADDITIONAL SOURCE TERMS
       IF(ICHAP.GT.3)GOTO 140
       RETURN
140    CONTINUE
C      CHAPTER 4  ADDITIONAL OUTPUT
       RETURN
200    IF(IPROBE.GT.2) GOTO 300
C      --- PROBE 2
       IF(ICHAP.GT.1) GOTO 220
C      CHAPTER 1  MODIFY DEFAULT DATA
```

```
      RETURN
220   CONTINUE
C     CHAPTER 2  TRANSIENT BOUNDARY CONDITIONS
      IF(ICHAP.GT.2)GOTO 230
      RETURN
230   CONTINUE
C     CHAPTER 3  ADDITIONAL SOURCE TERMS
      IF(ICHAP.GT.3)GOTO 240
      RETURN
240   CONTINUE
C     CHAPTER 4  ADDITIONAL OUTPUT
      RETURN
300   IF(IPROBE.GT.3) RETURN
C     --- PROBE 3
      IF(ICHAP.GT.1) GOTO 320
C     CHAPTER 1  MODIFY DEFAULT DATA
      RETURN
320   CONTINUE
C     CHAPTER 2  TRANSIENT BOUNDARY CONDITIONS
      IF(ICHAP.GT.2)GOTO 330
      RETURN
330   CONTINUE
C     CHAPTER 3  ADDITIONAL SOURCE TERMS
      IF(ICHAP.GT.3)GOTO 340
      RETURN
340   CONTINUE
C     CHAPTER 4  ADDITIONAL OUTPUT
      RETURN
      END
```

BIOLA has been simulated with PROBEs for different subbasins. For a lake with several subbasins, the linking between the PROBEs are done with the inflow and outflow variables. It is important to remember that e.g. `QOUTFL(I)` or `F(I,J)` from one PROBE is not available in the next PROBEs CASE. Local variables need to be declared in the `CASE` subroutine for keeping such information that is needed for coupling the subbasins.

```
      SUBROUTINE CASE(ICHAP)
      REAL FLOW1TO2,HEAT1TO2,FLOW2TO3,HEAT2TO3
      SAVE FLOW1TO2,HEAT1TO2,FLOW2TO3,HEAT2TO3
      ..
C     --- PROBE 1
C     CHAPTER 2  TRANSIENT BOUNDARY CONDITIONS
      FLOW1TO2=QOUTFL(NM1)
      HEAT1TO2=F(NM1,JH)
      ..
C     --- PROBE 2
C     CHAPTER 2  TRANSIENT BOUNDARY CONDITIONS
      QINFL(NM1)=FLOW1TO2
      PHIIN(NM1,JH)=HEAT1TO2
      ..
      FLOW2TO3=QOUTFL(NM1)
      HEAT2TO3=F(NM1,JH)
      ..
C     --- PROBE 3
C     CHAPTER 2  TRANSIENT BOUNDARY CONDITIONS
      QINFL(2)=FLOW2TO3
      PHIIN(2,JH)=HEAT2TO3
      ..
```

The calculation of flows between the subbasins can be done with required complexity, e.g. read from file, depending on water levels, in different directions at different levels. It is important to keep in mind that the cells are numbered from below, and that cell 5 in one PROBE probably do not correspond to the level of cell 5 in another PROBE.

When using BIOLA with several subbasins most local variables in CASE need to be declared as arrays. For instance, if the total process flows are needed for each subbasin they need to be declared as arrays. The correct element of the local array is then used in respective PROBE. PROBE variables, e.g. nutrient concentrations of outflow, that need to be known in other PROBEs, need to be saved in new local variables in CASE. The parameter values, as well as atmospheric deposition and point sources are the same in all PROBEs. Different parameter values for different PROBEs have been used, but it requires some changes in the code. If different point sources are needed these should not be applied through the `POINTDIN/PO4/DET/BIP` parameters (see Section 6.4 and 6.5 for alternatives). The output for each subbasin is written to separate files. If they need to be on the same file, special local variables must be used to save the output until all PROBEs has been calculated. As an example, the simple CASE in Section 4.4 will be applied to a three-subbasin lake.

In the example, the `ETD` and `SAREAKM2` variables will be different for different PROBEs. The total process flows will not be separated for the different subbasins. They will sum the total daily flow in the whole lake. They are therefore written to file in the last subbasin and reinitialised there. The upper boundary variables need to be separated even if the same meteorological forcing is used for the whole lake, if they depend on the temperature of the lake for instance. The number of result file will have to be multiplied to give results from each subbasin. Most of the transient boundary conditions can be the same for all subbasins. If the meteo-file is read in the first subbasin and the same formulas are used for momentum and heat for all subbasins, the setting of `FLUXHZ` and `FLXRAD` need only be repeated in the following subbasins. The local variables will keep their values. The oxygen exchange with atmosphere depends on the subbasins' properties, so `OXYFLX` needs to be called in every subbasin. The same parameter values will be used in all subbasins. The coupling between basins will be the most simple one here. The same flow is assumed to flow out of each subbasin as flows into the same subbasin. The three subbasins are assumed to lie in a row feeding each other. All flows are assumed to take place in the uppermost layer. This coupling requires that the concentration of all variables that will follow the flow to next subbasin is saved locally in CASE. This is done with the arrays `CONC1TO2` and `CONC2TO3`. In Chapter 3 of CASE, BIOLA is called with the correct element of the local variables. The output for each subbasin is written to separate files.

```
      SUBROUTINE CASE(ICHAP)

      INCLUDE 'COMP97.INC'
      INCLUDE 'BIOPAR.INC'

      INTEGER ITIMESTEP,IYMD,TSDAY,I
      SAVE    ITIMESTEP,IYMD,TSDAY
      REAL    UG,VG,FH,FSB,QROUT
      SAVE    UG,VG,FH,FSB,QROUT

      LOGICAL MACOFF
      SAVE    MACOFF
      REAL ETD(3),SAREAKM2(3)
      SAVE ETD,SAREAKM2
      REAL TOTGRZ,TOTPRED,TOTASINKI,TOTASINKO,
     1 TOTGRW,TOTAGRZ,TOTDSINKI,TOTDSINKO,TOTDECOM,
     1 TOTDFEK,TOTDFI,TOTDDEN,TOTDPRED,
     1 TOTSSINKP,TOTSSINKD,TOTSSINKN,TOTSMACM,
     1 TOTSDENI,TOTDIFFN,TOTWDENI,TOTDDC4,TOTGRO,
     1 TOTGRA,TOTNSINKI,TOTNSINKO,TOTNGRA,TOTNGRAX,
```

```
     1 TOTXGRZ,TOTMGROM,TOTMACM,TOTDIFFP,TOTPDENI,
     1 TOTDDC5,TOTPGRX,TOTPGROM,TOTOGRO,TOTOGRA,
     1 TOTOGRX,TOTDDC6,TOTONIT,TOTBPMIN,TOTBPDEN,
     1 TOTBPMAC,TOTBNMIN,TOTBNDEN,TOTNNIT,TOTBNMAC,
     1 TOTBNDIF,TOTINIT,TOTBIDEN,TOTBIDIF,TOTSMIN,
     1 TOTGROM,TOTFAGR,TOTFAM,TOTFBGR,TOTFBM,
     1 TOTZMRT,TOTDZM,
     1 TOTPGRO,TOTOSMIN,TOTOGROM,TOTBPDIF,TOTPGRA,
     1 TOTDMRTX,TOTXMRT,TOTDGRZ,TOTSMFI,TOTDETP,
     1 TOTETP,TOTSEQ
      SAVE TOTGRZ,TOTPRED,TOTASINKI,TOTASINKO,
     1 TOTGRW,TOTAGRZ,TOTDSINKI,TOTDSINKO,TOTDECOM,
     1 TOTDFEK,TOTDFI,TOTDDEN,TOTDPRED,
     1 TOTSSINKP,TOTSSINKD,TOTSSINKN,TOTSMACM,
     1 TOTSDENI,TOTDIFFN,TOTWDENI,TOTDDC4,TOTGRO,
     1 TOTGRA,TOTNSINKI,TOTNSINKO,TOTNGRA,TOTNGRAX,
     1 TOTXGRZ,TOTMGROM,TOTMACM,TOTDIFFP,TOTPDENI,
     1 TOTDDC5,TOTPGRX,TOTPGROM,TOTOGRO,TOTOGRA,
     1 TOTOGRX,TOTDDC6,TOTONIT,TOTBPMIN,TOTBPDEN,
     1 TOTBPMAC,TOTBNMIN,TOTBNDEN,TOTNNIT,TOTBNMAC,
     1 TOTBNDIF,TOTINIT,TOTBIDEN,TOTBIDIF,TOTSMIN,
     1 TOTGROM,TOTFAGR,TOTFAM,TOTFBGR,TOTFBM,
     1 TOTZMRT,TOTDZM,
     1 TOTPGRO,TOTOSMIN,TOTOGROM,TOTBPDIF,TOTPGRA,
     1 TOTDMRTX,TOTXMRT,TOTDGRZ,TOTSMFI,TOTDETP,
     1 TOTETP,TOTSEQ
      REAL FOXY,DININ,PO4IN
      SAVE FOXY,DININ,PO4IN
      REAL CONC1TO2(NIM),CONC2TO3(NIM)
      SAVE CONC1TO2,CONC2TO3

      IF(IPROBE.GT.1) GOTO 200
C     --- PROBE 1
      IF(ICHAP.GT.1) GOTO 120
C     CHAPTER 1  MODIFY DEFAULT DATA
C     GROUP 1
      N=26
      LSTEP=1000
      TFRAC(2)=600.
      TSDAY=INT(86400./TFRAC(2))
C     GROUP 2
      ZDIM=18.
      AREAHZ=5.5E6
      SAREAKM2(IPROBE)=AREAHZ/1000000.
      INDARE=2
C     GROUP 3
      SOLVAR(JRHOU)=.TRUE.
      SOLVAR(JRHOV)=.TRUE.
      SOLVAR(JH)   =.TRUE.
      SOLVAR(JK)   =.TRUE.
      SOLVAR(JD)   =.TRUE.
      DO I=JZOO,JBNO
         SOLVAR(I) =.TRUE.
      ENDDO
C     GROUP 4
      RHOREF=999.975
C     GROUP 7
      CORI=1.2E-4
      BETA=1.6
      OPEN(UNIT=8,FILE='FLOW.DAT',FORM='FORMATTED',STATUS='OLD')
      OPEN(UNIT=10,FILE='METEO.DAT',FORM='FORMATTED',STATUS='OLD')
      OPEN(UNIT=7,FILE='BIOPAR.PAR',FORM='FORMATTED',STATUS='OLD')
      CALL BIOPAR(7)
```

```
      CLOSE(7)
      ETD(IPROBE)=10.
      OPEN(UNIT=9,FILE='NUTIN.DAT',FORM='FORMATTED',STATUS='OLD')
C     GROUP 8
      VST2(JH)=16.76E6
      VST2(JK)=1.E-5
      VST2(JD)=5.E-8
      VST2(JPHY)=0.1
      VST2(JZOO)=VST2(JPHY)/700.
      VST2(JDET)=0.1
      VST2(JSED)=4.
      VST2(JDIN)=1.1
      VST2(JANF)=0.1
      VST2(JPO4)=0.01
      VST2(JOXY)=10.0
      VST2(JBIP)=0.3
      VST2(JBNH)=0.01
      VST2(JBNO)=0.01
      MACOFF=.TRUE.
C     GROUP 9
      IKBLZ(JRHOU)=1
      IKBLZ(JRHOV)=1
C     GROUP 11
      INIOUT=.FALSE.
      NSTAT=9000000
      NPROF=NSTAT
      ITPLOT=1
      OPEN(UNIT=15,FILE='RESULTF1.DAT',STATUS='UNKNOWN')
      OPEN(UNIT=16,FILE='PHYSPRO1.DAT',STATUS='UNKNOWN')
      OPEN(UNIT=17,FILE='NUTPRO1.DAT',STATUS='UNKNOWN')
      OPEN(UNIT=18,FILE='FLOWZOO.DAT',STATUS='UNKNOWN')
C     GROUP 12
      NPROBE=3
      RETURN
120   CONTINUE

C     CHAPTER 2  TRANSIENT BOUNDARY CONDITIONS
      IF(ICHAP.GT.2)GOTO 130
      READ(10,*) ITIMESTEP,UG,VG,FH,FSB
      FLUXHZ(JRHOU)=-1.3E-3*UG*SQRT(UG**2+VG**2)
      FLUXHZ(JRHOV)=-1.3E-3*VG*SQRT(UG**2+VG**2)
      FLUXHZ(JH)=FH
      FLXRAD=FSB
      CALL OXYFLX(SQRT(UG**2+VG**2),FOXY)
      FLUXHZ(JOXY)=FOXY
      FLUXHZ(JDIN)=-NDEPPAR/1000./365./86400.
     1             -POINTDIN/SAREAKM2(IPROBE)/1000./365./86400.
      FLUXHZ(JPO4)=-PDEPPAR/1000./365./86400.
     1             -POINTPO4/SAREAKM2(IPROBE)/1000./365./86400.
      IF(MOD(ISTEP,TSDAY).EQ.0)THEN
         READ(8,*) IYMD,QROUT
         READ(9,*) IYMD,DININ,PO4IN
         QINFL(NM1)=QROUT
         QOUTFL(NM1)=QROUT
         PHIIN(NM1,JH)=F(NM1,JH)
         PHIIN(NM1,JDIN)=DININ
         PHIIN(NM1,JPO4)=PO4IN
      ENDIF
      RETURN
130   CONTINUE

C     CHAPTER 3  ADDITIONAL SOURCE TERMS
      IF(ICHAP.GT.3)GOTO 140
```

```
      IF(ISTEP.GT.0)THEN
         IF(J.GE.JZOO.AND.J.LE.JBNO)THEN
            CALL BIOLA(-FLXRAD,MACOFF,ETD(IPROBE),SAREAKM2(IPROBE),
     1 TOTGRZ,TOTPRED,TOTASINKI,TOTASINKO,TOTGRW,
     1 TOTAGRZ,TOTDSINKI,TOTDSINKO,TOTDECOM,TOTDGRZ,TOTDPRED,
     1 TOTDFEK,TOTDFI,TOTSMFI,TOTDDEN,TOTSSINKP,TOTSSINKD,TOTSSINKN,
     1 TOTSMACM,TOTSDENI,TOTSMIN,TOTDIFFN,TOTWDENI,TOTDDC4,TOTGRO,
     1 TOTGRA,TOTGROM,TOTNSINKI,TOTNSINKO,TOTNGRA,
     1 TOTNGRAX,TOTXGRZ,TOTMGROM,TOTMACM,TOTDIFFP,TOTPDENI,
     1 TOTDDC5,TOTPGRO,TOTPGRA,TOTPGRX,TOTPGROM,TOTOGRO,TOTOGRA,
     1 TOTOGRX,TOTOGROM,TOTOSMIN,TOTDDC6,TOTONIT,TOTBPMIN,
     1 TOTBPDEN,TOTBPMAC,TOTBPDIF,TOTBNMIN,TOTBNDEN,TOTNNIT,
     1 TOTBNMAC,TOTBNDIF,TOTINIT,TOTBIDEN,TOTBIDIF, TOTFAGR,
     1 TOTFAM,TOTFBGR,TOTFBM,TOTZMRT,TOTDZM,TOTDMRTX,TOTXMRT,
     1 TOTDETP,TOTETP,TOTSEQ)
         ENDIF
      ENDIF
      RETURN
140   CONTINUE

C     CHAPTER 4  ADDITIONAL OUTPUT
      WRITE(15,*)IYMD,Z(N),F(NM1,JTE)
      IF(MOD(ISTEP,TSDAY).EQ.0)THEN
         DO I=NM1,2,-1
            WRITE(16,*)IYMD,Z(I),F(I,JTE)
            WRITE(17,*)IYMD,Z(I),F(I,JDIN),F(I,JPO4)
         ENDDO
      ENDIF
      CONC1TO2(JH)=F(NM1,JH)
      DO I=JZOO,JBNO
         CONC1TO2(I)=F(NM1,I)
      ENDDO
      RETURN

200   IF(IPROBE.GT.2) GOTO 300
C     --- PROBE 2
      IF(ICHAP.GT.1) GOTO 220
C     CHAPTER 1  MODIFY DEFAULT DATA
C     GROUP 1
      N=26
C     GROUP 2
      ZDIM=18.
      AREAHZ=5.5E6
      SAREAKM2(IPROBE)=AREAHZ/1000000.
C     GROUP 7
      ETD(IPROBE)=10.
C     GROUP 8
      VST2(JH)=16.76E6
      VST2(JK)=1.E-5
      VST2(JD)=5.E-8
      VST2(JPHY)=0.1
      VST2(JZOO)=VST2(JPHY)/700.
      VST2(JDET)=0.1
      VST2(JSED)=4.
      VST2(JDIN)=1.1
      VST2(JANF)=0.1
      VST2(JPO4)=0.01
      VST2(JOXY)=10.0
      VST2(JBIP)=0.3
      VST2(JBNH)=0.01
      VST2(JBNO)=0.01
C     GROUP 11
      OPEN(UNIT=25,FILE='RESULTF2.DAT',STATUS='UNKNOWN')
```

```
      OPEN(UNIT=26,FILE='PHYSPRO2.DAT',STATUS='UNKNOWN')
      OPEN(UNIT=27,FILE='NUTPRO2.DAT',STATUS='UNKNOWN')
      RETURN
220   CONTINUE

C     CHAPTER 2  TRANSIENT BOUNDARY CONDITIONS
      IF(ICHAP.GT.2)GOTO 230
      FLUXHZ(JRHOU)=-1.3E-3*UG*SQRT(UG**2+VG**2)
      FLUXHZ(JRHOV)=-1.3E-3*VG*SQRT(UG**2+VG**2)
      FLUXHZ(JH)=FH
      FLXRAD=FSB
      CALL OXYFLX(SQRT(UG**2+VG**2),FOXY)
      FLUXHZ(JOXY)=FOXY
      FLUXHZ(JDIN)=-NDEPPAR/1000./365./86400.
     1               -POINTDIN/SAREAKM2(IPROBE)/1000./365./86400.
      FLUXHZ(JPO4)=-PDEPPAR/1000./365./86400.
     1               -POINTPO4/SAREAKM2(IPROBE)/1000./365./86400.
      IF(MOD(ISTEP,TSDAY).EQ.0)THEN
         QINFL(NM1)=QROUT
         QOUTFL(NM1)=QROUT
         PHIIN(NM1,JH)=CONC1TO2(JH)
         PHIIN(NM1,JDIN)=CONC1TO2(JDIN)
         PHIIN(NM1,JPO4)=CONC1TO2(JPO4)
         PHIIN(NM1,JDET)=CONC1TO2(JDET)
         PHIIN(NM1,JPHy)=CONC1TO2(JPHY)
         PHIIN(NM1,JANF)=CONC1TO2(JANF)
         PHIIN(NM1,JZOO)=CONC1TO2(JZOO)
      ENDIF
      RETURN
230   CONTINUE

C     CHAPTER 3  ADDITIONAL SOURCE TERMS
      IF(ICHAP.GT.3)GOTO 240
      IF(ISTEP.GT.0)THEN
         IF(J.GE.JZOO.AND.J.LE.JBNO)THEN
            CALL BIOLA(-FLXRAD,MACOFF,ETD(IPROBE),SAREAKM2(IPROBE),
     1 TOTGRZ,TOTPRED,TOTASINKI,TOTASINKO,TOTGRW,
     1 TOTAGRZ,TOTDSINKI,TOTDSINKO,TOTDECOM,TOTDGRZ,TOTDPRED,
     1 TOTDFEK,TOTDFI,TOTSMFI,TOTDDEN,TOTSSINKP,TOTSSINKD,TOTSSINKN,
     1 TOTSMACM,TOTSDENI,TOTSMIN,TOTDIFFN,TOTWDENI,TOTDDC4,TOTGRO,
     1 TOTGRA,TOTGROM,TOTNSINKI,TOTNSINKO,TOTNGRA,
     1 TOTNGRAX,TOTXGRZ,TOTMGROM,TOTMACM,TOTDIFFP,TOTPDENI,
     1 TOTDDC5,TOTPGRO,TOTPGRA,TOTPGRX,TOTPGROM,TOTOGRO,TOTOGRA,
     1 TOTOGRX,TOTOGROM,TOTOSMIN,TOTDDC6,TOTONIT,TOTBPMIN,
     1 TOTBPDEN,TOTBPMAC,TOTBPDIF,TOTBNMIN,TOTBNDEN,TOTNNIT,
     1 TOTBNMAC,TOTBNDIF,TOTINIT,TOTBIDEN,TOTBIDIF, TOTFAGR,
     1 TOTFAM,TOTFBGR,TOTFBM,TOTZMRT,TOTDZM,TOTDMRTX,TOTXMRT,
     1 TOTDETP,TOTETP,TOTSEQ)
         ENDIF
      ENDIF
      RETURN
240   CONTINUE

C     CHAPTER 4  ADDITIONAL OUTPUT
      WRITE(25,*)IYMD,Z(N),F(NM1,JTE)
      IF(MOD(ISTEP,TSDAY).EQ.0)THEN
         DO I=NM1,2,-1
            WRITE(26,*)IYMD,Z(I),F(I,JTE)
            WRITE(27,*)IYMD,Z(I),F(I,JDIN),F(I,JPO4)
         ENDDO
      ENDIF
      CONC2TO3(JH)=F(NM1,JH)
      DO I=JZOO,JBNO
```

```
          CONC2TO3(I)=F(NM1,I)
      ENDDO
      RETURN

300   IF(IPROBE.GT.3) RETURN
C     --- PROBE 3
      IF(ICHAP.GT.1) GOTO 320
C     CHAPTER 1  MODIFY DEFAULT DATA
C     GROUP 1
      N=26
C     GROUP 2
      ZDIM=18.
      AREAHZ=5.5E6
      SAREAKM2(IPROBE)=AREAHZ/1000000.
C     GROUP 7
      ETD(IPROBE)=10.
C     GROUP 8
      VST2(JH)=16.76E6
      VST2(JK)=1.E-5
      VST2(JD)=5.E-8
      VST2(JPHY)=0.1
      VST2(JZOO)=VST2(JPHY)/700.
      VST2(JDET)=0.1
      VST2(JSED)=4.
      VST2(JDIN)=1.1
      VST2(JANF)=0.1
      VST2(JPO4)=0.01
      VST2(JOXY)=10.0
      VST2(JBIP)=0.3
      VST2(JBNH)=0.01
      VST2(JBNO)=0.01
C     GROUP 11
      OPEN(UNIT=35,FILE='RESULTF3.DAT',STATUS='UNKNOWN')
      OPEN(UNIT=36,FILE='PHYSPRO3.DAT',STATUS='UNKNOWN')
      OPEN(UNIT=37,FILE='NUTPRO3.DAT',STATUS='UNKNOWN')
      RETURN
320   CONTINUE

C     CHAPTER 2  TRANSIENT BOUNDARY CONDITIONS
      IF(ICHAP.GT.2)GOTO 330
      FLUXHZ(JRHOU)=-1.3E-3*UG*SQRT(UG**2+VG**2)
      FLUXHZ(JRHOV)=-1.3E-3*VG*SQRT(UG**2+VG**2)
      FLUXHZ(JH)=FH
      FLXRAD=FSB
      CALL OXYFLX(SQRT(UG**2+VG**2),FOXY)
      FLUXHZ(JOXY)=FOXY
      FLUXHZ(JDIN)=-NDEPPAR/1000./365./86400.
     1             -POINTDIN/SAREAKM2(IPROBE)/1000./365./86400.
      FLUXHZ(JPO4)=-PDEPPAR/1000./365./86400.
     1             -POINTPO4/SAREAKM2(IPROBE)/1000./365./86400.
      IF(MOD(ISTEP,TSDAY).EQ.0)THEN
          QINFL(NM1)=QROUT
          QOUTFL(NM1)=QROUT
          PHIIN(NM1,JH)=CONC2TO3(JH)
          PHIIN(NM1,JDIN)=CONC2TO3(JDIN)
          PHIIN(NM1,JPO4)=CONC2TO3(JPO4)
          PHIIN(NM1,JDET)=CONC2TO3(JDET)
          PHIIN(NM1,JPHy)=CONC2TO3(JPHY)
          PHIIN(NM1,JANF)=CONC2TO3(JANF)
          PHIIN(NM1,JZOO)=CONC2TO3(JZOO)
      ENDIF
      RETURN
330   CONTINUE
```

```
C       CHAPTER 3  ADDITIONAL SOURCE TERMS
        IF(ICHAP.GT.3)GOTO 340
        IF(ISTEP.GT.0)THEN
           IF(J.GE.JZOO.AND.J.LE.JBNO)THEN
              CALL BIOLA(-FLXRAD,MACOFF,ETD(IPROBE),SAREAKM2(IPROBE),
     1 TOTGRZ,TOTPRED,TOTASINKI,TOTASINKO,TOTGRW,
     1 TOTAGRZ,TOTDSINKI,TOTDSINKO,TOTDECOM,TOTDGRZ,TOTDPRED,
     1 TOTDFEK,TOTDFI,TOTSMFI,TOTDDEN,TOTSSINKP,TOTSSINKD,TOTSSINKN,
     1 TOTSMACM,TOTSDENI,TOTSMIN,TOTDIFFN,TOTWDENI,TOTDDC4,TOTGRO,
     1 TOTGRA,TOTGROM,TOTNSINKI,TOTNSINKO,TOTNGRA,
     1 TOTNGRAX,TOTXGRZ,TOTMGROM,TOTMACM,TOTDIFFP,TOTPDENI,
     1 TOTDDC5,TOTPGRO,TOTPGRA,TOTPGRX,TOTPGROM,TOTOGRO,TOTOGRA,
     1 TOTOGRX,TOTOGROM,TOTOSMIN,TOTDDC6,TOTONIT,TOTBPMIN,
     1 TOTBPDEN,TOTBPMAC,TOTBPDIF,TOTBNMIN,TOTBNDEN,TOTNNIT,
     1 TOTBNMAC,TOTBNDIF,TOTINIT,TOTBIDEN,TOTBIDIF, TOTFAGR,
     1 TOTFAM,TOTFBGR,TOTFBM,TOTZMRT,TOTDZM,TOTDMRTX,TOTXMRT,
     1 TOTDETP,TOTETP,TOTSEQ)
           ENDIF
        ENDIF
        RETURN
340     CONTINUE
C       CHAPTER 4  ADDITIONAL OUTPUT
        WRITE(35,*)IYMD,Z(N),F(NM1,JTE)
        IF(MOD(ISTEP,TSDAY).EQ.0)THEN
           DO I=NM1,2,-1
              WRITE(36,*)IYMD,Z(I),F(I,JTE)
              WRITE(37,*)IYMD,Z(I),F(I,JDIN),F(I,JPO4)
           ENDDO
           WRITE(18,*)ISTEP,TOTGRZ,TOTPRED,TOTZMRT
           TOTGRZ=0
           TOTPRED=0
           TOTZMRT=0
        ENDIF
        RETURN
        END
```

6.12 Use postprocessing program

The CASE output can only be used so far for getting results from the model. It is a good idea to write a special program for handling your results and put it in a format you want. Perhaps you want to compare the results from several runs, calculate yearly budgets, or compare results with environmental criteria. Such tasks are sometimes better done in a separate program.

7 Calibration

The BIOLA model needs calibration to observations from the specific lake it is going to simulate. Calibration is a tiresome task, since no automatic routine is available for calibrating BIOLA, but it is necessary. Several different variables can be observed and used for calibration (Section 3), the more the better. The available data are with advantage divided in two periods, one is used for calibration and the other is left for validation, i.e. a split sample test. When modelling several subbasins it is possible to choose some subbasins for calibration and others for validation.

The physical part, i.e. PROBE, does most often not need calibration, so available temperature observations can be used for validation. On the other hand, the physical model needs to be checked for grid independence (see PROBE Manual).

At model set up, all variables are given initial values (from observations or guesses). These are sometimes changed later during calibration to avoid trends in the variables. Some parameters of the model are then calibrated against measurements. These parameters are tested sequentially during calibration. The visual appearance of time series compared to observations has been used for choosing the best parameter values based on the level and seasonal variation of the state variables. In addition the process flows give information that is useful for understanding the effect of changing parameters.

BIOLA is calibrated manually without objective criteria for what is the best parameter set. This means that during calibration there is flexibility regarding which type of discrepancy is considered most acceptable. As the processes depend on several parameters different parameters may be used to influence a process. In addition the variables are tightly coupled, so an adjustment of one parameter can affect several variables, some positively and some negatively. To counteract the ambiguity in calibration that comes from having multiple parameters influencing the same state variables and multiple parameters influencing the same process, many different data series are needed for calibration. For examples see Section 3. Since the sediment variables seldom are calibrated, it is important to have a validation data set. In the lakes modelled so far the differences between data and model results are similar for calibration and validation periods.

In Pers and Persson (2003) the parameter values resulting from calibration of BIOLA to three lakes are discussed. The hope is that these comparisons will be the first step towards making general recommendations on suitable parameters in different types of lakes, so that work during set-up can be reduced and less data needed for calibration in the future. Because the model continued to be developed during the applications, the model set-ups compared in the paper are somewhat different.

7.1 Parameter recommendations for BIOLA

Calibration is typically done in several steps. First the nutrient level and algae concentrations need to be attacked. These make the core of BIOLA. The are tightly couple, if the nutrient levels are wrong the phytoplankton is doomed to be wrong also, but algae are also influenced by the other processes. When a general picture of the lake with reasonable levels of algae and nutrients are reached, the calibration zooms in on other variables. Further calibration may perturb the nutrient and algae concentrations and they most probably have to be assessed again. A guideline can be to attack the largest differences first, because then subsequent changes will more probable be close to the final value.

Parameter definitions can be found in Table 3, while Table 6 groups the parameters according to the process (process subroutine) they influence. All these parameter could in principle be used to influence the process, but not all are suitable for calibration. In reality only some of the parameters are calibrated. Table 13 lists the most common parameters used to influence the different state variables. Below these parameters and some of their effects will be discussed. Note that the effect will vary between lakes.

Table 13. Parameters most commonly used to influence the state variables. Parameter definitions are in Table 3 and state variables in Table 1. Parameters in bold are those primarily calibrated.

Parameter	State variable
ADEN	DIN
DDCC	D, DIN, PO4
FMRT	FB
KDIFF	DIN, PO4
KDIN	A, CB, DIN
KPO4	A, CB, PO4
MINMAX	B
MMRT	M
MYMAXM	M
P2	D, Z
P3	CB, Z
PREDF	FA, FB
PREDZ	Z, FA
RESP	B, D
SEQCON	B
SINKD	D, B
SINKP	A, B
XMRT	CB, A, D
ZMRT	Z, A, CB, D

The first parameters to calibrate for nutrient and algae concentrations are KDIFF and ZMRT. KDIFF is used to regulate the nutrient exchange with sediment, while ZMRT, the zooplankton mortality, influence the algae level. KDIFF may limit the denitrification in sediments by regulating the supply of DIN from the water. Too high ZMRT kills the zooplankton leaving the algae to explode, while a too low ZMRT exterminate the algae. With the ZMRT parameter phytoplankton and zooplankton can be forced to one or several peaks a year.

If the nutrient level in the lake are simulated well, and the total algae concentration are too high or low, the half saturation parameters for algal growth (KDIN, KPO4) may have to be adjusted for the algae to grow/not explode. The alga species in a lake can be adjusted to the nutrient level of the lake. To balance phytoplankton and cyanobacteria concentrations primarily the mortality of cyanobacteria (XMRT) is used, but also the sinking velocity of phytoplankton can be used (SINKP).

The algal growth is often responsible for the seasonal variation in nutrient in the lake, but denitrification can contribute as it has a temperature dependence. Denitrification (ADEN) will reduce the nitrogen in the lake. Influencing the nutrients, but often minor compared to the supply by the inflow, is the mineralisation of detritus (DDCC). The loss of detritus concentration is a balance between mineralisation and sinking (SINKD). Lower sinking velocity means that more of the detritus will be mineralised in the water.

Modelled zooplankton prefers phytoplankton before cyanobacteria and detritus as food. This preference can be adjusted with P2 and P3. The parameters may influence the zooplankton concentration and the balance between phytoplankton and cyanobacteria.

The mineralisation of sediment organic material (MINMAX) can also contribute to nutrients in the lake. Without observations the initial value of the sediment organic matter is balanced against the mineralisation to a stable concentration of organic matter in sediment after the supply through settling of phytoplankton and detritus is calibrated. Pay attention to the initial value of the sediment variables (Table 1), since these can determine if the sediment will act as a sink or a source of nutrients. It is possible to include a sequestering of sediment organic matter (SEQCON) if necessary.

Resuspension is derived for 3-hour interval wind observations. If other wind is used the resuspension should be adjusted (RESP).

Macrophytes are calibrated with their growth rate (MYMAXM) and their mortality rate (MMRT). Fishes are calibrated primarily with the mortality rate of FB (FMRT), but the predation rates (PREDZ, PREDF) can also be used. If no observations are available, which is common for these variables, they are calibrated to a steady state.

Acknowledgement

This work was performed within the Swedish Water Management Research Program (VASTRA) which is financed by the Swedish Foundation for Strategic Environmental Research (MISTRA). The development of BIOLA has also been supported by the Swedish Environmental Protection Agency (SEPA) through the project 'Eutrofieringsproblem och retentionsprocesser' (Eutrophication and retention processes) and by the Swedish Meteorological and Hydrological Institute (SMHI).

References

References in the Manual

Ahlkrona, M. (2002). Phosphorus in a biogeochemical lake model. *SMHI RH Report* No. 17, Swedish Meteorological and Hydrological Institute, Norrköping, 56 pp.

Ashton, G.D. (1983). Lake ice decay. *Cold Regions Science and Technology*, 8(1), pp. 83-86.

Correll, D.L. (1998). The role of phosphorus in the eutrophication of receiving waters: A review. *J. Environ. Qual.*, 27, pp. 261-266.

Elser, J.J., and R.P. Hassett (1994). A stoichiometric analysis of the zooplankton-phytoplankton interaction in marine and freshwater ecosystems. *Nature*, 370, pp. 211-213.

Hessen, D.O., T. Andersen, P. Brettum, and B.A. Faafeng (2003). Phytoplankton contribution to sestonic mass and elemental ratios in lakes: Implications for zooplankton nutrition. *Limnol. Oceanogr.*, 48, pp. 1289-1296.

Hondzo, M., and H.G. Stefan (1993). Lake Water Temperature Simulation Model. *Journal of Hydraulic Engineering*, 119, pp. 1251-1273.

Marmefelt, E., B. Håkansson, A.C. Erichsen, and I.S. Hansen (2000). Development of an Ecological Model System for the Kattegat and the Southern Baltic. Final Report to the Nordic Councils of Ministers. *SMHI RO Report* No. 29, Swedish Meteorological and Hydrological Institute, Norrköping, 76 pp.

Omstedt, A. (1990). A coupled one-dimensional sea ice-ocean model applied to a semi-enclosed basin. *Tellus*, 42A, pp. 568-582.

Pers, B.C. (2002). Model description of BIOLA - a biogeochemical lake model (including literature review of processes). *SMHI RH Report* No. 16, Swedish Meteorological and Hydrological Institute, Norrköping, 69 pp.

Pers, B.C., and I. Persson (2003). Comparison of a biogeochemical model in different lakes. *Nordic Hydrology*, 34(5), pp.

Rutgersson, A., A.-S. Smedman, and A. Omstedt (2001). Measured and simulated latent and sensible heat fluxes at two marine sites in the Baltic Sea. *Boundary-Layer Meteorology*, 99, pp. 53-84.

Sahlberg, J. (1988). Modelling the thermal regime of a lake during the winter season. *Cold Regions Science and Technology*, 15, pp. 151-159.

Stigebrandt, A. (1991). Computations of oxygen fluxes through the sea surface and the net production of organic matter with application to the Baltic and adjacent Seas. *Limnol. Oceanogr.*, 36, pp. 4444-454.

Svensson, U. (1986). Dynamics of coupled reservoirs. CASE Report C10, SMHI, Norrköping.

Svensson, U. (1998). PROBE Program Boundary Layers in the Environment - System Description and Manual. *SMHI RO Report* No. 24, Swedish Meteorological and Hydrological Institute, Norrköping, 42 pp.

References of BIOLA

Ahlkrona, M. (2002). Phosphorus in a biogeochemical lake model. *SMHI RH Report* No. 17, Swedish Meteorological and Hydrological Institute, Norrköping, 56 pp.

Arheimer, B., B.C. Pers and I. Persson (2002). Eutrofieringsproblem och retentionsprocesser (Eutrophication problems and retention processes), *Vastra Working Paper* 24 (in Swedish), 19 pp.

Pers, B.C. (2002). Model description of BIOLA - a biogeochemical lake model (including literature review of processes). *SMHI RH Report* No. 16, Swedish Meteorological and Hydrological Institute, Norrköping, 69 pp.

Pers, B.C., and B. Arheimer (2001). A model of biogeochemical processes in lakes to be used in eutrophication management. In: *Connecting phosphorus transfer from agriculture to impacts in surface waters*. Eds: Haygarth, P.M., Condron, P.J., Butler, P.J. and Crisholm, J.S. Proceedings at the IPTW Workshop, England, 28th Aug.-1th Sept. 2001, Inst. Grassland and Environmental research, Plymouth University, p. 92.

Pers, B.C., and I. Persson (2003). Comparison of a biogeochemical model in different lakes. *Nordic Hydrology*, 34(5), pp.

Pers, B.C., and I. Persson (2002). Comparison of a biogeochemical model in different lakes, In: XXII Nordic Hydrological Conference, Nordic Association for Hydrology, Røros, Norway 4-7 August 2002, Ed: Å. Killingtveit, *NHP Report* No. 47, Volume I, pp. 369-378.

Selected references of PROBE

Omstedt, A. (1990). A coupled one-dimensional sea ice-ocean model applied to a semi-enclosed basin. *Tellus*, 42A, pp. 568-582.

Sahlberg, J. (1984). Ice covered lake with sediment heat flux. *CASE Report* C4, Swedish Meteorological and Hydrological Institute, Norrköping.

Sahlberg, J. (1988). Modelling the thermal regime of a lake during the winter season. *Cold Regions Science and Technology*, 15, pp. 151-159.

Svensson, U. (1984). Thermocline development in a reservoir with in- and outflows. *CASE Report* C3, Swedish Meteorological and Hydrological Institute, Norrköping.

Svensson, U. (1986). Dynamics of coupled reservoirs. *CASE Report* C10, Swedish Meteorological and Hydrological Institute, Norrköping.

Svensson, U. (1998). PROBE Program Boundary Layers in the Environment - System Description and Manual. *SMHI RO Report* No. 24, Swedish Meteorological and Hydrological Institute, Norrköping, 42 pp.

Appendix A. The equations of the biogeochemical processes

Collection of model equations, constants, model parameters and variables for easy access.

Table A1. Symbol of state variables in BIOLA process equations.

Symbol	State variable
A	Phytoplankton
B	Sediment organic carbon
BIP	Dissolved phosphorus in sediment water
BNH	Dissolved ammonium in sediment water
BNO	Dissolved nitrate in sediment water
CB	Cyanobacteria
D	Detritus
DIN	Dissolved inorganic nitrogen
FA	Planktivorous fish
FB	Piscivorous fish
M	Macrophyte
O2	Oxygen gas
PO4	Phosphate
Z	Zooplankton

Table A2. Model formulation of biogeochemical sources Φ and sinks ϕ.

Process	Equations
Phytoplankton growth	$\Phi_A = \mu_A * \theta^{T-T_{ref}} * \frac{I(z)}{I_{opt}} * e^{1-\frac{I(z)}{I_{opt}}} * \min\left(\frac{PO4}{k_{PO4}+PO4}, \frac{DIN}{k_{DIN}+DIN}\right) * A$ $\Phi_{O2} = C_{OC} * \Phi_A$ $\phi_{DIN} = -C_{NC} * \Phi_A$ $\phi_{PO4} = -C_{PC} * \Phi_A$
Light attenuation	$I(z) = I_0 * e^{-k*z}$ $I_0 = I0 * k_I$ $k = k_w + k_M * M + k_P * (A + CB) + k_D * D$
Cyanobacteria growth on DIN	$\Phi_{CB2} = \mu_{ANFIX} * \theta^{T-T_{ref}} * \frac{I(z_s)}{I_{opt}} * e^{1-\frac{I(z_s)}{I_{opt}}} * \frac{PO4}{k_{PO4}+PO4} * CB$ if $\frac{PO4}{k_{PO4}+PO4} \le \frac{DIN}{k_{DIN}+DIN}$ else $\Phi_{CB2} = 0$

Table A2. Continued.

Process	Equations
Cyanobacteria growth on N_2, i.e. nitrogen fixation	$\Phi_{CB1} = \mu_{ANFIX} * \theta^{T-T_{ref}} * \frac{I(z_s)}{I_{opt}} * e^{1-\frac{I(z_s)}{I_{opt}}} * \frac{PO4}{k_{PO4} + PO4} * \frac{1}{1+\left(\frac{DIN}{PO4 * k_{NFIX}}\right)^{snf4}} * CB$ if $\frac{DIN}{k_{DIN} + DIN} < \frac{PO4}{k_{PO4} + PO4}$ else $\Phi_{CB1} = 0$
Cyanobacteria growth both on DIN and N_2	$\Phi_{O2} = C_{OC} * (\Phi_{CB1} + \Phi_{CB2})$ $\phi_{DIN} = -C_{NC} * \Phi_{CB2}$ $\phi_{PO4} = -C_{PC} * (\Phi_{CB1} + \Phi_{CB2})$
Macrophyte growth	$\Phi_M = \mu_M * \theta_M{}^{T-T_{ref}} * \frac{I(z_s)}{I_{Mopt}} * e^{1-\frac{I(z_s)}{I_{Mopt}}} * \min\left(\frac{\frac{BIP}{d_{sed}} + PO4}{k_{MIP} + \frac{BIP}{d_{sed}} + PO4}, \frac{\frac{BNH}{d_{sed}} + DIN}{k_{MIN} + \frac{BNH}{d_{sed}} + DIN}\right) * M$ $\Phi_{O2} = C_{OC} * \Phi_M$ $\phi_{BNH} = -C_{NC} * \Phi_M * z_s$ if $C_{NC} * \Phi_M * z_s < BNH - BNH_{\min}$ $\begin{cases} \phi_{BNH} = -(BNH - BNH_{\min}) \\ \phi_{DIN} = -\left(\frac{C_{NC} * \Phi_M * z_s + \phi_{BNH}}{\Delta z_{cell}}\right) \end{cases}$ if $C_{NC} * \Phi_M * z_s > BNH - BNH_{\min}$ $\phi_{BIP} = -C_{PC} * \Phi_M * z_s$ if $C_{PC} * \Phi_M * z_s < BIP - BIP_{\min}$ $\begin{cases} \phi_{BIP} = -(BIP - BIP_{\min}) \\ \phi_{PO4} = -\left(\frac{C_{PC} * \Phi_M * z_s + \phi_{BIP}}{\Delta z_{cell}}\right) \end{cases}$ if $C_{PC} * \Phi_M * z_s > BIP - BIP_{\min}$
Cyanobacteria mortality	$\phi_{CB} = -m_{ANFIX} * CB$ $\Phi_D = -\phi_{CB}$
Macrophyte mortality	$\phi_M = -m_M * M$ $\Phi_B = -\phi_M * z$
Zooplankton grazing on phytoplankton, cyanobacteria and detritus	$\phi_A = -k_G * \frac{s_A * A}{k_g + s_A * A + s_{ANFIX} * CB + s_D * D} * Z$ $\phi_{Cb} = -k_G * \frac{s_{ANFIX} * CB}{k_g + s_A * A + s_{ANFIX} * CB + s_D * D} * Z$ $\phi_D = -k_G * \frac{s_D * D}{k_g + s_A * A + s_{ANFIX} * CB + s_D * D} * Z$ $\Phi_Z = -e * (\phi_A + \phi_{CB} + \phi_D)$ $\Phi_D = \frac{1-e}{e} \Phi_Z$
Zooplankton mortality	$\phi_Z = -k_{zm} * Z$ $\Phi_D = -\phi_Z$

Table A2. Continued.

Process	Equations
Fish predation on zooplankton	$\phi_Z = -k_{pA} * \theta^{T-T_{ref}} * \frac{Z^2}{k_Z^2 + Z^2} * FA$ $\Phi_{FA} = -\Omega * \phi_Z$ $\Phi_D = -(1-\Omega) * \phi_Z$
Fish predation on fish	$\phi_{FA} = -k_{pB} * \theta^{T-T_{ref}} * \frac{FA^2}{k_{FA}^2 + FA^2} * FB$ $\Phi_{FB} = -\Omega * \phi_{FA}$ $\Phi_D = -(1-\Omega) * \phi_{FA}$
Piscivorous fish mortality	$\phi_{FB} = -k_{mort} * FB$ $\Phi_B = -\phi_{FB} * \frac{A_{cell} * \Delta z_{cell}}{A_{bottom}}$
Detritus mineralisation	$\phi_D = -k_d * \theta^{T-T_{ref}} * D$ $\Phi_{DIN} = -C_{NC} * \phi_D$ $\Phi_{PO4} = -C_{PC} * \phi_D$ $\phi_{O2} = C_{OC} * \phi_D$
Denitrification	$\phi_{BNO} = -k_{dena} * \theta^{T-T_{ref}} * \frac{BNO}{K_m * d_{sed} + BNO} * h(O2)$ $\phi_{DIN} = -k_{den} * \theta^{T-T_{ref}} * \frac{DIN}{K_m + DIN} * (1 - h(O2))$ $\phi_B = C_{CNdenit} * \phi_{BNO}$ $\Phi_{BIP} = -C_{PC} * \phi_B$ $\Phi_{BHN} = -C_{NC} * \phi_B$ $\phi_D = C_{CNdenit} * \phi_{DIN}$ $\Phi_{PO4} = -C_{PC} * \phi_D$ $\Phi_{DIN} = -C_{NC} * \phi_D$
Nitrification	$\phi_{BNH} = -k_{nitr} * \theta_n^{T-T_{nref}} * \min\left(BNH, \frac{O2}{C_{ONnitr} * d_{sed}}\right)$ $\Phi_{BNO} = -\phi_{BNH}$ $\phi_{O2} = C_{ONnitr} * \phi_{BNH} * \frac{A_{bottom}}{A_{cell} * \Delta z_{cell}}$

Table A2. Continued.

Process	Equations
Sediment mineralisation	$\phi_B = -\max\left(k_{sm} * \theta^{T-T_{ref}} * B + \phi_B(denitr), 0\right)$ $\phi_{O2} = C_{OC} * \phi_B * \frac{A_{bottom}}{A_{cell} * \Delta z_{cell}}$ $\Phi_{BNH} = -C_{NC} * \phi_B$ $\Phi_{BIP} = -C_{PC} * \phi_B$
Exchange of nutrients between water and sediment	$\phi_{BIP} = -k_{diff} * \frac{\left(\frac{BIP}{(1+f) * h_{ar}} - PO4\right)}{h_{ar}}$ $f = \frac{BIP_{ads}}{BIP_{dis}}$ $BIP = BIP_{ads} + BIP_{dis}$ $BIP_{ads} = psc * \frac{BIP_{dis}}{k_{ad} + BIP_{dis}}$ $psc = \begin{cases} a_{cap} * h_{ar} & O2 > O_{limit} \\ a_{capan} * h_{ar} * \left(a_{cap} / a_{capan} \right)^{O2 / O_{limit}} & O2 \le O_{limit} \end{cases}$ $\phi_{BNO} = -k_{diff} * \frac{(BNO - DIN * \Delta z_{cell})}{h_{ar}^2}$ $\phi_{BNH} = -k_{diff} * \frac{BNH}{h_{ar}^2}$ $\Phi_{PO4} = -\frac{\phi_{BIP} * A_{botten}}{A_{cell} * \Delta z_{cell}}$ $\Phi_{DIN} = -\frac{(\phi_{BNO} + \phi_{BNH}) * A_{botten}}{A_{cell} * \Delta z_{cell}}$
Sinking and sedimentation of phytoplankton, cyanobacteria and detritus	$\Phi_C = \frac{\partial (v_C * C)}{\partial z}$, $C = A, CB$, or D $\Phi_B = v_C * C$
Resuspension	$\phi_B = \begin{cases} -k_{resp} * C_{resp} * w^2 * B & z > etd \\ 0 & z \le etd \end{cases}$ $\Phi_D = -\phi_B * \frac{A_{bottom}}{A_{cell} * \Delta z_{cell}}$
Sequestering	$\phi_B = -k_s * B$

Table A3. Parameters used in model equations in Table A2.

Symbol	Name in code	Parameter
μ_A	MYMAX	maximum growth rate for phytoplankton at reference temperature (d^{-1})
θ	TETAG	general temperature coefficient
T_{ref}	TREFG	general reference temperature (°C)
I_{opt}	IOPTA	optimal light intensity for phytoplankton and cyanobacteria ($W\ m^{-2}$)
k_{PO4}	KPO4	half-saturation concentration for phytoplankton and cyanobacteria uptake of phosphorus ($mg\ P\ L^{-1}$)
k_{DIN}	KDIN	half-saturation concentration for phytoplankton and cyanobacteria uptake of nitrogen ($mg\ N\ L^{-1}$)
k_w	KW	extinction coefficient of the water (m^{-1})
k_M	KMKONST	macrophyte shading coefficient ($m^2\ g^{-1}$)
k_P	KPKONST	shading coefficient for phytoplankton and cyanobacteria ($m^2\ g^{-1}$)
k_D	KDKONST	detritus shading coefficient ($m^2\ g^{-1}$)
k_I	IZKONST	part of short-wave radiation that penetrates into water
μ_{ANFIX}	MYMAXA	maximum growth rate for cyanobacteria using DIN at reference temperature (d^{-1})
μ_{ANFIX}	MYMAXB	maximum growth rate for cyanobacteria using N_2 at reference temperature (d^{-1})
k_{NFIX}	KNFIX	Redfield ratio N/P in organic matter
$snf4$	SNF4	coefficient in nitrogen fixation equation
μ_M	MYMAXM	maximum growth rate for macrophytes at reference temperature (d^{-1})
θ_M	TETAMAC	temperature coefficient for macrophyte
I_{Mopt}	IOPTM	optimal light intensity for macrophyte ($W\ m^{-2}$)
k_{MIP}	KMIP	half-saturation concentration for macrophyte uptake of phosphorus ($mg\ P\ L^{-1}$)
k_{MIN}	KMIN	half-saturation concentration for macrophyte uptake of nitrogen ($mg\ N\ L^{-1}$)
d_{sed}	DSED2	thickness of active sediment layer (m)
m_{ANFIX}	XMRT	mortality rate for cyanobacteria (d^{-1})
m_M	MMRT	mortality rate for macrophyte (d^{-1})
k_G	GAMMA	grazing rate (d^{-1})
e	EFFI	efficiency of zooplankton grazing
k_g	KG	half-saturation concentration for grazing ($mg\ C\ L^{-1}$)
s_A	P1	selectivity coefficient for grazing on phytoplankton
s_{ANFIX}	P3	selectivity coefficient for grazing on cyanobacteria
s_D	P2	selectivity coefficient for grazing on detritus
k_{zm}	ZMRT	mortality rate for zooplankton (d^{-1})
k_{pA}	PREDZ	maximum predation rate on zooplankton at reference temperature (d^{-1})
k_Z	KZO	half-saturation concentration for zooplankton ($mg\ C\ L^{-1}$)
Ω	OMEGA	part of fish predation that is used for growth (both fishes)
k_{pB}	PREDF	maximum predation rate on planktivorous fish at reference temperature (d^{-1})
k_{FA}	KFIA	half-saturation concentration for fish ($mg\ C\ L^{-1}$)
k_{mort}	FMRT	mortality rate for piscivorous fish (d^{-1})
k_d	DDCC	degradation rate at reference temperature (d^{-1})
k_{dena}	ADEN	maximum area denitrification rate at reference temperature ($g\ N\ m^{-2}\ d^{-1}$)
k_{den}	BDEN	maximum denitrification rate at reference temperature ($mg\ N\ L^{-1}\ d^{-1}$)
k_m	KBNO	half saturation concentration for nitrogen used in denitrification ($mg\ N\ L^{-1}$)

Table A3. Continued.

Symbol	Name in code	Parameter
k_{nitr}	TSC	nitrification rate, depends on d_{sed2} (d^{-1})
θ_n	TETANFIC	temperature coefficient for nitrification
T_{nref}	TNFIC	reference temperature for nitrification (°C)
k_{sm}	MINMAX	mineralisation rate in sediment (d^{-1})
k_s	SEQCON	sequestering rate (d^{-1})
k_{diff}	KDIFF	diffusion coefficient for sediment ($m^2\ s^{-1}$)
h_{ar}	HAR	aerobic sediment layer (m)
a_{capan}	ADSCAPAN	phosphate sorption capacity in anaerobic sediment layer (mg P L^{-1})
a_{cap}	ADSCAP	phosphate sorption capacity in aerobic sediment layer (mg P L^{-1})
O_{limit}	OXYLIMIT	critical oxygen concentration (mg L^{-1})
k_{ad}	KAD	half-saturation concentration for sorption (mg P L^{-1})
v_P	SINKP	sinking velocity of phytoplankton (m d^{-1})
v_{CB}	SINKAN	sinking velocity of blue-green algae (m d^{-1})
v_D	SINKD	sinking velocity of detritus (m d^{-1})
k_{resp}	RESP	resuspension coefficient

Table A4. Constants used in the equations in Table A2.

Constant	Definition	Value
BIP_{min}	Minimum value of BIP	10^{-10}
BNH_{min}	Minimum value of BNH	10^{-10}
$C_{CNdenit}$	Stoichiometric ratio C/N during denitrification (mg C (mg N)$^{-1}$)	1.071
C_{NC}	Stoichiometric ratio N/C in organic matter (mg N (mg C)$^{-1}$)	0.176
C_{OC}	Stoichiometric ratio O_2/C in organic matter (mg O_2 (mg C)$^{-1}$)	2.667
C_{ONnitr}	Stoichiometric ratio O_2/N during nitrification (mg O_2 (mg N)$^{-1}$)	4.572
C_{PC}	Stoichiometric ratio P/C in organic matter (mg P (mg C)$^{-1}$)	0.0244
C_{resp}	Resuspended proportion of sediment organic matter (s^{-1})	$1.5978*10^{-9}$

Table A5. Other symbols used in Table A2.

Symbol	Definition	Unit
A_{cell}	Horizontal area of cell	m^2
A_{bottom}	Bottom area of cell	m^2
etd	z-coordinate for border between erosion-transport bottoms and accumulation bottoms	m
h(x)	Heaviside's step function	$h(x) = \begin{cases} 1 & x > 0 \\ 0 & x < 0 \end{cases}$
I0	Short-wave radiation	W m^{-2}
T	Water temperature	°C
w	3-hour interval wind speed	m s^{-1}
Δz_{cell}	Thickness of cell	m

Appendix B. Example of a BIOLA CASE subroutine

This is the same example that is described in detail in Section 4.4.

```
      SUBROUTINE CASE(ICHAP)

      INCLUDE 'COMP97.INC'
      INCLUDE 'BIOPAR.INC'

      INTEGER ITIMESTEP,IYMD,TSDAY
      SAVE    ITIMESTEP,IYMD,TSDAY
      REAL    UG,VG,FH,FSB,QROUT
      SAVE    UG,VG,FH,FSB,QROUT

      LOGICAL MACOFF
      SAVE    MACOFF
      REAL ETD,SAREAKM2
      SAVE ETD,SAREAKM2
      REAL TOTGRZ,TOTPRED,TOTASINKI,TOTASINKO,
     1 TOTGRW,TOTAGRZ,TOTDSINKI,TOTDSINKO,TOTDECOM,
     1 TOTDFEK,TOTDFI,TOTDDEN,TOTDPRED,
     1 TOTSSINKP,TOTSSINKD,TOTSSINKN,TOTSMACM,
     1 TOTSDENI,TOTDIFFN,TOTWDENI,TOTDDC4,TOTGRO,
     1 TOTGRA,TOTNSINKI,TOTNSINKO,TOTNGRA,TOTNGRAX,
     1 TOTXGRZ,TOTMGROM,TOTMACM,TOTDIFFP,TOTPDENI,
     1 TOTDDC5,TOTPGRX,TOTPGROM,TOTOGRO,TOTOGRA,
     1 TOTOGRX,TOTDDC6,TOTONIT,TOTBPMIN,TOTBPDEN,
     1 TOTBPMAC,TOTBNMIN,TOTBNDEN,TOTNNIT,TOTBNMAC,
     1 TOTBNDIF,TOTINIT,TOTBIDEN,TOTBIDIF,TOTSMIN,
     1 TOTGROM,TOTFAGR,TOTFAM,TOTFBGR,TOTFBM,
     1 TOTZMRT,TOTDZM,
     1 TOTPGRO,TOTOSMIN,TOTOGROM,TOTBPDIF,TOTPGRA,
     1 TOTDMRTX,TOTXMRT,TOTDGRZ,TOTSMFI,TOTDETP,
     1 TOTETP,TOTSEQ
      SAVE TOTGRZ,TOTPRED,TOTASINKI,TOTASINKO,
     1 TOTGRW,TOTAGRZ,TOTDSINKI,TOTDSINKO,TOTDECOM,
     1 TOTDFEK,TOTDFI,TOTDDEN,TOTDPRED,
     1 TOTSSINKP,TOTSSINKD,TOTSSINKN,TOTSMACM,
     1 TOTSDENI,TOTDIFFN,TOTWDENI,TOTDDC4,TOTGRO,
     1 TOTGRA,TOTNSINKI,TOTNSINKO,TOTNGRA,TOTNGRAX,
     1 TOTXGRZ,TOTMGROM,TOTMACM,TOTDIFFP,TOTPDENI,
     1 TOTDDC5,TOTPGRX,TOTPGROM,TOTOGRO,TOTOGRA,
     1 TOTOGRX,TOTDDC6,TOTONIT,TOTBPMIN,TOTBPDEN,
     1 TOTBPMAC,TOTBNMIN,TOTBNDEN,TOTNNIT,TOTBNMAC,
     1 TOTBNDIF,TOTINIT,TOTBIDEN,TOTBIDIF,TOTSMIN,
     1 TOTGROM,TOTFAGR,TOTFAM,TOTFBGR,TOTFBM,
     1 TOTZMRT,TOTDZM,
     1 TOTPGRO,TOTOSMIN,TOTOGROM,TOTBPDIF,TOTPGRA,
     1 TOTDMRTX,TOTXMRT,TOTDGRZ,TOTSMFI,TOTDETP,
     1 TOTETP,TOTSEQ
      REAL FOXY,DININ,PO4IN
      SAVE FOXY,DININ,PO4IN

C     CHAPTER 1  DATA
      IF(ICHAP.GT.1) GOTO 120
C     CHAPTER 1  MODIFY DEFAULT DATA
C     GROUP 1
      N=26
      LSTEP=1000
      TFRAC(2)=600.
      TSDAY=INT(86400./TFRAC(2))
C     GROUP 2
```

```
      ZDIM=18.
      AREAHZ=5.5E6
      SAREAKM2=AREAHZ/1000000.
      INDARE=2
C     GROUP 3
      SOLVAR(JRHOU)=.TRUE.
      SOLVAR(JRHOV)=.TRUE.
      SOLVAR(JH)   =.TRUE.
      SOLVAR(JK)   =.TRUE.
      SOLVAR(JD)   =.TRUE.
      DO I=JZOO,JBNO
         SOLVAR(I) =.TRUE.
      ENDDO
C     GROUP 4
      RHOREF=999.975
C     GROUP 7
      CORI=1.2E-4
      BETA=1.6
      OPEN(UNIT=8,FILE='FLOW.DAT',FORM='FORMATTED',STATUS='OLD')
      OPEN(UNIT=10,FILE='METEO.DAT',FORM='FORMATTED',STATUS='OLD')
      OPEN(UNIT=7,FILE='BIOPAR.PAR',FORM='FORMATTED',STATUS='OLD')
      CALL BIOPAR(7)
      CLOSE(7)
      ETD=10.
      OPEN(UNIT=9,FILE='NUTIN.DAT',FORM='FORMATTED',STATUS='OLD')
C     GROUP 8
      VST2(JH)=16.76E6
      VST2(JK)=1.E-5
      VST2(JD)=5.E-8
      VST2(JPHY)=0.1
      VST2(JZOO)=VST2(JPHY)/700.
      VST2(JDET)=0.1
      VST2(JSED)=4.
      VST2(JDIN)=1.1
      VST2(JANF)=0.1
      VST2(JPO4)=0.01
      VST2(JOXY)=10.0
      VST2(JBIP)=0.3
      VST2(JBNH)=0.01
      VST2(JBNO)=0.01
      MACOFF=.TRUE.
C     GROUP 9
      IKBLZ(JRHOU)=1
      IKBLZ(JRHOV)=1
C     GROUP 11
      INIOUT=.FALSE.
      NSTAT=9000000
      NPROF=NSTAT
      ITPLOT=1
      OPEN(UNIT=15,FILE='RESULTF.DAT',STATUS='UNKNOWN')
      OPEN(UNIT=16,FILE='PHYSPRO.DAT',STATUS='UNKNOWN')
      OPEN(UNIT=17,FILE='NUTPRO.DAT',STATUS='UNKNOWN')
      OPEN(UNIT=18,FILE='FLOWZOO.DAT',STATUS='UNKNOWN')
      RETURN
120   CONTINUE

C     CHAPTER 2  TRANSIENT BOUNDARY CONDITIONS
      IF(ICHAP.GT.2)GOTO 130
      READ(10,*) ITIMESTEP,UG,VG,FH,FSB
      FLUXHZ(JRHOU)=-1.3E-3*UG*SQRT(UG**2+VG**2)
      FLUXHZ(JRHOV)=-1.3E-3*VG*SQRT(UG**2+VG**2)
      FLUXHZ(JH)=FH
      FLXRAD=FSB
```

```
      CALL OXYFLX(SQRT(UG**2+VG**2),FOXY)
      FLUXHZ(JOXY)=FOXY
      FLUXHZ(JDIN)=-NDEPPAR/1000./365./86400.
     1             -POINTDIN/SAREAKM2/1000./365./86400.
      FLUXHZ(JPO4)=-PDEPPAR/1000./365./86400.
     1             -POINTPO4/SAREAKM2/1000./365./86400.
      IF(MOD(ISTEP,TSDAY).EQ.0)THEN
         READ(8,*) IYMD,QROUT
         READ(9,*) IYMD,DININ,PO4IN
         QINFL(NM1)=QROUT
         QOUTFL(NM1)=QROUT
         PHIIN(NM1,JH)=F(NM1,JH)
         PHIIN(NM1,JDIN)=DININ
         PHIIN(NM1,JPO4)=PO4IN
      ENDIF
      RETURN
130   CONTINUE

C     CHAPTER 3  ADDITIONAL SOURCE TERMS
      IF(ICHAP.GT.3)GOTO 140
      IF(ISTEP.GT.0)THEN
         IF(J.GE.JZOO.AND.J.LE.JBNO)THEN
            CALL BIOLA(-FLXRAD,MACOFF,ETD,SAREAKM2,
     1 TOTGRZ,TOTPRED,TOTASINKI,TOTASINKO,TOTGRW,
     1 TOTAGRZ,TOTDSINKI,TOTDSINKO,TOTDECOM,TOTDGRZ,TOTDPRED,
     1 TOTDFEK,TOTDFI,TOTSMFI,TOTDDEN,TOTSSINKP,TOTSSINKD,TOTSSINKN,
     1 TOTSMACM,TOTSDENI,TOTSMIN,TOTDIFFN,TOTWDENI,TOTDDC4,TOTGRO,
     1 TOTGRA,TOTGROM,TOTNSINKI,TOTNSINKO,TOTNGRA,
     1 TOTNGRAX,TOTXGRZ,TOTMGROM,TOTMACM,TOTDIFFP,TOTPDENI,
     1 TOTDDC5,TOTPGRO,TOTPGRA,TOTPGRX,TOTPGROM,TOTOGRO,TOTOGRA,
     1 TOTOGRX,TOTOGROM,TOTOSMIN,TOTDDC6,TOTONIT,TOTBPMIN,
     1 TOTBPDEN,TOTBPMAC,TOTBPDIF,TOTBNMIN,TOTBNDEN,TOTNNIT,
     1 TOTBNMAC,TOTBNDIF,TOTINIT,TOTBIDEN,TOTBIDIF, TOTFAGR,
     1 TOTFAM,TOTFBGR,TOTFBM,TOTZMRT,TOTDZM,TOTDMRTX,TOTXMRT,
     1 TOTDETP,TOTETP,TOTSEQ)
         ENDIF
      ENDIF
      RETURN
140   CONTINUE

C     CHAPTER 4  ADDITIONAL OUTPUT
      IF(ISTEP.GT.0)THEN
         WRITE(15,*)ISTEP,Z(N),F(NM1,JTE)
         IF(MOD(ISTEP,TSDAY).EQ.0)THEN
            DO I=NM1,2,-1
               WRITE(16,*)IYMD,Z(I),F(I,JTE)
               WRITE(17,*)IYMD,Z(I),F(I,JDIN),F(I,JPO4)
            ENDDO
         ENDIF
         WRITE(18,*)ISTEP,TOTGRZ,TOTPRED,TOTZMRT
         TOTGRZ=0.0
         TOTPRED=0.0
         TOTZMRT=0.0
         RETURN
      ENDIF
      END
```

Appendix C. The code of BIOLA

BIOSO.FOR

The file BIOSO.FOR contains the subroutines OXYFLX, ETADEPTH, BIOLA, CHKMIN, ZOOPL, PHYPL, DETR, SEDORG, DISSNITR, CBACT, MACKOL, PHOS, OXYGEN, BPHOS, BAMMO, BNITR, FISHA and PISCI.

```
C*********************************************************************
      SUBROUTINE OXYFLX(WIND,FOT)
C*********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'BIOPAR.INC'

      REAL WIND,FOT,TEMPC,TEMPK,O2,A,B,O2S,SC,V
      DATA BUBBEL/0.025/

      TEMPC=F(NM1,JTE)
      TEMPK=F(NM1,JTE)+273.15

C     TRANSFORM FROM MG/L TO ML/L.
      O2=F(NM1,JOXY)*0.7

      IF (WIND.LE.3.6) THEN
        A=0.17
        B=0.
      ELSEIF (WIND.GT.3.6.AND.WIND.LE.13.) THEN
        A=2.85
        B=-9.65
      ELSE
        A=5.9
        B=-49.3
      ENDIF

      O2S=EXP(-173.4292+249.6339*100./TEMPK+
     &143.3483*LOG(TEMPK/100.)-21.8492*TEMPK/100.)
      SC=1450.+(-71.+1.1*TEMPC)*TEMPC
      V=5.9*(A*WIND+B)/SQRT(SC)/86400.
      FOT=V*(O2-O2S*(1.+BUBBEL))

C     FOT IS IN ML/L o S. TRANSFORM TO MG/L o S
      FOT=FOT/0.700

      RETURN
      END

C*********************************************************************
      SUBROUTINE ETADEPTH(AREA,Z,DZCELL,NIM,N,ETD)
C*********************************************************************
      INTEGER NIM, N, DINDEX
      REAL AREA(NIM),Z(NIM),DZCELL(NIM)
      REAL VOL,MEANDEPTH,PART,A,ETD

      VOL = 0.0
      DO I = 2, N-1
         VOL = VOL + AREA(I) * DZCELL(I)
      ENDDO
      MEANDEPTH = VOL / AREA(N-1)
      PART = 25.0*SQRT(AREA(N-1)/1.0E6)/MEANDEPTH*41.0**(0.061*
     1          MEANDEPTH/SQRT(AREA(N-1)/1.0E6))
```

```
      A = AREA(N-1)*(100.0-PART)/100.0
      DO I = 2, N-1
          IF (AREA(I).LT.A) THEN
          ELSE
              DINDEX = I
              EXIT
          ENDIF
      ENDDO
      ETD = Z(DINDEX)
      RETURN
      END
C*******************************************************************
      SUBROUTINE BIOLA(I0,MACOFF,ETD,SAREA,TOTGRZ,TOTPRED,
     1 TOTASINKI,TOTASINKO,TOTGRW,TOTAGRZ,TOTDSINKI,TOTDSINKO,
     1 TOTDECOM,TOTDGRZ,TOTDPRED,TOTDFEK,TOTDFI,TOTSMFI,TOTDDEN,
     1 TOTSSINKP,TOTSSINKD,TOTSSINKN,TOTSMACM,TOTSDENI,TOTSMIN,
     1 TOTDIFFN,TOTWDENI,TOTDDC4,TOTGRO,TOTGRA,TOTGROM,
     1 TOTNSINKI,TOTNSINKO,TOTNGRA,TOTNGRAX,TOTXGRZ,
     1 TOTMGROM,TOTMACM,TOTDIFFP,TOTPDENI,TOTDDC5,TOTPGRO,TOTPGRA,
     1 TOTPGRX,TOTPGROM,TOTOGRO,TOTOGRA,TOTOGRX,TOTOGROM,TOTOSMIN,
     1 TOTDDC6,TOTONIT,TOTBPMIN,TOTBPDEN,TOTBPMAC,TOTBPDIF,
     1 TOTBNMIN,TOTBNDEN,TOTNNIT,TOTBNMAC,TOTBNDIF,TOTINIT,
     1 TOTBIDEN,TOTBIDIF,TOTFAGR,TOTFAM,TOTFBGR,TOTFBM,TOTZMRT,
     1 TOTDZM,TOTDMRTX,TOTXMRT,TOTDETP,TOTETP,TOTSEQ)
C*******************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'

      INTEGER I
      LOGICAL MACOFF

      REAL I0
      REAL ETD,SAREA
      REAL TOTGRZ,TOTPRED,TOTASINKI,TOTASINKO,
     1 TOTGRW,TOTAGRZ,TOTDSINKI,TOTDSINKO,TOTDECOM,TOTDGRZ,TOTDPRED,
     1 TOTDFEK,TOTDFI,TOTSMFI,TOTDDEN,
     1 TOTSSINKP,TOTSSINKD,TOTSSINKN,TOTSMACM,TOTSDENI,TOTSMIN,
     1 TOTDIFFN,TOTWDENI,TOTDDC4,TOTGRO,TOTGRA,TOTGROM,
     1 TOTNSINKI,TOTNSINKO,TOTNGRA,TOTNGRAX,TOTXGRZ,
     1 TOTMGROM,TOTMACM,TOTDIFFP,TOTPDENI,TOTDDC5,TOTPGRO,TOTPGRA,
     1 TOTPGRX,TOTPGROM,TOTOGRO,TOTOGRA,TOTOGRX,TOTOGROM,TOTOSMIN,
     1 TOTDDC6,TOTONIT,TOTBPMIN,TOTBPDEN,TOTBPMAC,TOTBPDIF,
     1 TOTBNMIN,TOTBNDEN,TOTNNIT,TOTBNMAC,TOTBNDIF,
     1 TOTINIT,TOTBIDEN,TOTBIDIF,TOTFAGR,TOTFAM,TOTFBGR,TOTFBM
     1 TOTZMRT,TOTDZM,TOTDMRTX,TOTXMRT,TOTDETP,TOTETP,TOTSEQ
      REAL SAQIN(NIM),SAQOUT(NIM),SAQZ(NIM)
      SAVE SAQIN,SAQOUT,SAQZ
      REAL GRO(NIM),ZGRAZE(NIM),AGRAZE(NIM),
     1     DETGRZ(NIM),CBXGRZ(NIM),DETFEK(NIM)
      SAVE GRO,ZGRAZE,AGRAZE,DETGRZ,CBXGRZ,DETFEK
      REAL PREZ(NIM),PREFI(NIM),FMORT(NIM),GRA(NIM),GRAX(NIM)
      SAVE PREZ,PREFI,FMORT,GRA,GRAX
      REAL SDEN(NIM),WDEN(NIM),SMIN(NIM)
      SAVE SDEN,WDEN,SMIN
      REAL NIT(NIM),DIFP(NIM),DIFNH(NIM),DIFNO(NIM)
      SAVE NIT,DIFP,DIFNH,DIFNO
      REAL GROM(NIM),DDC(NIM),MACM(NIM),SEQ(NIM)
      SAVE GROM,DDC,MACM,SEQ
      REAL GMBN(NIM),GMDN(NIM),GMBP(NIM),GMDP(NIM)
      SAVE GMBN,GMDN,GMBP,GMDP
      REAL SINKOP(NIM),SINKOD(NIM),SINKOX(NIM)
      REAL SINKIP(NIM),SINKID(NIM)
```

```
       SAVE SINKOP,SINKOD,SINKOX,SINKIP,SINKID
       REAL SINKIX(NIM),SEDP(NIM),SEDD(NIM),SEDX(NIM),ETPROP(NIM)
       SAVE SINKIX,SEDP,SEDD,SEDX,ETPROP
       REAL ZMORT(NIM),MORTX(NIM)
       SAVE ZMORT,MORTX

      IF(J.EQ.JZOO) GOTO 11
      IF(J.EQ.JPHY) GOTO 12
      IF(J.EQ.JDET) GOTO 13
      IF(J.EQ.JSED) GOTO 14
      IF(J.EQ.JDIN) GOTO 15
      IF(J.EQ.JANF) GOTO 16
      IF(J.EQ.JMAC) GOTO 17
      IF(J.EQ.JPO4) GOTO 18
      IF(J.EQ.JOXY) GOTO 19
      IF(J.EQ.JBIP) GOTO 20
      IF(J.EQ.JBNH) GOTO 21
      IF(J.EQ.JBNO) GOTO 22
      IF(J.EQ.JFIA) GOTO 23
      IF(J.EQ.JFIB) GOTO 24
      RETURN
C----------------------------------------------------------------------
CHAPTER  A  A  A  A  A  A  PRELIMINARIES  A  A  A  A  A  A  A  A  A
   11 CONTINUE

C     SAVE QINFL,QOUTFL,QZ TEMPORARY
      DO I=1,N
          SAQIN(I)=QINFL(I)
          SAQOUT(I)=QOUTFL(I)
          SAQZ(I)=QZ(I)
      ENDDO

C      CALCULATE NUTRIENT AND CARBON PROCESS RATES
       CALL GROWTH(I0,GRO,MACOFF)
       CALL GRAZER(ZGRAZE,AGRAZE,DETGRZ,CBXGRZ,DETFEK)
       CALL PREDATZ(PREZ)
       CALL PREDATF(PREFI)
       CALL FISHMORT(FMORT)
       CALL CBGROWTH(I0,GRA,GRAX,MACOFF)
       CALL DENITR(SDEN,WDEN)
       CALL SEDMINE(SMIN,SDEN)
       CALL NITFIC(NIT)
       CALL ADSCOEFF(FAKTA)
       CALL SEDDIFF(DIFP,DIFNO,DIFNH,FAKTA)
       CALL MACROF(I0,GROM,GMBN,GMDN,GMBP,GMDP,MACOFF)
       CALL MACMORT(MACM,MACOFF)
       CALL DDCOMP(DDC)
       CALL SEQUEST(SEQ)
       CALL SINKPHY(SINKOP,SINKIP,SEDP)
       CALL SINKDET(SINKOD,SINKID,SEDD)
       CALL SINKCB(SINKOX,SINKIX,SEDX)
       CALL ZOOMORT(ZMORT)
       CALL CBMORT(MORTX)
       CALL ETADVECTION(ETD,WG,ETPROP)
       CALL CHKMIN(GRO,ZGRAZE,AGRAZE,DETGRZ,CBXGRZ,DETFEK,PREZ,PREFI,
     1 FMORT,GRA,GRAX,SDEN,WDEN,SMIN,NIT,DIFP,DIFNO,DIFNH,GROM,GMBN,
     1  GMDN,GMBP,GMDP,MACM,DDC,SINKOD,SINKID,SEDD,SINKOP,SINKIP,
     1 SEDP,SEQ,SINKOX,SINKIX,SEDX,ZMORT,MORTX,ETPROP)
C----------------------------------------------------------------------
CHAPTER  B  B  B  B  B  B  ZOOPLANKTON  B  B  B  B  B  B  B  B  B
       CALL ZOOPL(ZGRAZE,PREZ,ZMORT,TOTGRZ,TOTPRED,TOTZMRT)
       RETURN
```

```
C---------------------------------------------------------------------
CHAPTER  C  C  C  C  C  C  PHYTOPLANKTON  C  C  C  C  C  C  C  C  C
   12 CONTINUE
      CALL PHYPL(GRO,AGRAZE,SINKOP,SINKIP,
     1 TOTASINKI,TOTGRW,TOTASINKO,TOTAGRZ)
      RETURN
C---------------------------------------------------------------------
CHAPTER  D  D  D  D  D  D  DETRITUS  D  D  D  D  D  D  D  D  D  D  D
   13 CONTINUE
      CALL DETR(DETFEK,DETGRZ,PREZ,PREFI,ZMORT,WDEN,DDC,SINKOD,
     1 SINKID,MORTX,ETPROP,TOTDSINKI,TOTDPRED,TOTDFEK,TOTDFI,TOTDDEN,
     1 TOTDSINKO,TOTDECOM,TOTDGRZ,TOTDZM,TOTDMRTX,TOTDETP)
      RETURN
C---------------------------------------------------------------------
CHAPTER  E  E  E  E  E  E  SEDIMENT ORGANIC MATTER  E  E  E  E  E  E
   14 CONTINUE
      CALL SEDORG(SDEN,SMIN,MACM,SEQ,SEDP,SEDD,SEDX,ETPROP,FMORT,
     1 TOTSSINKP,TOTSSINKD,TOTSSINKN,TOTSMACM,TOTSDENI,TOTSMIN,
     1 TOTSMFI,TOTETP,TOTSEQ)
C     EXCLUDE ADVECTION AND DIFFUSION
      DO I=1,N
          QINFL(I)=0.
          QOUTFL(I)=0.
          QZ(I)=0.
          DIF(I)=0.
      ENDDO
      RETURN
C---------------------------------------------------------------------
CHAPTER  F  F  F  F  F  F  DISSOLVED INORGANIC NITROGEN  F  F  F  F
   15 CONTINUE
      CALL DISSNITR(WDEN,GRO,GRA,DIFNO,DIFNH,GMDN,DDC,
     1 TOTDIFFN,TOTWDENI,TOTDDC4,TOTGRO,TOTGRA,TOTGROM)
C     INCLUDE ADVECTION AGAIN
      DO I=1,N
          QINFL(I)=SAQIN(I)
          QOUTFL(I)=SAQOUT(I)
          QZ(I)=SAQZ(I)
      ENDDO
      RETURN
C---------------------------------------------------------------------
CHAPTER  G  G  G  G  G  G  CYANOBACTERIA  G  G  G  G  G  G  G  G  G
   16 CONTINUE
      CALL CBACT(GRA,GRAX,CBXGRZ,SINKOX,SINKIX,MORTX,TOTNSINKI,
     1   TOTNSINKO,TOTNGRA,TOTNGRAX,TOTXGRZ,TOTXMRT)
      RETURN
C---------------------------------------------------------------------
CHAPTER  H  H  H  H  H  H  MACROPHYTES  H  H  H  H  H  H  H  H  H  H
   17 CONTINUE
      CALL MACKOL(GROM,MACM,TOTMGROM,TOTMACM,MACOFF)
C     EXCLUDE ADVECTION AND DIFFUSION
      DO I=1,N
          QINFL(I)=0.
          QOUTFL(I)=0.
          QZ(I)=0.
          DIF(I)=0.
      ENDDO
      RETURN
C---------------------------------------------------------------------
CHAPTER  I  I  I  I  I  I  PHOSPHATE  I  I  I  I  I  I  I  I  I  I
   18 CONTINUE
      CALL PHOS(GRO,GRA,GRAX,WDEN,DIFP,GMDP,DDC,TOTDIFFP,
     1 TOTPDENI,TOTDDC5,TOTPGRO,TOTPGRA,TOTPGRX,TOTPGROM)
C     INCLUDE ADVECTION AGAIN
```

```
      DO I=1,N
          QINFL(I)=SAQIN(I)
          QOUTFL(I)=SAQOUT(I)
          QZ(I)=SAQZ(I)
      ENDDO
      RETURN
C-----------------------------------------------------------------------
CHAPTER  J  J  J  J  J  J  OXYGEN  J  J  J  J  J  J  J  J  J  J  J
   19 CONTINUE
      CALL OXYGEN(GRO,GRA,GRAX,SMIN,NIT,GROM,DDC,
     1 TOTOGRO,TOTOGRA,TOTOGRX,TOTOGROM,TOTOSMIN,TOTDDC6,TOTONIT)
      RETURN
C-----------------------------------------------------------------------
CHAPTER  K  K  K  K  K  K  BENTHIC PHOSPHORUS  K  K  K  K  K  K  K
   20 CONTINUE
      CALL BPHOS(SMIN,SDEN,DIFP,GMBP,TOTBPMIN,TOTBPDEN,
     1 TOTBPMAC,TOTBPDIF,SAREA)
C     EXCLUDE ADVECTION AND DIFFUSION
      DO I=1,N
          QINFL(I)=0.
          QOUTFL(I)=0.
          QZ(I)=0.
          DIF(I)=0.
       ENDDO
      RETURN
C-----------------------------------------------------------------------
CHAPTER  L  L  L  L  L  L  BENTHIC AMMONIUM  L  L  L  L  L  L  L  L
   21 CONTINUE
      CALL BAMMO(SMIN,SDEN,NIT,DIFNH,GMBN,
     1 TOTBNMIN,TOTBNDEN,TOTNNIT,TOTBNMAC,TOTBNDIF)
C      EXCLUDE DIFFUSION, NEEDS TO BE DONE EVERY J
       DO I=1,N
          DIF(I)=0.
       ENDDO
      RETURN
C-----------------------------------------------------------------------
CHAPTER  M  M  M  M  M  M  BENTHIC NITRATE  M  M  M  M  M  M  M  M
   22 CONTINUE
      CALL BNITR(SDEN,NIT,DIFNO,TOTINIT,TOTBIDEN,TOTBIDIF)
C      EXCLUDE DIFFUSION, NEEDS TO BE DONE EVERY J
       DO I=1,N
          DIF(I)=0.
       ENDDO
      RETURN
C-----------------------------------------------------------------------
CHAPTER  N  N  N  N  N  N  PLANKTIVOROUS FISH  N  N  N  N  N  N  N
   23 CONTINUE
      CALL FISHA(PREZ,PREFI,TOTFAGR,TOTFAM)
      RETURN
C-----------------------------------------------------------------------
CHAPTER  O  O  O  O  O  O  PISCIVOROUS FISH  O  O  O  O  O  O  O  O
   24 CONTINUE
      CALL PISCI(PREFI,FMORT,TOTFBGR,TOTFBM)
      RETURN
      END
```

```
C*********************************************************************
      SUBROUTINE CHKMIN(GRO,ZGRAZE,AGRAZE,DETGRZ,CBXGRZ,DETFEK,PREZ,
     1 PREFI,FMORT,GRA,GRAX,SDEN,WDEN,SMIN,NIT,DIFP,DIFNO,DIFNH,GROM
     1  ,GMBN,GMDN,GMBP,GMDP,MACM,DDC,SINKOD,SINKID,SEDD,SINKOP,
     1 SINKIP,SEDP,SEQ,SINKOX,SINKIX,SEDX,ZMORT,MORTX,ETPROP)
C*********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'

       INTEGER I
       REAL TESTMIN
       REAL GRO(NIM),ZGRAZE(NIM),AGRAZE(NIM),DETGRZ(NIM),CBXGRZ(NIM)
       REAL DETFEK(NIM),PREZ(NIM),PREFI(NIM),FMORT(NIM),GRA(NIM)
       REAL GRAX(NIM),ZMORT(NIM),MORTX(NIM)
       REAL SDEN(NIM),WDEN(NIM),SMIN(NIM),NIT(NIM),DIFP(NIM)
       REAL DIFNO(NIM),DIFNH(NIM),GROM(NIM),GMBN(NIM),GMDN(NIM)
       REAL GMBP(NIM),GMDP(NIM),MACM(NIM),DDC(NIM)
       REAL SINKOD(NIM),SINKOP(NIM),SINKIP(NIM),SEDP(NIM)
       REAL SINKID(NIM),SEDD(NIM),SEQ(NIM),SINKOX(NIM),SINKIX(NIM)
       REAL SEDX(NIM),ETPROP(NIM)

C      CHECK EVERY VARIABLE FOR SINKS THAT REDUCES IT BELOW MINIMUM
C      VALUE
      DO I=2,NM1
C
C      ZOOPLANKTON
       TESTMIN=F(I,JZOO)-PREZ(I)*DT-ZMORT(I)*DT
       IF (TESTMIN.LT.ZOOMIN) THEN
         TESTMIN=TESTMIN+PREZ(I)*DT
         PREZ(I)=0.
         IF (TESTMIN.LT.ZOOMIN) THEN
           ZMORT(I)=0.
         ENDIF
       ENDIF
C
C      PHYTOPLANKTON
       TESTMIN=F(I,JPHY)-AGRAZE(I)*DT-SINKOP(I)*DT
       IF (TESTMIN.LT.PHYMIN) THEN
         TESTMIN=TESTMIN+AGRAZE(I)*DT
         ZGRAZE(I)=ZGRAZE(I)-EFFI*AGRAZE(I)
         DETFEK(I)=(1-EFFI)/EFFI*ZGRAZE(I)
         AGRAZE(I)=0.
         IF (TESTMIN.LT.PHYMIN) THEN
           SINKOP(I)=0.
           SINKIP(I-1)=0.
           SEDP(I)=0.
         ENDIF
       ENDIF
C
C      DETRITUS
       TESTMIN=F(I,JDET)-SINKOD(I)*DT-DDC(I)*DT
     & -DETGRZ(I)*DT-WDEN(I)*CCNDENIT*DT
       IF (TESTMIN.LT.DETMIN) THEN
         TESTMIN=TESTMIN+WDEN(I)*CCNDENIT*DT
         WDEN(I)=0.
         IF (TESTMIN.LT.DETMIN) THEN
           ZGRAZE(I)=ZGRAZE(I)-EFFI*DETGRZ(I)
           DETFEK(I)=(1-EFFI)/EFFI*ZGRAZE(I)
           TESTMIN=TESTMIN+DETGRZ(I)*DT
           DETGRZ(I)=0.
           IF (TESTMIN.LT.DETMIN) THEN
             TESTMIN=TESTMIN+DDC(I)*DT
             DDC(I)=0.
```

```
            IF (TESTMIN.LT.DETMIN) THEN
              SINKOD(I)=0.
              SINKID(I-1)=0.
              SEDD(I)=0.
            ENDIF
          ENDIF
        ENDIF
      ENDIF
C
C     SEDIMENT ORGANIC MATTER
      TESTMIN=F(I,JSED)-SDEN(I)*CCNDENIT*DT-SMIN(I)*DT-
     &  SEQ(I)*DT-ETPROP(I)*F(I,JSED)*DT
      IF (TESTMIN.LT.SEDMIN) THEN
        TESTMIN=TESTMIN+SMIN(I)*DT
        SMIN(I)=0.
        IF (TESTMIN.LT.SEDMIN) THEN
          TESTMIN=TESTMIN+ETPROP(I)*F(I,JSED)*DT
          ETPROP(I)=0.
          IF (TESTMIN.LT.SEDMIN) THEN
            TESTMIN=TESTMIN+SEQ(I)*DT
            SEQ(I)=0.
            IF (TESTMIN.LT.SEDMIN) THEN
              SDEN(I)=0.
            ENDIF
          ENDIF
        ENDIF
      ENDIF
C
C     DISSOLVED NITROGEN
      IF((DIFNO(I).LE.0.).AND.(DIFNH(I).LE.0.))THEN
        TESTMIN=F(I,JDIN)-GRO(I)*CNC*DT-WDEN(I)*DT-GRA(I)*CNC*DT-
     & GMDN(I)*DT+(DIFNO(I)+DIFNH(I))*
     & (AREA(I)-AREA(I-1))/AREA(I)/DZCELL(I)*DT
      ELSEIF((DIFNO(I).GT.0.).AND.(DIFNH(I).LT.0.))THEN
        TESTMIN=F(I,JDIN)-GRO(I)*CNC*DT-WDEN(I)*DT-GRA(I)*CNC*DT-
     & GMDN(I)*DT+DIFNH(I)*(AREA(I)-AREA(I-1))/AREA(I)/DZCELL(I)*DT
      ELSEIF((DIFNO(I).LT.0.).AND.(DIFNH(I).GT.0.))THEN
        TESTMIN=F(I,JDIN)-GRO(I)*CNC*DT-WDEN(I)*DT-GRA(I)*CNC*DT-
     & GMDN(I)*DT+DIFNO(I)*(AREA(I)-AREA(I-1))/AREA(I)/DZCELL(I)*DT
      ELSE
        TESTMIN=F(I,JDIN)-GRO(I)*CNC*DT-WDEN(I)*DT-GRA(I)*CNC*DT
     & -GMDN(I)*DT
      ENDIF
      IF (TESTMIN.LT.DINMIN) THEN
        TESTMIN=TESTMIN+WDEN(I)*DT
        WDEN(I)=0.
        IF (TESTMIN.LT.DINMIN) THEN
          IF (GMDN(I).GT.0.0) THEN
            TESTMIN=TESTMIN+GMDN(I)*DT
            GMDN(I)=0.
            GMBN(I)=0.
            GMDP(I)=0.
            GMBP(I)=0.
            GROM(I)=0.
          ENDIF
          IF (TESTMIN.LT.DINMIN) THEN
            TESTMIN=TESTMIN+GRA(I)*CNC*DT
            GRA(I)=0.
            IF (TESTMIN.LT.DINMIN) THEN
              TESTMIN=TESTMIN+GRO(I)*CNC*DT
              GRO(I)=0.
              IF (TESTMIN.LT.DINMIN) THEN
                DIFNO(I)=0.
```

```
                DIFNH(I)=0.
              ENDIF
            ENDIF
          ENDIF
        ENDIF
      ENDIF
C
C     NITROGEN FIXATING ALGAE
      TESTMIN=F(I,JANF)-SINKOX(I)*DT-CBXGRZ(I)*DT-MORTX(I)*DT
      IF (TESTMIN.LT.ANFMIN) THEN
        TESTMIN=TESTMIN+CBXGRZ(I)*DT
        ZGRAZE(I)=ZGRAZE(I)-EFFI*CBXGRZ(I)
        DETFEK(I)=(1-EFFI)/EFFI*ZGRAZE(I)
        CBXGRZ(I)=0.
        IF (TESTMIN.LT.ANFMIN) THEN
          TESTMIN=TESTMIN+MORTX(I)*DT
          MORTX(I)=0.
          IF (TESTMIN.LT.ANFMIN) THEN
            SINKOX(I)=0.
            SINKIX(I-1)=0.
            SEDX(I)=0.
          ENDIF
        ENDIF
      ENDIF
C
C     MACROPHYTES
C     HAVE ONLY ONE SINK, WHICH IS CHECKED WHEN CALCULATED
C
C     PHOSPHATE
      IF(DIFP(I).LT.0.)THEN
        TESTMIN=F(I,JPO4)-GRO(I)*CPC*DT+
     &    DIFP(I)*(AREA(I)-AREA(I-1))/AREA(I)/DZCELL(I)*DT
     &  -GRA(I)*CPC*DT-GRAX(I)*CPC*DT-GMDP(I)*DT
      ELSE
       TESTMIN=F(I,JPO4)-GRO(I)*CPC*DT
     &  -GRA(I)*CPC*DT-GRAX(I)*CPC*DT-GMDP(I)*DT
      ENDIF
      IF (TESTMIN.LT.PO4MIN) THEN
        IF (GMDP(I).GT.0.0) THEN
          TESTMIN=TESTMIN+GMDP(I)*DT
          GMDP(I)=0.
          GMBP(I)=0.
          GMDN(I)=0.
          GMBN(I)=0.
          GROM(I)=0.
        ENDIF
        IF (TESTMIN.LT.PO4MIN) THEN
          TESTMIN=TESTMIN+GRAX(I)*CPC*DT
          GRAX(I)=0.
          IF (TESTMIN.LT.PO4MIN) THEN
            TESTMIN=TESTMIN+GRA(I)*CPC*DT
            GRA(I)=0.
            IF (TESTMIN.LT.PO4MIN) THEN
              TESTMIN=TESTMIN+GRO(I)*CPC*DT
              GRO(I)=0.
              IF (TESTMIN.LT.PO4MIN) THEN
                DIFP(I)=0.
              ENDIF
            ENDIF
          ENDIF
        ENDIF
      ENDIF
C
```

```
C       OXYGEN
       TESTMIN=F(I,JOXY)-DDC(I)*COC*DT
     &   -SMIN(I)*COC*(AREA(I)-AREA(I-1))/AREA(I)/DZCELL(I)*DT-
     &    NIT(I)*CONNITR*(AREA(I)-AREA(I-1))/AREA(I)/DZCELL(I)*DT
        IF (TESTMIN.LT.OXYMIN) THEN
          TESTMIN=TESTMIN+NIT(I)*CONNITR*(AREA(I)-AREA(I-1))
     1    /AREA(I)/DZCELL(I)*DT
          NIT(I)=0.
          IF (TESTMIN.LT.OXYMIN) THEN
            TESTMIN=TESTMIN+SMIN(I)*COC*(AREA(I)-AREA(I-1))
     1      /AREA(I)/DZCELL(I)*DT
            SMIN(I)=0.
            IF (TESTMIN.LT.OXYMIN) THEN
              DDC(I)=0.
            ENDIF
          ENDIF
        ENDIF
C
C       BENTHIC PHOSPHATE
        IF(DIFP(I).GT.0.)THEN
          TESTMIN=F(I,JBIP)-DIFP(I)*DT-GMBP(I)*DT
        ELSE
          TESTMIN=F(I,JBIP)-GMBP(I)*DT
        ENDIF
        IF (TESTMIN.LT.BIPMIN) THEN
          IF (GMBP(I).GT.0.0) THEN
            TESTMIN=TESTMIN+GMBP(I)*DT
            GMBP(I)=0.
            GMDP(I)=0.
            GMBN(I)=0.
            GMDN(I)=0.
            GROM(I)=0.
          ENDIF
          IF (TESTMIN.LT.BIPMIN) THEN
            DIFP(I)=0.
          ENDIF
        ENDIF
C
C       BENTIC AMMONIUM
        IF(DIFNH(I).GT.0.)THEN
          TESTMIN=F(I,JBNH)-NIT(I)*DT-DIFNH(I)*DT-GMBN(I)*DT
        ELSE
          TESTMIN=F(I,JBNH)-NIT(I)*DT-GMBN(I)*DT
        ENDIF
        IF (TESTMIN.LT.BNHMIN) THEN
          IF (GMBN(I).GT.0.0) THEN
            TESTMIN=TESTMIN+GMBN(I)*DT
            GMBN(I)=0.
            GMDN(I)=0.
            GMBP(I)=0.
            GMDP(I)=0.
            GROM(I)=0.
          ENDIF
          IF (TESTMIN.LT.BNHMIN) THEN
            TESTMIN=TESTMIN+NIT(I)*DT
            NIT(I)=0.
            IF (TESTMIN.LT.BNHMIN) THEN
              DIFNH(I)=0.
            ENDIF
          ENDIF
        ENDIF

C
```

```
C     SEDIMENT NITRATE
       IF(DIFNO(I).GT.0.) THEN
         TESTMIN=F(I,JBNO)-SDEN(I)*DT-DIFNO(I)*DT
       ELSE
         TESTMIN=F(I,JBNO)-SDEN(I)*DT
       ENDIF
       IF (TESTMIN.LT.BNOMIN) THEN
         TESTMIN=TESTMIN+SDEN(I)*DT
         SDEN(I)=0.
         IF (TESTMIN.LT.BNOMIN) THEN
           DIFNO(I)=0.
         ENDIF
       ENDIF

      ENDDO
      RETURN
      END

C*********************************************************************
      SUBROUTINE ZOOPL(ZGRAZE,PREZ,ZMORT,TOTGRZ,TOTPRED,TOTZMRT)
C*********************************************************************
       INCLUDE 'comp97.inc'
       INCLUDE 'biopar.inc'
       INTEGER I
       REAL ZGRAZE(NIM),PREZ(NIM),ZMORT(NIM),TOTGRZ,TOTPRED,TOTZMRT

C-----ASSIGN VALUES TO SI AND SIP
      DO I=2,NM1
          SI(I)=ZGRAZE(I)
          SIP(I)=-PREZ(I)/F(I,JZOO)-ZMORT(I)/F(I,JZOO)
C-----VERTICAL INTEGRATION OF PROCESSES
          TOTGRZ=TOTGRZ+ZGRAZE(I)*DZCELL(I)*AREA(I)*DT
          TOTPRED=TOTPRED+PREZ(I)*DZCELL(I)*AREA(I)*DT
          TOTZMRT=TOTZMRT+ZMORT(I)*DZCELL(I)*AREA(I)*DT
      ENDDO
      RETURN
      END

C*********************************************************************
      SUBROUTINE PHYPL(GRO,AGRAZE,SINKOP,SINKIP,
     1 TOTASINKI,TOTGRW,TOTASINKO,TOTAGRZ)
C*********************************************************************
       INCLUDE 'comp97.inc'
       INCLUDE 'biopar.inc'
      INTEGER I
      REAL GRO(NIM),AGRAZE(NIM),TOTASINKI,TOTGRW,TOTASINKO,TOTAGRZ
      REAL SINKOP(NIM),SINKIP(NIM)

C-----ASSIGN VALUES TO SI AND SIP
      DO I=2,NM1
          SI(I)=SINKIP(I)+GRO(I)
          SIP(I)=-SINKOP(I)/F(I,JPHY)-AGRAZE(I)/F(I,JPHY)
C-----VERTICAL INTEGRATION OF PROCESSES
          TOTASINKI=TOTASINKI+SINKIP(I)*DZCELL(I)*DT*AREA(I)
          TOTGRW=TOTGRW+GRO(I)*DZCELL(I)*AREA(I)*DT
          TOTASINKO=TOTASINKO+SINKOP(I)*DZCELL(I)*DT*AREA(I)
          TOTAGRZ=TOTAGRZ+AGRAZE(I)*DZCELL(I)*AREA(I)*DT
      ENDDO
      RETURN
      END
```

```
C*********************************************************************
      SUBROUTINE DETR(DETFEK,DETGRZ,PREZ,PREFI,ZMORT,WDEN,DDC,
     1 SINKOD,SINKID,MORTX,ETPROP,TOTDSINKI,TOTDPRED,TOTDFEK,TOTDFI,
     1 TOTDDEN,TOTDSINKO,TOTDECOM,TOTDGRZ,TOTDZM,TOTDMRTX,TOTDETP)
C*********************************************************************
       INCLUDE 'comp97.inc'
       INCLUDE 'biopar.inc'
       INTEGER I
       REAL SINKID(NIM),SINKOD(NIM)
       REAL DDC(NIM),WDEN(NIM),ZMORT(NIM),MORTX(NIM),ETPROP(NIM)
       REAL DETFEK(NIM),DETGRZ(NIM),PREZ(NIM),PREFI(NIM)
       REAL TOTDSINKI,TOTDPRED,TOTDFEK,TOTDFI,TOTDDEN
       REAL TOTDSINKO,TOTDECOM,TOTDGRZ,TOTDZM,TOTDMRTX,TOTDETP

C-----ASSIGN VALUES TO SI AND SIP
      DO I=2,NM1
          SI(I)=SINKID(I)+DETFEK(I)+(1-OMEGA)*PREZ(I)+MORTX(I)+
     &        (1-OMEGA)*PREFI(I)+ZMORT(I)-DDC(I)-DETGRZ(I)
          SIP(I)=-SINKOD(I)/F(I,JDET)+ETPROP(I)*F(I,JSED)
     &        /F(I,JDET)*(AREA(I)-AREA(I-1))/(AREA(I)*DZCELL(I))
     &        -WDEN(I)*CCNDENIT/F(I,JDET)
C-----VERTICAL INTEGRATION OF PROCESSES
          TOTDSINKI=TOTDSINKI+SINKID(I)*DZCELL(I)*DT*AREA(I)
          TOTDPRED=TOTDPRED+(1-OMEGA)*PREZ(I)*DZCELL(I)*DT*AREA(I)
          TOTDFEK=TOTDFEK+DETFEK(I)*DZCELL(I)*DT*AREA(I)
          TOTDFI=TOTDFI+(1-OMEGA)*PREFI(I)*DZCELL(I)*DT*AREA(I)
          TOTDDEN=TOTDDEN+WDEN(I)*CCNDENIT*DZCELL(I)*DT*AREA(I)
          TOTDSINKO=TOTDSINKO+SINKOD(I)*DZCELL(I)*DT*AREA(I)
          TOTDECOM=TOTDECOM+DDC(I)*DZCELL(I)*DT*AREA(I)
          TOTDGRZ=TOTDGRZ+DETGRZ(I)*DZCELL(I)*DT*AREA(I)
          TOTDZM=TOTDZM+ZMORT(I)*DZCELL(I)*DT*AREA(I)
          TOTDMRTX=TOTDMRTX+MORTX(I)*DZCELL(I)*DT*AREA(I)
          TOTDETP=TOTDETP+ETPROP(I)*F(I,JSED)*(AREA(I)-AREA(I-1))*DT
       ENDDO
       RETURN
       END

C*********************************************************************
      SUBROUTINE SEDORG(SDEN,SMIN,MACM,SEQ,SEDP,SEDD,SEDX,ETPROP,
     1 FMORT,TOTSSINKP,TOTSSINKD,TOTSSINKN,TOTSMACM,TOTSDENI,
     1 TOTSMIN,TOTSMFI,TOTETP,TOTSEQ)
C*********************************************************************
       INCLUDE 'comp97.inc'
       INCLUDE 'biopar.inc'
       INTEGER I
       REAL SDEN(NIM),SMIN(NIM),MACM(NIM),SEQ(NIM),
     & SEDP(NIM),SEDD(NIM),SEDX(NIM),FMORT(NIM),ETPROP(NIM)
       REAL TOTSSINKP,TOTSSINKD, TOTSSINKN, TOTSMACM,
     & TOTSDENI,TOTSMIN,TOTSMFI,TOTETP,TOTSEQ

C-----ASSIGN VALUES TO SI AND SIP
      DO I=2,NM1
          SI(I)=SEDP(I)+SEDD(I)+SEDX(I)+MACM(I)*(ZDIM-ZBOUND(I-1))
     &        +FMORT(I)*DZCELL(I)*AREA(I)/(AREA(I)-AREA(I-1))
          SIP(I)=-SMIN(I)/F(I,JSED)-ETPROP(I)
     &        -SEQ(I)/F(I,JSED)
     &        -SDEN(I)*CCNDENIT/F(I,JSED)
C-----VERTICAL INTEGRATION OF PROCESSES
          TOTSSINKP=TOTSSINKP+SEDP(I)*(AREA(I)-AREA(I-1))*DT
          TOTSSINKD=TOTSSINKD+SEDD(I)*(AREA(I)-AREA(I-1))*DT
          TOTSSINKN=TOTSSINKN+SEDX(I)*(AREA(I)-AREA(I-1))*DT
          TOTSMACM=TOTSMACM+MACM(I)*(ZDIM-ZBOUND(I-1))*
     &            (AREA(I)-AREA(I-1))*DT
```

```fortran
         TOTSDENI=TOTSDENI+SDEN(I)*CCNDENIT*(AREA(I)-AREA(I-1))*DT
         TOTSMIN=TOTSMIN+SMIN(I)*(AREA(I)-AREA(I-1))*DT
         TOTSMFI=TOTSMFI+FMORT(I)*DZCELL(I)*DT*AREA(I)
         TOTETP=TOTETP+ETPROP(I)*F(I,JSED)*(AREA(I)-AREA(I-1))*DT
         TOTSEQ=TOTSEQ+SEQ(I)*(AREA(I)-AREA(I-1))*DT
      ENDDO
      RETURN
      END

C*******************************************************************
      SUBROUTINE DISSNITR(WDEN,GRO,GRA,DIFNO,DIFNH,GMDN,DDC,
     1 TOTDIFFN,TOTWDENI,TOTDDC4,TOTGRO,TOTGRA,TOTGROM)
C*******************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL WDEN(NIM),GRO(NIM),GRA(NIM)
      REAL DIFNO(NIM),DIFNH(NIM),GMDN(NIM)
      REAL DDC(NIM)
      REAL TOTDIFFN,TOTWDENI,TOTDDC4,TOTGRO,TOTGRA,TOTGROM

C-----ASSIGN VALUES TO SI AND SIP
      DO I=2,NM1
         SI(I)=DDC(I)*CNC+
     &         WDEN(I)*CCNDENIT*CNC
         SIP(I)=-GRO(I)*CNC/F(I,JDIN)-WDEN(I)/F(I,JDIN)
     &          -GRA(I)*CNC/F(I,JDIN)-GMDN(I)/F(I,JDIN)
         IF(DIFNO(I)+DIFNH(I).GT.0.)THEN
            SI(I)=SI(I)+(DIFNO(I)+DIFNH(I))*(AREA(I)-AREA(I-1))/
     &         AREA(I)/DZCELL(I)
         ELSE
            SIP(I)=SIP(I)+(DIFNO(I)+DIFNH(I))*(AREA(I)-AREA(I-1))/
     &         AREA(I)/DZCELL(I)/F(I,JDIN)
         ENDIF
C-----VERTICAL INTEGRATION OF PROCESSES
         TOTDIFFN=TOTDIFFN+(DIFNO(I)+DIFNH(I))*
     1           (AREA(I)-AREA(I-1))*DT
         TOTWDENI=TOTWDENI+WDEN(I)*CCNDENIT*CNC*DZCELL(I)*
     1            DT*AREA(I)-WDEN(I)*DZCELL(I)*DT*AREA(I)
         TOTDDC4=TOTDDC4+DDC(I)*CNC*DZCELL(I)*DT*AREA(I)
         TOTGRO=TOTGRO+GRO(I)*CNC*DZCELL(I)*DT*AREA(I)
         TOTGRA=TOTGRA+GRA(I)*CNC*DZCELL(I)*DT*AREA(I)
         TOTGROM=TOTGROM+GMDN(I)*DZCELL(I)*DT*AREA(I)
      ENDDO
      RETURN
      END

C*******************************************************************
      SUBROUTINE CBACT(GRA,GRAX,CBXGRZ,SINKOX,SINKIX,MORTX,
     1 TOTNSINKI,TOTNSINKO,TOTNGRA,TOTNGRAX,TOTXGRZ,TOTXMRT)
C*******************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL SINKIX(NIM),SINKOX(NIM),MORTX(NIM)
      REAL GRA(NIM),CBXGRZ(NIM),GRAX(NIM)
      REAL TOTNSINKI,TOTNSINKO,TOTNGRA,TOTNGRAX,TOTXGRZ,TOTXMRT

C-----ASSIGN VALUES TO SI AND SIP
      DO I=2,NM1
         SI(I)=SINKIX(I)+GRA(I)+GRAX(I)
         SIP(I)=-SINKOX(I)/F(I,JANF)-CBXGRZ(I)/F(I,JANF)
     1   -MORTX(I)/F(I,JANF)
```

```
C-----VERTICAL INTEGRATION OF PROCESSES
          TOTNSINKI=TOTNSINKI+SINKIX(I)*DZCELL(I)*DT*AREA(I)
          TOTNSINKO=TOTNSINKO+SINKOX(I)*DZCELL(I)*DT*AREA(I)
          TOTNGRA=TOTNGRA+GRA(I)*DZCELL(I)*DT*AREA(I)
          TOTNGRAX=TOTNGRAX+GRAX(I)*DZCELL(I)*DT*AREA(I)
          TOTXGRZ=TOTXGRZ+CBXGRZ(I)*DZCELL(I)*DT*AREA(I)
          TOTXMRT=TOTXMRT+MORTX(I)*DZCELL(I)*DT*AREA(I)
       ENDDO
      RETURN
      END

C*******************************************************************
      SUBROUTINE MACKOL(GROM,MACM,TOTMGROM,TOTMACM,MACOFF)
C*******************************************************************
       INCLUDE 'comp97.inc'
       INCLUDE 'biopar.inc'
       INTEGER I
       REAL GROM(NIM),MACM(NIM),TOTMGROM,TOTMACM
       LOGICAL MACOFF

C-----ASSIGN VALUES TO SI AND SIP
      IF(MACOFF) GOTO 50
      DO I=2,NM1
          SI(I)=GROM(I)
          SIP(I)=-MACM(I)/F(I,JMAC)
C-----VERTICAL INTEGRATION OF PROCESSES
          TOTMGROM=TOTMGROM+GROM(I)*DZCELL(I)*DT*AREA(I)
          TOTMACM=TOTMACM+MACM(I)*DZCELL(I)*DT*AREA(I)
      ENDDO
      RETURN
50    DO I=2,NM1
          SI(I)=0.
          SIP(I)=0.
       ENDDO
       TOTMGROM=0.
       TOTMACM=0.
       RETURN
       END

C*******************************************************************
      SUBROUTINE PHOS(GRO,GRA,GRAX,WDEN,DIFP,GMDP,DDC,
     1 TOTDIFFP,TOTPDENI,TOTDDC5,TOTPGRO,TOTPGRA,TOTPGRX,TOTPGROM)
C*******************************************************************
       INCLUDE 'comp97.inc'
       INCLUDE 'biopar.inc'
       INTEGER I
       REAL DDC(NIM),GRAX(NIM)
       REAL GRO(NIM),GRA(NIM),WDEN(NIM),DIFP(NIM),GMDP(NIM)
       REAL TOTDIFFP,TOTPDENI,TOTDDC5,TOTPGRO,TOTPGRA,TOTPGRX
       REAL TOTPGROM

C-----ASSIGN VALUES TO SI AND SIP
      DO I=2,NM1
          SI(I)=DDC(I)*CPC+
     &          WDEN(I)*CCNDENIT*CPC
          SIP(I)=-GRO(I)*CPC/F(I,JPO4) -GRA(I)*CPC/F(I,JPO4)-
     &          GRAX(I)*CPC/F(I,JPO4)-GMDP(I)/F(I,JPO4)
          IF(DIFP(I).GT.0.)THEN
             SI(I)=SI(I)+DIFP(I)*(AREA(I)-AREA(I-1))/
     &          AREA(I)/DZCELL(I)
          ELSE
             SIP(I)=SIP(I)+DIFP(I)*(AREA(I)-AREA(I-1))/AREA(I)/
     &          DZCELL(I)/F(I,JPO4)
```

```
          ENDIF
C-----VERTICAL INTEGRATION OF PROCESSES
          TOTDIFFP=TOTDIFFP+DIFP(I)*(AREA(I)-AREA(I-1))*DT
          TOTPDENI=TOTPDENI+WDEN(I)*CCNDENIT*CPC*DZCELL(I)*
     &         DT*AREA(I)
          TOTDDC5=TOTDDC5+DDC(I)*CPC*DZCELL(I)*DT*AREA(I)
          TOTPGRO=TOTPGRO+GRO(I)*CPC*DZCELL(I)*DT*AREA(I)
          TOTPGRA=TOTPGRA+GRA(I)*CPC*DZCELL(I)*DT*AREA(I)
          TOTPGRX=TOTPGRX+GRAX(I)*CPC*DZCELL(I)*DT*AREA(I)
          TOTPGROM=TOTPGROM+GMDP(I)*DZCELL(I)*DT*AREA(I)
      ENDDO
      RETURN
      END

C*****************************************************************
      SUBROUTINE OXYGEN(GRO,GRA,GRAX,SMIN,NIT,GROM,DDC,
     1 TOTOGRO,TOTOGRA,TOTOGRX,TOTOGROM,TOTOSMIN,TOTDDC6,TOTONIT)
C*****************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL GRO(NIM),GRA(NIM),GRAX(NIM),SMIN(NIM)
      REAL DDC(NIM),GROM(NIM),NIT(NIM)
      REAL TOTOGRO,TOTOGRA,TOTOGRX,TOTOGROM,TOTOSMIN,TOTDDC6,TOTONIT

C-----ASSIGN VALUES TO SI AND SIP
C     NOTE THAT SINKS ARE INCLUDED IN THE SI-TERM
C     THIS MAKES IT POSSIBLE FOR JOXY TO BECOME NEGATIVE!
      DO I=2,NM1
          SI(I)=GRO(I)*COC-SMIN(I)*COC*(AREA(I)-AREA(I-1))/
     &     AREA(I)/DZCELL(I)+GRA(I)*COC+GRAX(I)*COC+GROM(I)*COC-
     &     DDC(I)*COC-NIT(I)*CONNITR*(AREA(I)-AREA(I-1))/
     &      AREA(I)/DZCELL(I)
          SIP(I)=0.
C-----VERTICAL INTEGRATION OF PROCESSES
          TOTOGRO=TOTOGRO+GRO(I)*COC*DZCELL(I)*DT*AREA(I)
          TOTOGRA=TOTOGRA+GRA(I)*COC*DZCELL(I)*DT*AREA(I)
          TOTOGRX=TOTOGRX+GRAX(I)*COC*DZCELL(I)*DT*AREA(I)
          TOTOGROM=TOTOGROM+GROM(I)*COC*DZCELL(I)*DT*AREA(I)
          TOTOSMIN=TOTOSMIN+SMIN(I)*COC*(AREA(I)-AREA(I-1))*DT
          TOTDDC6=TOTDDC6+DDC(I)*COC*DZCELL(I)*DT*AREA(I)
          TOTONIT=TOTONIT+NIT(I)*CONNITR*(AREA(I)-AREA(I-1))*DT
      ENDDO
      RETURN
      END

C*****************************************************************
      SUBROUTINE BPHOS(SMIN,SDEN,DIFP,GMBP,TOTBPMIN,TOTBPDEN,
     1 TOTBPMAC,TOTBPDIF,SAREA)
C*****************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL SMIN(NIM),SDEN(NIM),DIFP(NIM),GMBP(NIM)
      REAL SAREA
      REAL TOTBPMIN,TOTBPDEN,TOTBPMAC,TOTBPDI

C-----ASSIGN VALUES TO SI AND SIP
      DO I=2,NM1
          SI(I)=SMIN(I)*CPC+
     &        POINTBIP/SAREA/1000./365./86400.*
     &        (AREA(I)-AREA(I-1))/AREA(N)
     &        +SDEN(I)*CPC*CCNDENIT
```

```
          SIP(I)=-GMBP(I)/F(I,JBIP)
          IF(DIFP(I).GT.0.)THEN
             SIP(I)=SIP(I)-DIFP(I)/F(I,JBIP)
          ELSE
             SI(I)=SI(I)-DIFP(I)
          ENDIF
C-----VERTICAL INTEGRATION OF PROCESSES
          TOTBPMIN=TOTBPMIN+SMIN(I)*CPC*(AREA(I)-AREA(I-1))*DT
          TOTBPDEN=TOTBPDEN+SDEN(I)*CPC*CCNDENIT*
     &          (AREA(I)-AREA(I-1))*DT
          TOTBPMAC=TOTBPMAC+GMBP(I)*(AREA(I)-AREA(I-1))*DT
          TOTBPDIF=TOTBPDIF+DIFP(I)*(AREA(I)-AREA(I-1))*DT
       ENDDO
       RETURN
       END

C*******************************************************************
      SUBROUTINE BAMMO(SMIN,SDEN,NIT,DIFNH,GMBN,
     1 TOTBNMIN,TOTBNDEN,TOTNNIT,TOTBNMAC,TOTBNDIF)
C*******************************************************************
       INCLUDE 'comp97.inc'
       INCLUDE 'biopar.inc'
       INTEGER I
       REAL SMIN(NIM),SDEN(NIM),NIT(NIM),DIFNH(NIM),GMBN(NIM)
       REAL TOTBNMIN,TOTBNDEN,TOTNNIT,TOTBNMAC,TOTBNDIF

C-----ASSIGN VALUES TO SI AND SIP
      DO I=2,NM1
             SI(I)=SMIN(I)*CNC
     &          +SDEN(I)*CNC*CCNDENIT
             SIP(I)=-NIT(I)/F(I,JBNH)-
     &            GMBN(I)/F(I,JBNH)
          IF(DIFNH(I).GT.0.)THEN
             SIP(I)=SIP(I)-DIFNH(I)/F(I,JBNH)
          ELSE
             SI(I)=SI(I)-DIFNH(I)
          ENDIF
C-----VERTICAL INTEGRATION OF PROCESSES
          TOTBNMIN=TOTBNMIN+SMIN(I)*CNC*(AREA(I)-AREA(I-1))*DT
          TOTBNDEN=TOTBNDEN+SDEN(I)*CNC*CCNDENIT*
     &          (AREA(I)-AREA(I-1))*DT
          TOTNNIT=TOTNNIT+NIT(I)*(AREA(I)-AREA(I-1))*DT
          TOTBNMAC=TOTBNMAC+GMBN(I)*(AREA(I)-AREA(I-1))*DT
          TOTBNDIF=TOTBNDIF+DIFNH(I)*(AREA(I)-AREA(I-1))*DT
       ENDDO
       RETURN
       END

C*******************************************************************
      SUBROUTINE BNITR(SDEN,NIT,DIFNO,TOTINIT,TOTBIDEN,TOTBIDIF)
C*******************************************************************
       INCLUDE 'comp97.inc'
       INCLUDE 'biopar.inc'
       INTEGER I
       REAL SDEN(NIM),NIT(NIM),DIFNO(NIM)
       REAL TOTINIT,TOTBIDEN,TOTBIDIF

C-----ASSIGN VALUES TO SI AND SIP
      DO I=2,NM1
          SI(I)=NIT(I)
          SIP(I)=-SDEN(I)/F(I,JBNO)
          IF(DIFNO(I).GT.0.)THEN
             SIP(I)=SIP(I)-DIFNO(I)/F(I,JBNO)
```

```
         ELSE
            SI(I)=SI(I)-DIFNO(I)
         ENDIF
C-----VERTICAL INTEGRATION OF PROCESSES
         TOTINIT=TOTINIT+NIT(I)*(AREA(I)-AREA(I-1))*DT
         TOTBIDEN=TOTBIDEN+SDEN(I)*(AREA(I)-AREA(I-1))*DT
         TOTBIDIF=TOTBIDIF+DIFNO(I)*(AREA(I)-AREA(I-1))*DT
      ENDDO
      RETURN
      END

C*****************************************************************
      SUBROUTINE FISHA(PREZ,PREFI,TOTFAGR,TOTFAM)
C*****************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL PREZ(NIM),PREFI(NIM),TOTFAGR,TOTFAM

C-----ASSIGN VALUES TO SI AND SIP
      DO I=2,NM1
         SI(I)=OMEGA*PREZ(I)
         SIP(I)=-PREFI(I)/F(I,JFIA)
C-----VERTICAL INTEGRATION OF PROCESSES
         TOTFAGR=TOTFAGR+OMEGA*PREZ(I)*DZCELL(I)*DT*AREA(I)
         TOTFAM=TOTFAM+PREFI(I)*DZCELL(I)*DT*AREA(I)
      ENDDO
      RETURN
      END

C*****************************************************************
      SUBROUTINE PISCI(PREFI,FMORT,TOTFBGR,TOTFBM)
C*****************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL PREFI(NIM),FMORT(NIM),TOTFBGR,TOTFBM

C-----ASSIGN VALUES TO SI AND SIP
      DO I=2,NM1
         SI(I)=OMEGA*PREFI(I)
         SIP(I)=-FMORT(I)/F(I,JFIB)
C-----VERTICAL INTEGRATION OF PROCESSES
         TOTFBGR=TOTFBGR+OMEGA*PREFI(I)*DZCELL(I)*DT*AREA(I)
         TOTFBM=TOTFBM+FMORT(I)*DZCELL(I)*DT*AREA(I)
      ENDDO
      RETURN
      END
```

BIOPRO.FOR

The file BIOPRO.FOR contains the subroutines ADSCOEFF, CBMORT, DDCOMP, DENITR, ETADVECTION, FISHMORT, GRAZER, GROWTH, MACMORT, MACROF, CBGROWTH, NITFIC, PREDATF, PREDATZ, SEDDIFF, SEDMINE, SEQUEST, SINKCB, SINKDET, SINKPHY, ZOOMORT, LIGHT and TEMPBER.

```
C*******************************************************************
      SUBROUTINE ADSCOEFF(FAKTA)
C*******************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL FAKTA(NIM),PSC,D,A,K

      DO I=2,NM1
         IF (F(I,JOXY).LT.OXYLIMIT) THEN
            PSC=ADSCAPAN*(ANACOEFF**(F(I,JOXY)/OXYLIMIT))*HAR
         ELSE
            PSC=ADSCAP*HAR
         ENDIF
         K=KAD*HAR
         D=DANS*HAR
         DO
            A=PSC*D/(D+K)
            IF (ABS((A+D)-F(I,JBIP)).LT.(0.001*F(I,JBIP))) EXIT
            D=D*F(I,JBIP)/(D+A)
         ENDDO
         FAKTA(I)=A/D
      ENDDO
      RETURN
      END

C*******************************************************************
      SUBROUTINE CBMORT(MORTX)
C*******************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL MORTX(NIM)

      DO I=NM1,2,-1
         MORTX(I)=XMRT/86400.*F(I,JANF)
         IF(F(I,JANF)-MORTX(I)*DT.LE.ANFMIN)MORTX(I)=0.
      ENDDO
      RETURN
      END

C*******************************************************************
      SUBROUTINE DDCOMP(DDC)
C*******************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL DDC(NIM),MYTEMP(NIM)

      CALL TEMPBER(TREFG,TETAG,MYTEMP)
      DO I=2,NM1
         DDC(I)=DDCC/86400.*MYTEMP(I)*F(I,JDET)
         IF(F(I,JDET)-DDC(I)*DT.LE.DETMIN) DDC(I)=0.
      ENDDO
```

```
      RETURN
      END

C*********************************************************************
      SUBROUTINE DENITR(SDEN,WDEN)
C*********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL POSDEN,MYTEMP(NIM),SDEN(NIM),WDEN(NIM)

      CALL TEMPBER(TDEN,TETADEN,MYTEMP)
      DO I=2,NM1
         IF(F(I,JOXY).GT.0.0)THEN
            POSDEN=F(I,JBNO)/(KBNO*DSED+F(I,JBNO))
            SDEN(I)=ADEN/86400.*MYTEMP(I)*POSDEN
            WDEN(I)=0.0
         ELSE
            POSDEN=F(I,JDIN)/(KBNO+F(I,JDIN))
            SDEN(I)=0.0
            WDEN(I)=BDEN/86400.*MYTEMP(I)*POSDEN
         ENDIF
         IF(F(I,JBNO)-SDEN(I)*DT.LT.BNOMIN) SDEN(I)=0.
         IF(F(I,JDIN)-WDEN(I)*DT.LT.DINMIN) WDEN(I)=0.
      ENDDO
      RETURN
      END

C*********************************************************************
         SUBROUTINE ETADVECTION(ETD,WG,ETPROP)
C*********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL ETD,WG,ETPROP(NIM)

      DO I = 2, N-1
         IF (Z(I).LT.ETD) THEN
            ETPROP(I) = 0.0
         ELSE
            ETPROP(I)=RESP*9.5866E-7*WG**2/600.
         ENDIF
      ENDDO
      RETURN
      END

C*********************************************************************
      SUBROUTINE FISHMORT(FISHMRT)
C*********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL FISHMRT(NIM)

      DO I=2,NM1
         FISHMRT(I)=FMRT/86400.*F(I,JFIB)
         IF((F(I,JFIB)-FISHMRT(I)*DT).LE.FIBMIN) FISHMRT(I)=0.
      ENDDO
      RETURN
      END
```

```
C*********************************************************************
      SUBROUTINE GRAZER(ZZ,GG,DD,AA,D2)
C*********************************************************************
       INCLUDE 'comp97.inc'
       INCLUDE 'biopar.inc'
       INTEGER I
       REAL PP,PD,PA
       REAL GG(NIM),DD(NIM),ZZ(NIM),AA(NIM),D2(NIM)

       DO I=2,NM1
         PP=P1*F(I,JPHY)/(P1*F(I,JPHY)+P2*F(I,JDET)+P3*F(I,JANF)+KG)*
     1     GAMMA/86400.*F(I,JZOO)
         PD=P2*F(I,JDET)/(P1*F(I,JPHY)+P2*F(I,JDET)+P3*F(I,JANF)+KG)*
     1     GAMMA/86400.*F(I,JZOO)
         PA=P3*F(I,JANF)/(P1*F(I,JPHY)+P2*F(I,JDET)+P3*F(I,JANF)+KG)*
     1     GAMMA/86400.*F(I,JZOO)
         IF(F(I,JPHY)-PP*DT.LE.PHYMIN)PP=0.
         IF(F(I,JDET)-PD*DT.LE.DETMIN)PD=0.
         IF(F(I,JANF)-PA*DT.LE.ANFMIN)PA=0.
         ZZ(I)=EFFI*(PP+PD+PA)
         GG(I)=PP
         DD(I)=PD
         AA(I)=PA
         D2(I)=(1-EFFI)/EFFI*ZZ(I)
       ENDDO
       RETURN
       END

C*********************************************************************
      SUBROUTINE GROWTH(I0,GR,MACOFF)
C*********************************************************************
       INCLUDE 'comp97.inc'
       INCLUDE 'biopar.inc'
       INTEGER I,IH
       REAL X2(NIM),X3(NIM),GR(NIM),I0,LGHW
       REAL MYLGTH(NIM,NIM),MYTEMP(NIM)

       CALL LIGHT(I0,IOPTA,MYLGTH,MACOFF)
       CALL TEMPBER(TPHY,TETAPHY,MYTEMP)

       DO I=2,NM1
C-----NITROGEN/PHOSPHORUS LIMITATION
          X2(I)=F(I,JDIN)/(KDIN+F(I,JDIN))
          X3(I)=F(I,JPO4)/(KPO4+F(I,JPO4))
C-----PHYTOPLANKTON GROWTH
          IF (X2(I).LE.X3(I)) THEN
             GR(I)=MYMAX/86400.*MYTEMP(I)*X2(I)*F(I,JPHY)
          ELSE
             GR(I)=MYMAX/86400.*MYTEMP(I)*X3(I)*F(I,JPHY)
          ENDIF
          LGHW=0.0
          DO IH=2,I
             LGHW=LGHW+MYLGTH(I,IH)*(AREA(IH)-AREA(IH-1))/AREA(I)
          ENDDO
          GR(I)=GR(I)*LGHW
      ENDDO
      RETURN
      END
```

```
C*********************************************************************
      SUBROUTINE MACMORT(MMORT,MACOFF)
C*********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL MMORT(NIM)
      LOGICAL MACOFF

      IF (MACOFF) GOTO 50
      DO I=2,NM1
         MMORT(I)=MMRT/86400.*F(I,JMAC)
         IF (F(I,JMAC)-MMORT(I)*DT.LE.MACMIN) MMORT(I)=0.
      ENDDO
      RETURN

50    DO I=2,NM1
         MMORT(I)=0.
      ENDDO
      RETURN
      END

C*********************************************************************
      SUBROUTINE MACROF(I0,GRM,GMBN,GMDN,GMBP,GMDP,MACOFF)
C*********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I,ID
      REAL X2(NIM),X3(NIM),GRM(NIM),I0,LGHW
      REAL MYLGTH(NIM,NIM),MYTEMP(NIM)
      REAL GMBP(NIM),GMDP(NIM),GMBN(NIM),GMDN(NIM)
      REAL X4(NIM),X5(NIM)
      LOGICAL MACOFF
C
      IF (MACOFF) GOTO 50
      CALL LIGHT(I0,IOPTM,MYLGTH,MACOFF)
      CALL TEMPBER(TMAC,TETAMAC,MYTEMP)

      DO I=2,NM1
C-----NITROGEN/PHOSPHORUS LIMITATION
         X2(I)=(F(I,JBNH)/DSED2+F(I,JDIN))/
     1           (KMIN+F(I,JBNH)/DSED2+F(I,JDIN))
         X3(I)=(F(I,JBIP)/DSED2+F(I,JPO4))/
     1           (KMIP+F(I,JBIP)/DSED2+F(I,JPO4))
         IF (X2(I).LE.X3(I)) THEN
            GRM(I)=MYMAXM/86400.*MYTEMP(I)*X2(I)*F(I,JMAC)
         ELSE
            GRM(I)=MYMAXM/86400.*MYTEMP(I)*X3(I)*F(I,JMAC)
         ENDIF
         LGHW=0.0
         DO ID=I,NM1
            LGHW=LGHW+MYLGTH(ID,I)*DZCELL(ID)/(ZDIM-ZBOUND(I-1))
         ENDDO
         GRM(I)=GRM(I)*LGHW
         X4(I)=CNC*GRM(I)*(ZDIM-ZBOUND(I-1))
         X5(I)=CPC*GRM(I)*(ZDIM-ZBOUND(I-1))
         IF (X4(I)*DT.LT.(F(I,JBNH)-BNHMIN)) THEN
            GMBN(I)=X4(I)
            GMDN(I)=0.0
         ELSEIF(F(I,JBNH).GT.BNHMIN)THEN
            GMBN(I)=F(I,JBNH)-BNHMIN
            GMDN(I)=X4(I)/DZCELL(I)-GMBN(I)/DZCELL(I)
            IF (F(I,JDIN)-GMDN(I)*DT.LT.DINMIN) GOTO 49
```

```
          ELSE
              GMBN(I)=0.
              GMDN(I)=X4(I)/DZCELL(I)
              IF (F(I,JDIN)-GMDN(I)*DT.LT.DINMIN) GOTO 49
          ENDIF
          IF (X5(I)*DT.LT.(F(I,JBIP)-BIPMIN)) THEN
              GMBP(I)=X5(I)
              GMDP(I)=0.0
          ELSEIF (F(I,JBIP).GT.BIPMIN)THEN
              GMBP(I)=F(I,JBIP)-BIPMIN
              GMDP(I)=X5(I)/DZCELL(I)-GMBP(I)/DZCELL(I)
              IF (F(I,JPO4)-GMDP(I)*DT.LT.PO4MIN) GOTO 49
          ELSE
              GMBP(I)=0.
              GMDP(I)=X5(I)/DZCELL(I)
              IF (F(I,JPO4)-GMDP(I)*DT.LT.PO4MIN) GOTO 49
          ENDIF
          RETURN
49        GRM(I)=0.
          GMBP(I)=0.
          GMDP(I)=0.
          GMBN(I)=0.
          GMDN(I)=0.
      ENDDO
      RETURN
50    DO I=2,NM1
          GRM(I)=0.
          GMBP(I)=0.
          GMDP(I)=0.
          GMBN(I)=0.
          GMDN(I)=0.
      ENDDO
      RETURN
      END

C*********************************************************************
      SUBROUTINE CBGROWTH(I0,GRA,GRAX,MACOFF)
C*********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I,IH
      REAL X2(NIM),X3(NIM),GRA(NIM),GRAX(NIM),I0,LGHW
      REAL MYLGTH(NIM,NIM),MYTEMP(NIM)

      CALL LIGHT(I0,IOPTA,MYLGTH,MACOFF)
      CALL TEMPBER(TPHY,TETAPHY,MYTEMP)

      DO I=2,NM1
C-----NITROGEN/PHOSPHORUS LIMITATION
          X2(I)=F(I,JDIN)/(KDIN+F(I,JDIN))
          X3(I)=F(I,JPO4)/(KPO4+F(I,JPO4))
C-----CYANOBACTERIA GROWTH
          IF (X2(I).LE.X3(I)) THEN
              GRAX(I)=MYMAXA/86400.*MYTEMP(I)*
     1        X3(I)/(1+(F(I,JDIN)/F(I,JPO4)/KNFIX)**SNF4)*F(I,JANF)
              GRA(I)=0.
          ELSE
              GRAX(I)=0.
              GRA(I)=MYMAXB/86400.*MYTEMP(I)*X3(I)*F(I,JANF)
          ENDIF
          LGHW=0.0
          DO IH=2,I
```

```
            LGHW=LGHW+MYLGTH(I,IH)*(AREA(IH)-AREA(IH-1))/AREA(I)
         ENDDO
         GRA(I)=GRA(I)*LGHW
         GRAX(I)=GRAX(I)*LGHW
      ENDDO
      RETURN
      END

C**********************************************************************
      SUBROUTINE NITFIC(NIT)
C**********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL NIT(NIM),MYTEMP(NIM)

      CALL TEMPBER(TNFIC,TETANFIC,MYTEMP)

      DO I=2,NM1
         NIT(I)=TSC/86400.*MYTEMP(I)*
     1   MAX(MIN(F(I,JBNH),F(I,JOXY)/CONNITR/DSED2),0.)
         IF ((F(I,JBNH)-NIT(I)*DT.LE.BNHMIN).OR.
     1      (F(I,JOXY).LE.NIT(I)*DT *CONNITR*
     1      (AREA(I)-AREA(I-1))/AREA(I)/DZCELL(I))) THEN
            NIT(I)=0.
         ENDIF
      ENDDO
      RETURN
      END

C**********************************************************************
      SUBROUTINE PREDATF(PREFI)
C**********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL PREFI(NIM)

      DO I=2,NM1
        PREFI(I)=PREDF/86400.*EXP(LOG(TETAG)*(F(I,JTE)-TREFG))*
     1     F(I,JFIA)**2/(KFIA**2+F(I,JFIA)**2)*F(I,JFIB)
        IF((F(I,JFIA)-PREFI(I)*DT).LE.FIAMIN) PREFI(I)=0.
      ENDDO
      RETURN
      END

C**********************************************************************
      SUBROUTINE PREDATZ(PREZ)
C**********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL PREZ(NIM)

      DO I=2,NM1
        PREZ(I)=PREDZ/86400.*EXP(LOG(TETAG)*(F(I,JTE)-TREFG))*
     1     F(I,JZOO)**2/(KZO**2+F(I,JZOO)**2)*F(I,JFIA)
        IF((F(I,JZOO)-PREZ(I)*DT).LE.ZOOMIN) PREZ(I)=0.
      ENDDO
      RETURN
      END
```

```
C*********************************************************************
      SUBROUTINE SEDDIFF(RP,RNO,RNH,FAKTA)
C*********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL RP(NIM),RNH(NIM),RNO(NIM),FAKTA(NIM)

      DO I=2,NM1
        RP(I)=KDIFF*(F(I,JBIP)/(1+FAKTA(I))/HAR-F(I,JPO4))/HAR
        RNH(I)=KDIFF*F(I,JBNH)/HAR/HAR
        RNO(I)=KDIFF*(F(I,JBNO)-F(I,JDIN)*DZCELL(I))/HAR/HAR
      ENDDO
      RETURN
      END

C*********************************************************************
      SUBROUTINE SEDMINE(SMIN,SDEN)
C*********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL POSDEN,MYTEMP(NIM),SMIN(NIM),SDEN(NIM)

      CALL TEMPBER(TSMIN,TETASMIN,MYTEMP)
      DO I=2,NM1
         SMIN(I)=MAX(MINMAX*MYTEMP(I)*F(I,JSED)-SDEN(I)*CCNDENIT,0.)
      ENDDO
      RETURN
      END

C*********************************************************************
      SUBROUTINE SEQUEST(SEQ)
C*********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL SEQ(NIM)

      DO I=2,NM1
         SEQ(I)=SEQCON*F(I,JSED)
         IF(F(I,JSED)-SEQ(I)*DT.LT.SEDMIN) SEQ(I)=0.
      ENDDO
      RETURN
      END

C*********************************************************************
      SUBROUTINE SINKCB(SINKO,SINKI,SED)
C*********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL SINKO(NIM),SINKI(NIM),SED(NIM)

C-----SINKING OF CYANOBACTERIA
      DO I=NM1,2,-1
         SINKO(I)=SINKAN/86400./DZCELL(I)*F(I,JANF)
         IF(F(I,JANF)-SINKO(I)*DT.LE.ANFMIN)SINKO(I)=0.
         SINKI(I-1)=SINKO(I)*DZCELL(I)/DZCELL(I-1)
         SED(I)=SINKO(I)*DZCELL(I)
      ENDDO
      SINKI(NM1)=0.
      SINKI(1)=0.
```

```
      RETURN
      END

C*********************************************************************
      SUBROUTINE SINKDET(SINKO,SINKI,SED)
C*********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL SINKO(NIM),SINKI(NIM),SED(NIM)

C-----DETRITUS SINKING
      DO I=NM1,2,-1
         SINKO(I)=SINKD/86400./DZCELL(I)*F(I,JDET)
         IF(F(I,JDET)-SINKO(I)*DT.LE.DETMIN)SINKO(I)=0.
         SINKI(I-1)=SINKO(I)*DZCELL(I)/DZCELL(I-1)
         SED(I)=SINKO(I)*DZCELL(I)
      ENDDO
      SINKI(NM1)=0.
      SINKI(1)=0.
      RETURN
      END

C*********************************************************************
      SUBROUTINE SINKPHY(SINKO,SINKI,SED)
C*********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL SINKO(NIM),SINKI(NIM),SED(NIM)

C-----PHYTOPLANKTON SINKING (POSITIVE DOWNWARD)
      DO I=NM1,2,-1
         SINKO(I)=SINKP/86400./DZCELL(I)*F(I,JPHY)
         IF(F(I,JPHY)-SINKO(I)*DT.LE.PHYMIN)SINKO(I)=0.
         SINKI(I-1)=SINKO(I)*DZCELL(I)/DZCELL(I-1)
         SED(I)=SINKO(I)*DZCELL(I)
      ENDDO
      SINKI(NM1)=0.
      SINKI(1)=0.
      RETURN
      END

C*********************************************************************
      SUBROUTINE ZOOMORT(ZNMRT)
C*********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL ZNMRT(NIM)

      DO I=2,NM1
        ZNMRT(I)=ZMRT/86400.*F(I,JZOO)
        IF((F(I,JZOO)-ZNMRT(I)*DT).LE.ZOOMIN) ZNMRT(I)=0.
      ENDDO
      RETURN
      END

C*********************************************************************
      SUBROUTINE LIGHT(I0,IOPT,MYLGHT,MACOFF)
C*********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
```

```
      INTEGER ID,IH,I
      LOGICAL MACOFF
      REAL I0,K(NIM,NIM),K2(NIM,NIM),IZ(NIM,NIM),IOPT
      REAL KP,KD,KM
      REAL MYLGHT(NIM,NIM)

      DO ID=NM1,2,-1
          DO IH=ID,2,-1
C-----EXTINCTION COEFFICIENT K
C-----------SELF SHADING OF PHYTOPLANKTON
            KP=KPKONST*(F(ID,JPHY)+F(ID,JANF))
C-----------SHADING BY DETRITUS
            KD=KDKONST*F(ID,JDET)
C-----------SHADING BY MACROPHYTES
            IF(MACOFF)THEN
               KM=0.
            ELSE
               KM=KMKONST*F(IH,JMAC)
            ENDIF
            K(ID,IH)=KW+KP+KD+KM
            K2(ID,IH)=0.0
          ENDDO
       ENDDO
      DO ID=NM1,2,-1
        DO IH=ID,2,-1
          DO I=ID,NM1
C-----------THE EXPONENT OF THE LIGHT FUNCTION
            K2(ID,IH)=K2(ID,IH)+K(I,IH)*(Z(I+1)-Z(I))
          ENDDO
          IZ(ID,IH)=IZKONST*I0*EXP(-K2(ID,IH))
          MYLGHT(ID,IH)=(IZ(ID,IH)/IOPT)*EXP(1.-IZ(ID,IH)/IOPT)
        ENDDO
      ENDDO
      RETURN
      END

C*********************************************************************
      SUBROUTINE TEMPBER(TKOEFF,TETA,MYTEMP)
C*********************************************************************
      INCLUDE 'comp97.inc'
      INCLUDE 'biopar.inc'
      INTEGER I
      REAL MYTEMP(NIM)
      REAL TKOEFF,TETA
      DO I=2,NM1
         MYTEMP(I)=EXP(LOG(TETA)*(F(I,JTE)-TKOEFF))
      ENDDO
      RETURN
      END
```

BIOPAR.FOR

The file BIOPAR.FOR contains only the subroutine BIOPAR.

```
C*******************************************************************
      SUBROUTINE BIOPAR(fileunit)
C*******************************************************************

      INTEGER fileunit
      REAL value
      LOGICAL equal
      CHARACTER(LEN=80) line
      CHARACTER(LEN=12) str

      INCLUDE 'biopar.inc'

C     Default values with reference
      P1=1.              !RH16
      P2=0.4             !RH16
      P3=0.2             !RH16
      SINKP=0.8          !RH16
      SINKD=1.2          !RH16
      SINKAN=0.          !RH16
      OMEGA=1.           !RH16, not used there
      ZMRT=0.005         !RH16
      KPKONST=0.3        !RH16
      KW=0.04            !RH16
      KDKONST=0.2        !RH16
      IZKONST=1.0        !RH16, not used there
      IOPTA=50.          !RH16
      MYMAX=2.           !RH16
      MYMAXA=1.          !RH16
      MYMAXB=1.          !RH16, same as MYMAXA
      KMKONST=0.16       !RH16
      KDIN=0.02          !RH16
      KPO4=0.01          !RH16
      KBNO=3.            !RH16
      KMIP=0.005         !RH16
      KMIN=0.01          !RH16
      TMAC=20.           !RH16, same as TPHY
      TETAMAC=1.04       !RH16
      MYMAXM=0.5         !RH16
      IOPTM=75.          !RH16
      MMRT=0.05          !RH16
      XMRT=0.            !RH16
      SNF4=4.            !RH16
      KNFIX=7.2          !RH16
      TSC=0.08           !RH16,depend on DSED2
      TNFIC=21.          !RH16
      TETANFIC=1.1       !RH16
      TETAG=1.07         !RH16, for fishpred, same as TETAPHY
      TREFG=20.          !RH16, for fishpred, same as TPHY
      TPHY=20.           !RH16
      TETAPHY=1.07       !RH16
      TSMIN=20.          !RH16, same as TPHY
      TETASMIN=1.07      !RH16, same as TETAPHY
      TDEN=20.           !RH16, same as TPHY
      TETADEN=1.07       !RH16, same as TETAPHY
      GAMMA=0.8          !RH16
      KG=0.5             !RH16
      EFFI=0.4           !RH16
      PREDZ=0.1          !RH16
```

```
 KZO=1.5            !RH16
 PREDF=0.1          !RH16
 KFIA=0.2           !RH16
 FMRT=0.0008        !RH16
 DDCC=0.02          !RH16
 KDIFF=1.E-9        !RH16
 HAR=0.03           !RH17
 OXYLIMIT=1.        !RH17
 ADSCAP=129.883     !RH17
 ADSCAPAN=51.953    !RH17
 KAD=0.652          !RH17
 DANS=4.            !RH17
 ADEN=0.16          !RH16
 BDEN=0.03          !RH16
 DSED=0.1           !RH16, for denitrification
 MINMAX=0.002       !RH16
 DSED2=0.1          !RH16, for macrophytes and nitrification, same
                          !as DSED
 SEQCON=0.          !RH16
 RESP=1.            !RH17
 NDEPPAR=1500.      !RH16   mg/m2/yr
 PDEPPAR=7.         !RH16
 POINTDIN=0.
 POINTDET=0.
 POINTPO4=0.
 POINTBIP=0.

DO
   READ(fileunit,600,END=300) line
   IF(line(1:1).EQ.'!') CYCLE
   i = 1
   DO
      equal = (line(i: i) .EQ. '=')
      IF(equal) THEN
        EXIT
      ELSE
        i=i+1
      ENDIF
   ENDDO
   str  = line(1:i-1)
   line = line(i+1:80)
   READ(line,*) value

   IF(str.EQ.'P1')          P1=value
   IF(str.EQ.'P2')          P2=value
   IF(str.EQ.'P3')          P3=value
   IF(str.EQ.'SINKP')       SINKP=value
   IF(str.EQ.'SINKD')       SINKD=value
   IF(str.EQ.'SINKAN')      SINKAN=value
   IF(str.EQ.'OMEGA')       OMEGA=value
   IF(str.EQ.'ZMRT')        ZMRT=value
   IF(str.EQ.'KPKONST')     KPKONST=value
   IF(str.EQ.'KW')          KW=value
   IF(str.EQ.'KDKONST')     KDKONST=value
   IF(str.EQ.'IZKONST')     IZKONST=value
   IF(str.EQ.'IOPTA')       IOPTA=value
   IF(str.EQ.'MYMAX')       MYMAX=value
   IF(str.EQ.'MYMAXA')      MYMAXA=value
   IF(str.EQ.'MYMAXB')      MYMAXB=value
   IF(str.EQ.'KMKONST')     KMKONST=value
   IF(str.EQ.'KDIN')        KDIN=value
   IF(str.EQ.'KPO4')        KPO4=value
   IF(str.EQ.'KBNO')        KBNO=value
```

```
      IF(str.EQ.'KMIP')         KMIP=value
      IF(str.EQ.'KMIN')         KMIN=value
      IF(str.EQ.'TMAC')         TMAC=value
      IF(str.EQ.'TETAMAC')      TETAMAC=value
      IF(str.EQ.'MYMAXM')       MYMAXM=value
      IF(str.EQ.'IOPTM')        IOPTM=value
      IF(str.EQ.'MMRT')         MMRT=value
      IF(str.EQ.'XMRT')         XMRT=value
      IF(str.EQ.'SNF4')         SNF4=value
      IF(str.EQ.'KNFIX')        KNFIX=value
      IF(str.EQ.'TSC')          TSC=value
      IF(str.EQ.'TNFIC')        TNFIC=value
      IF(str.EQ.'TETANFIC')     TETANFIC=value
      IF(str.EQ.'TETAG')        TETAG=value
      IF(str.EQ.'TREFG')        TREFG=value
      IF(str.EQ.'TPHY')         TPHY=value
      IF(str.EQ.'TETAPHY')      TETAPHY=value
      IF(str.EQ.'TSMIN')        TSMIN=value
      IF(str.EQ.'TETASMIN')     TETASMIN=value
      IF(str.EQ.'TDEN')         TDEN=value
      IF(str.EQ.'TETADEN')      TETADEN=value
      IF(str.EQ.'GAMMA')        GAMMA=value
      IF(str.EQ.'KG')           KG=value
      IF(str.EQ.'EFFI')         EFFI=value
      IF(str.EQ.'PREDZ')        PREDZ=value
      IF(str.EQ.'KZO')          KZO=value
      IF(str.EQ.'PREDF')        PREDF=value
      IF(str.EQ.'KFIA')         KFIA=value
      IF(str.EQ.'FMRT')         FMRT=value
      IF(str.EQ.'DDCC')         DDCC=value
      IF(str.EQ.'KDIFF')        KDIFF=value
      IF(str.EQ.'HAR')          HAR=value
      IF(str.EQ.'OXYLIMIT')     OXYLIMIT=value
      IF(str.EQ.'ADSCAP')       ADSCAP=value
      IF(str.EQ.'ADSCAPAN')     ADSCAPAN=value
      IF(str.EQ.'KAD')          KAD=value
      IF(str.EQ.'DANS')         DANS=value
      IF(str.EQ.'ADEN')         ADEN=value
      IF(str.EQ.'BDEN')         BDEN=value
      IF(str.EQ.'DSED')         DSED=value
      IF(str.EQ.'MINMAX')       MINMAX=value
      IF(str.EQ.'DSED2')        DSED2=value
      IF(str.EQ.'SEQCON')       SEQCON=value
      IF(str.EQ.'RESP')         RESP=value
      IF(str.EQ.'NDEPPAR')      NDEPPAR=value
      IF(str.EQ.'PDEPPAR')      PDEPPAR=value
      IF(str.EQ.'POINTDIN')     POINTDIN=value
      IF(str.EQ.'POINTDET')     POINTDET=value
      IF(str.EQ.'POINTPO4')     POINTPO4=value
      IF(str.EQ.'POINTBIP')     POINTBIP=value
   ENDDO
300   CONTINUE

   ANACOEFF = ADSCAP / ADSCAPAN

600   FORMAT(A80)
   END
```

BIOPAR.INC

The file BIOPAR.INC contains declarations of parameters and COMMON variables used in BIOLA.

```
C     BIOPAR.INC
C-----------------------------------------------------------------------
C     NAMES OF BIOVARIABLES
C-----------------------------------------------------------------------
      INTEGER JZOO,JPHY,JDET,JSED,JDIN,JANF,JMAC,JPO4,JOXY
      INTEGER JBIP,JBNH,JBNO,JFIA,JFIB
      PARAMETER(JZOO=11,JPHY=12,JDET=13,JSED=14,JDIN=15)
      PARAMETER(JANF=16,JMAC=17,JPO4=18,JOXY=19,JBIP=20)
      PARAMETER(JBNH=21,JBNO=22,JFIA=23,JFIB=24)
C-----------------------------------------------------------------------
C     UNIT TRANSFORM FACTORS
C-----------------------------------------------------------------------
      REAL COC,CNC,CPC,CCNDENIT,CONNITR
      PARAMETER(COC=2.667,CNC=0.176,CPC=0.0244)
      PARAMETER(CCNDENIT=1.071,CONNITR=4.572)
C
C-----------------------------------------------------------------------
C     LIMITIMG VALUES
C-----------------------------------------------------------------------
      REAL ZOOMIN,PHYMIN,DETMIN,SEDMIN,DINMIN,ANFMIN,MACMIN
      REAL PO4MIN,OXYMIN,BIPMIN,BNHMIN,BNOMIN,FIAMIN,FIBMIN
      PARAMETER(ZOOMIN=1.E-10,PHYMIN=1.E-4,DETMIN=1.E-10)
      PARAMETER(SEDMIN=1.E-10,DINMIN=1.E-10,ANFMIN=1.E-4)
      PARAMETER(MACMIN=1.E-4,PO4MIN=1.E-10,OXYMIN=-4.0)
      PARAMETER(BIPMIN=1.E-10,BNHMIN=1.E-10,BNOMIN=DINMIN*0.1)
                                          !BNOMIN=DINMIN*DSED3
      PARAMETER(FIBMIN=1.E-10,FIAMIN=1.E-10)
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C     PARAMETERS TO BE READ FROM FILE
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C     PREFERENCES OF ZOOPLANKTON GRAZING
C-----------------------------------------------------------------------
      REAL P1,P2,P3
C
C-----------------------------------------------------------------------
C     SINKING VELOCITY
C-----------------------------------------------------------------------
      REAL SINKD,SINKP,SINKAN
C
C-----------------------------------------------------------------------
C     ZOOPLANKTON MORTALITY
C-----------------------------------------------------------------------
      REAL OMEGA, ZMRT
C
C-----------------------------------------------------------------------
C     PHYTOPLANKTON GROWTH
C-----------------------------------------------------------------------
      REAL KPKONST,KW,KDKONST,IZKONST,IOPTA,MYMAX
      REAL MYMAXA,MYMAXB,KMKONST
C
C-----------------------------------------------------------------------
C     HALF SATURATION CONSTANTS
C-----------------------------------------------------------------------
```

```
      REAL KDIN,KPO4,KBNO,KMIP,KMIN
C
C-------------------------------------------------------------------
C     MACROPHYTE GROWTH
C-------------------------------------------------------------------
      REAL TMAC,TETAMAC,MYMAXM,IOPTM
C
C-------------------------------------------------------------------
C     MACROPHYTE MORTALITY
C-------------------------------------------------------------------
      REAL MMRT
C
C-------------------------------------------------------------------
C     BLUE-GREEN ALGAE MORTALITY
C-------------------------------------------------------------------
      REAL XMRT
C
C-------------------------------------------------------------------
C     NITROGEN FIXATION
C-------------------------------------------------------------------
      REAL SNF4,KNFIX
C
C-------------------------------------------------------------------
C     NITRIFICATION
C-------------------------------------------------------------------
      REAL TSC,TNFIC,TETANFIC
C
C-------------------------------------------------------------------
C     TEMPERATURE COEFFICIENTS
C-------------------------------------------------------------------
      REAL TETAG,TREFG,TPHY,TETAPHY,TSMIN,TETASMIN,TDEN,TETADEN
C
C-------------------------------------------------------------------
C     ZOOPLANKTON GRAZING
C-------------------------------------------------------------------
      REAL GAMMA,KG,EFFI
C
C-------------------------------------------------------------------
C     FISH PREDATION
C-------------------------------------------------------------------
      REAL PREDZ,KZO,PREDF,KFIA,FMRT
C
C-------------------------------------------------------------------
C     DETRITUS DECOMPOSITION
C-------------------------------------------------------------------
      REAL DDCC
C
C-------------------------------------------------------------------
C     SEDIMENT DIFFUSION
C-------------------------------------------------------------------
      REAL KDIFF,HAR
C
C-------------------------------------------------------------------
C     SEDIMENT ADSORPTION PHOSPHORUS
C-------------------------------------------------------------------
      REAL OXYLIMIT,ADSCAP,ADSCAPAN,ANACOEFF,KAD,DANS
C
C-------------------------------------------------------------------
C     DENITRIFICATION
C-------------------------------------------------------------------
      REAL ADEN,BDEN,DSED
C
C-------------------------------------------------------------------
```

```
C     SEDIMENT MINERALISATION SEQUESTERING AND RESUSPENSION
C-----------------------------------------------------------------------
      REAL MINMAX,DSED2,SEQCON,RESP
C
C-----------------------------------------------------------------------
C     DEPOSITION (mg/m2/yr)
C-----------------------------------------------------------------------
      REAL NDEPPAR,PDEPPAR
C
C-----------------------------------------------------------------------
C     SOURCES OF DIN, DET, PO4 AND SEDIMENT PHOSPHORUS (kg/yr)
C-----------------------------------------------------------------------
      REAL POINTDIN
      REAL POINTDET
      REAL POINTPO4
      REAL POINTBIP
C
C-----------------------------------------------------------------------

      COMMON /biola_parameters/
     +   P1,P2,P3,
     +   SINKD,SINKP,SINKAN,
     +   OMEGA,ZMRT,
     +   KPKONST,KW,KDKONST,IZKONST,IOPTA,MYMAX,
     +   MYMAXA,MYMAXB,KMKONST,
     +   KDIN,KPO4,KBNO,KMIP,KMIN,
     +   TMAC,TETAMAC,MYMAXM,IOPTM,
     +   MMRT,
     +   XMRT,
     +   SNF4,KNFIX,
     +   TSC,TNFIC,TETANFIC,
     +   TETAG,TREFG,TPHY,TETAPHY,TSMIN,TETASMIN,TDEN,TETADEN,
     +   GAMMA,KG,EFFI,
     +   PREDZ,KZO,PREDF,KFIA,FMRT,
     +   DDCC,
     +   KDIFF,HAR,
     +   OXYLIMIT,ADSCAP,ADSCAPAN,ANACOEFF,KAD,DANS,
     +   ADEN,BDEN,DSED,
     +   MINMAX,DSED2,SEQCON,RESP

      COMMON /biola_sources/
     +   NDEPPAR,PDEPPAR,
     +   POINTDIN,
     +   POINTDET,
     +   POINTPO4,
     +   POINTBIP
```

SMHI ger ut sex rapportserier. Tre av dessa, R-serierna är avsedda för internationell publik och skrivs därför oftast på engelska. I de övriga serierna används det svenska språket.

Seriernas namn	**Publiceras sedan**
RMK (Rapport Meteorologi och Klimatologi)	1974
RH (Rapport Hydrologi)	1990
RO (Rapport Oceanografi)	1986
METEOROLOGI	1985
HYDROLOGI	1985
OCEANOGRAFI	1985

I serien HYDROLOGI har tidigare utgivits:

1 Bengt Carlsson (1985)
Hydrokemiska data från de svenska fältforskningsområdena.

2 Martin Häggström och Magnus Persson (1986)
Utvärdering av 1985 års vårflödesprognoser.

3 Sten Bergström, Ulf Ehlin, SMHI, och Per-Eric Ohlsson, VASO (1986)
Riktlinjer och praxis vid dimensionering av utskov och dammar i USA. Rapport från en studieresa i oktober 1985.

4 Barbro Johansson, Erland Bergstrand och Torbjörn Jutman (1986)
Skåneprojektet - Hydrologisk och oceanografisk information för vattenplanering - Ett pilotprojekt.

5 Martin Häggström (1986)
Översiktlig sammanställning av den geografiska fördelningen av skador främst på dammar i samband med septemberflödet 1985.

6 Barbro Johansson (1986)
Vattenföringsberäkningar i Södermanlands län - ett försöksprojekt.

7 Maja Brandt (1986)
Areella snöstudier.

8 Bengt Carlsson, Sten Bergström, Maja Brandt och Göran Lindström (1987)
PULS-modellen: Struktur och tillämpningar.

9 Lennart Funkquist (1987)
Numerisk beräkning av vågor i kraftverksdammar.

10 Barbro Johansson, Magnus Persson, Enrique Aranibar and Robert Llobet (1987)
Application of the HBV model to Bolivian basins.

11 Cecilia Ambjörn, Enrique Aranibar and Roberto Llobet (1987)
Monthly streamflow simulation in Bolivian basins with a stochastic model.

12 Kurt Ehlert, Torbjörn Lindkvist och Todor Milanov (1987)
De svenska huvudvattendragens namn och mynningspunkter.

13 Göran Lindström (1987)
Analys av avrinningsserier för uppskattning av effektivt regn.

14 Maja Brandt, Sten Bergström, Marie Gardelin och Göran Lindström (1987)
Modellberäkning av extrem effektiv nederbörd.

15 Håkan Danielsson och Torbjörn Lindkvist (1987)
Sjökarte- och sjöuppgifter. Register 1987.

16 Martin Häggström och Magnus Persson (1987)
Utvärdering av 1986 års vårflödesprognoser.

17 Bertil Eriksson, Barbro Johansson, Katarina Losjö och Haldo Vedin (1987)
Skogsskador - klimat.

18 Maja Brandt (1987)
Bestämning av optimalt klimatstationsnät för hydrologiska prognoser.

19 Martin Häggström och Magnus Persson (1988)
Utvärdering av 1987 års vårflödesprognoser.

20 Todor Milanov (1988)
Frysförluster av vatten.

21 Martin Häggström, Göran Lindström, Luz Amelia Sandoval and Maria Elvira Vega (1988)
Application of the HBV model to the upper Río Cauca basin.

22 Mats Moberg och Maja Brandt (1988)
Snökartläggning med satellitdata i Kultsjöns avrinningsområde.

23 Martin Gotthardsson och Sten Lindell (1989)
Hydrologiska stationsnät 1989. Svenskt Vattenarkiv.

24 Martin Häggström, Göran Lindström, Luz Amelia Sandoval y Maria Elvira Vega (1989)
Aplicacion del modelo HBV a la cuenca superior del Río Cauca.

25 Gun Zachrisson (1989)
Svåra islossningar i Torneälven. Förslag till skadeförebyggande åtgärder.

26 Martin Häggström (1989)
Anpassning av HBV-modellen till Torneälven.

27 Martin Häggström and Göran Lindström (1990)
Application of the HBV model for flood forecasting in six Central American rivers.

28 Sten Bergström (1990)
Parametervärden för HBV-modellen i Sverige. Erfarenheter från modellkalibreringar under perioden 1975 - 1989.

29 Urban Svensson och Ingemar Holmström (1990)
Spridningsstudier i Glan.

30 Torbjörn Jutman (1991)
Analys av avrinningens trender i Sverige.

31 Mercedes Rodriguez, Barbro Johansson, Göran Lindström, Eduardo Planos y Alfredo Remont (1991)
Aplicacion del modelo HBV a la cuenca del Río Cauto en Cuba.

32 Erik Arnér (1991)
Simulering av vårflöden med HBV-modellen.

33 Maja Brandt (1991)
Snömätning med georadar och snötaxeringar i övre Luleälven.

34 Bent Göransson, Maja Brandt och Hans Bertil Wittgren (1991)
Markläckage och vattendragstransport av kväve och fosfor i Roxen/Glan-systemet, Östergötland.

35 Ulf Ehlin och Per-Eric Ohlsson, VASO (1991)
Utbyggd hydrologisk prognos- och varningstjänst.
Rapport från studieresa i USA 1991-04-22--30.

36 Martin Gotthardsson, Pia Rystam och Sven-Erik Westman (1992)
Hydrologiska stationsnät 1992/Hydrological network. Svenskt Vattenarkiv.

37 Maja Brandt (1992)
Skogens inverkan på vattenbalansen.

38 Joakim Harlin, Göran Lindström, Mikael Sundby (SMHI) och Claes-Olof Brandesten (Vattenfall Hydropower AB) (1992)
Känslighetsanalys av Flödeskommitténs riktlinjer för dimensionering av hel älv.

39 Sten Lindell (1993)
Realtidsbestämning av arealnederbörd.

40 Svenskt Vattenarkiv (1995)
Vattenföring i Sverige. Del 1. Vattendrag till Bottenviken.

41 Svenskt Vattenarkiv (1995)
Vattenföring i Sverige. Del 2. Vattendrag till Bottenhavet.

42 Svenskt Vattenarkiv (1993)
Vattenföring i Sverige. Del 3. Vattendrag till Egentliga Östersjön.

43 Svenskt Vattenarkiv (1994)
Vattenföring i Sverige. Del 4. Vattendrag till Västerhavet.

44 Martin Häggström och Jörgen Sahlberg (1993)
Analys av snösmältningsförlopp.

45 Magnus Persson (1993)
Utnyttjande av temperaturens persistens vid beräkning av volymsprognoser med HBV-modellen.

46 Göran Lindström, Joakim Harlin och Judith Olofsson (1993)
Uppföljning av Flödeskommitténs riktlinjer.

47 Bengt Carlsson (1993)
Alkalinitets- och pH-förändringar i Ume-älven orsakade av minimitappning.

48 Håkan Sanner, Joakim Harlin and Magnus Persson (1994)
Application of the HBV model to the Upper Indus River for inflow forecasting to the Tarbela dam.

49 Maja Brandt, Torbjörn Jutman och Hans Alexandersson (1994)
Sveriges vattenbalans. Årsmedelvärden 1961 - 1990 av nederbörd, avdunstning och avrinning.

50 Svenskt Vattenarkiv (1994)
Avrinningsområden i Sverige. Del 3. Vattendrag till Egentliga Östersjön och Öresund.

51 Martin Gotthardsson (1994)
Svenskt Vattenarkiv. Översvämningskänsliga områden i Sverige.

52 Åsa Evremar (1994)
Avdunstningens höjdberoende i svenska fjällområden bestämd ur vattenbalans och med modellering.

53 Magnus Edström och Pia Rystam (1994)
FFO - Stationsnät för fältforskningsområden 1994.

54 Zhang Xingnan (1994)
A comparative study of the HBV model and development of an automatic calibration scheme.

55 Svenskt Vattenarkiv (1994)
Svenskt dammregister - Södra Sverige.

56 Svenskt Vattenarkiv (1995)
Svenskt dammregister - Norra Sverige.

57 Martin Häggström (1994)
Snökartering i svenska fjällområdet med NOAA-satellitbilder.

58 Hans Bertil Wittgren (1995)
Kvävetransport till Slätbaken från Söderköpingsåns avrinningsområde

59 Ola Pettersson (1995)
Vattenbalans för fältforskningsområden.

60 Barbro Johansson, Katarina Losjö, Nils Sjödin, Remigio Chikwanha and Joseph Merka (1995)
Assessment of surface water resources in the Manyame catchment - Zimbabwe.

61 Behzad Koucheki (1995)
Älvtemperaturers variationer i Sverige under en tioårsperiod.

62 Svenskt Vattenarkiv (1995)
Sänkta och torrlagda sjöar.

63 Malin Kanth (1995)
Hydrokemi i fältforskningsområden.

64 Mikael Sundby, Rikard Lidén, Nils Sjödin, Helmer Rodriguez, Enrique Aranibar (1995)
Hydrometeorological Monitoring and Modelling for Water Resources Development and Hydropower Optimisation in Bolivia.

65 Maja Brandt, Kurt Ehlert (1996)
Avrinningen från Sverige till omgivande hav.

66 Sten Lindell, Håkan Sanner, Irena Nikolushkina, Inita Stikute (1996)
Application of the integrated hydrological modelling system IHMS-HBV to pilot basin in Latvia

67 Sten Lindell, Bengt Carlsson, Håkan Sanner, Alvina Reihan, Rimma Vedom (1996)
Application of the integrated hydrological modelling system IHMS-HBV to pilot basin in Estonia

68 Sara Larsson, Rikard Lidén (1996)
Stationstäthet och hydrologiska prognoser.

69 Maja Brandt (1996)
Sedimenttransport i svenska vattendrag exempel från 1967-1994.

70 Svenskt Vattenarkiv (1996)
Avrinningsområden i Sverige. Del 4. Vattendrag till Västerhavet.

71 Svenskt Vattenarkiv (1996)
Svenskt sjöregister. 2 delar

72 Sten Lindell, Lars O Ericsson, Håkan Sanner, Karin Göransson SMHI
Malgorzata Mierkiewicz, Andrzej Kadlubowski, IMGW (1997)
Integrated Hydrological Monitoring and Forecasting System for the Vistula River Basin. Final report.

73 Maja Brandt, Gun Grahn (1998)
Avdunstning och avrinningskoefficient i Sverige 1961-1990. Beräkningar med HBV-modellen.

74 Anna Eklund (1998)
Vattentemperaturer i sjöar, sommar och vinter - resultat från SMHIs mätningar.

75 Barbro Johansson, Magnus Edström, Katarina Losjö och Sten Bergström (1998)
Analys och beräkning av snösmältningsförlopp.

76 Anna Eklund (1998)
Istjocklek på sjöar.

77 Björn Bringfelt (1998)
An evapotranspiration model using SYNOP weather observations in the Penman-Monteith equation

78 Svenskt Vattenarkiv (1998)
Avrinningsområden i Sverige. Del 2 Vattendrag till Bottenhavet.

79 Maja Brandt, Anna Eklund (1999)
Snöns vatteninnehåll Modellberäkningar och statistik för Sverige

80 Bengt Carlsson (1999)
Some facts about the Torne and Kalix River Basins.
A contribution to the NEWBALTIC II workshop in Abisko June 1999.

81 Anna Eklund (1999)
Isläggning och islossning i svenska sjöar.

82 Svenskt Vattenarkiv (2000)
Avrinningsområden i Sverige. Del 1. Vattendrag till Bottenviken.

83 Anna Eklund, Marie Gardelin, Anders Lindroth (2000)
Vinteravdunstning i HBV-modellen - jämförelse med mätdata

84 Göran Lindström, Mikael Ottosson Löfvenius (2000)
Tjäle och avrinning i Svartberget – studier med HBV-modellen

85 Bengt Carlsson och Göran Lindström (2001)
HBV-modellen och flödesprognoser

86 Josef Källgården (2001)
Snow distribution in a mountainous region. A remote sensing study.

87 Johan Andréasson, Anders Gyllander, Barbro Johansson, Josef Källgården, Sten Lindell, Judith Olofsson, Angela Lundberg (2001)
Snötaxering med georadar - Bättre vårflödesprognoser med HBV-modellen?

88 Deliang Chen, Barbro Johansson
Temperaturens höjdberoende – En studie i Indalsälvens avrinningsområde.

89 Agne Lärke, Håkan Sanner, Anna Johnell (2003)
Utvärdering av SMHI:s prognos- och varningstjänsts verksamhet under flödena januari t o m mars 2002 i sydvästra Sverige

90 Barbro Johansson, Johan Andreasson och Johan Jansson (2003)
Satellite data on snow cover in the HBV model. Method development and evaluation

Sveriges meteorologiska och hydrologiska institut
601 76 Norrköping
Tel 011 -495 80 00 · Fax 011-495 80 01

ISSN 0283-7722