AQUATIC RESEARCH

Aquatic Research 2(2), 92-133 (2019) • https://doi.org/10.3153/AR19010 E-ISSN 2618-6365

Review Article

MODELLING THE PROGRESS AND EFFECTS OF EUTROPHICATION IN INLAND AND COASTAL WATERS

Ali Ertürk

Cite this article as:

Ertürk, A. (2019). Modelling the progress and effects of eutrophication in inland and coastal waters. *Aquatic Research*, 2(2), 92-133. https://doi.org/10.3153/AR19010

Istanbul University Faculty of Aquatic Sciences, Ordu Caddesi No:8 Laleli Fatih, Istanbul-Turkey

ORCID IDs of the authors:

A.E. 0000-0002-3532-2961

Submitted: 29.03.2019

Accepted: 01.04.2019

Published online: 12.04.2019

Correspondence:

Ali ERTÜRK

E-mail: erturkali@istanbul.edu.tr

Available online at

http://aquatres.scientificwebjournals.com

ABSTRACT

The aim of this paper is to give a detailed overview on the predictive-model building/coding techniques for simulating the progress and effects of eutrophication based on differently detailed model structures. First; historical development of predictive eutrophication modelling is reviewed. Then, a generic transport model that can be coupled with any eutrophication kinetics is described. In the following sections, ecological sub models based on eutrophication kinetics and food-web are described along with the bottom-up approach based linkage of nutrient kinetics, primary production and transfer of food to higher trophic levels are demonstrated together with an example case study based on previous studies. Finally, the paper is supported by two comprehensive appendices, one that guides the interested readers how to develop a simple eutrophication modelling tool from starch and another to that summarizes an example hydrodynamic model development for forcing the flow fields in the transport model described in this paper.

Keywords: Eutrophication, Model building, Ecological modelling, Water quality

Introduction

Mathematical models are theoretical constructs, together with assignment of numerical values to model parameters, incorporating some prior observation and data from field and/or laboratory and relating external inputs and forcing functions to system variable responses. Models can be defined as idealized formulations that represent the response of a physical system to external forcing. The cause-effect relationship between loading and concentration depends on the physical, chemical, and biological characteristics of the receiving water. In environmental science, ecological models are used to evaluate the potential impacts of external forcing factors and to understand the functioning of the system (Thomann and Mueller, 1987; Chapra, 1997; Arhonditsis and Brett, 2004). They are useful tools to get a holistic picture of ecosystems, fill in the gaps in field data or forecast the systems responses to different external forcings. Models can produce many instantaneous pictures of the ecosystem by spatially and temporally interpolation between monitoring data points, allow testing of hypotheses on how the ecosystem is functioning, forecast the ecosystem behaviour and give relatively fast answers to scientists, engineers and managers

Predictive Eutrophication Analysis Models

Historically, aquatic ecological modelling studies were initiated with simple models of nutrient cycles in fresh water ecosystem in late 1960s and early 1970s, when the focus on dissolved oxygen deficiency as the main environmental problem in aquatic ecosystems was shifted to the problems caused by excess nutrient inputs into aquatic ecosystems. The first models were relatively simple consisting only of simple nutrient balances (such as the ones shown in Equation 1) with assumptions such as completely mixed system, steady state conditions, representing a seasonal or annual average prevail, limiting nutrient being phosphorus only where total phosphorus is used as a measure of trophic status. An example of such models is given in Equation 1 and Equation 2,

$$V\frac{dP}{dt} = W - v_S A_S P - Q_{OUT} P$$

(Equation 1)

where V is volume [L^3], P is the total phosphorus concentration [$M\cdot L^{-3}$] Q_{OUT} is the outflow [$L^3\cdot T^{-1}$], A_S is the surface area [L^{-2}], v_S is the settling velocity [$L\cdot T^{-1}$] and W is the external sources for phosphorus [$M\cdot T^{-1}$]. Most of the analyses were done for steady state; hence, equations such as Equation 2 were used instead of Equation 1

$$P = \frac{W}{Q_{OUT} + v_S A_S}$$

(Equation 2)

Another type of simple models used in those years were empirical models that were derived by various researchers using curve fitting techniques, such as the ones listed below (N : Total nitrogen [$\mu g.l^{-1}$], P : Total phosphorus [$\mu g\cdot L^{-1}$], chl-A : Chlorophyll-A [$\mu g\cdot L^{-1}$]):

- Dillon and Rigler (1974)

$$\log(\text{chl - A}) = 1.449\log(P) - 1.136$$

(Equation 3)

- Bartsch and Gakstatter (1978)

$$\log(\text{chl - A}) = 0.807\log(P) - 0.194$$

(Equation 4)

- Rast and Lee (1978)

$$\log(\text{chl - A}) = 0.76\log(P) - 0.259$$

(Equation 5)

- Smith and Shapiro (1981)

$$\log(\text{chl - A}) = 1.55\log(P) - \log_{10}\left[\frac{6.404}{0.0204(N/P) + 0.334}\right]$$

(Equation 6)

where log and log10 are the natural and general algorithms respectively.

The trend considering the eutrophication environmental problem based on as lasted until 1980's. Therefore, extensive research was initiated on nutrients in aquatic ecosystems (O'Connor et al., 1968; Bloesch et al., 1977; Edmonson, 1979). Incorporation of nutrient cycles into water quality models necessitated introduction of new state variables such as Org-N, NH_4^+-N, NO_3^--N, Org-P, $PO4^{3-}$-P, phytoplankton biomass, etc. and chemical/biochemical processes. In other words, more complex models than were needed. Developments in the computer technology enabled scientists and engineers to design and develop these models. Models developed and used by Di Toro et al. (1971); Thomann et al. (1975) and Di Toro and Connolly (1980) are examples of such models. These models did not consider the aquatic ecosystem as fully mixed anymore. They were the first examples of box models and are considered as predecessors of modern nutrient dynamics modelling tools described in the following paragraphs.

WQRRS (Water Quality for River and Reservoir Systems)**,** is a one dimensional dynamic model which calculates the temporal variations of state variables in vertical dimension (z). WQRRS was developed by the United States Army Corps of Engineers (USACE), Hydraulic Engineering Centre (HEC, 1978). The model is designed to simulate nutrient dynamics in river and reservoir systems however state variables covered in the WQRRS make it also useful for ecological modelling in other aquatic ecosystems. Nutrients, phytoplankton, zooplankton, fish, and benthic organisms can be simulated by the model. CE-QUAL-R1 (Environmental Laboratory, 1995) is derived from this model can also simulate the sulphur cycle, iron and manganese under aerobic and anaerobic conditions.

Water Quality Analysis Simulation Program (WASP) (Di Toro et al., 1983; Ambrose et al., 1993; Wool et al., 2001) was developed by United States Environmental Protection Agency (USEPA). WASP covers transportation dynamics of advection-dispersion and suspended sediment transport. The model describes six transport fields; water column, water in sediment blanks, user defined settling and resuspension velocities in water for three sediment groups, and transportation due to precipitation and evaporation. WASP is a box model it is possible to generate 0, 1, 2, and 3 dimensional model networks depending on the number and topology of the boxes. Several hydrodynamic modelling software such as DYNHYD5, RIVMOD (Hosseinipour et al., 1990), SED3D (Sheng et al, 1991), and EFDC (Hamrick, 1996) can produce outputs, which can be used by WASP through external hydrodynamic linkage.

CE-QUAL-W2 (Cole and Wells, 2006) is a two-dimensional model which does both hydrodynamic and water quality simulations in longitudinal and vertical dimensions (x, z). State variables constituted in the model are temperature, salinity, dissolved oxygen, CBOD, organic material composed of carbon, nitrogen, and phosphorus (dissolved and labile, dissolved and refractory, particulate and labile, particulate and refractory), ammonia nitrogen, nitrate nitrogen, phosphorus, dissolved and particulate silica, and unlimited number of phytoplankton, zooplankton, epiphyte and rooted aquatic macrophyte groups.

CE-QUAL-ICM (Cerco and Cole, 1994; Cerco and Cole, 1995) is capable of simulating sediment processes in detail. However, it only includes water quality codes and to run the model output codes of CH3D hydrodynamic model, which is also developed by USACE, is necessary. Together with the CH3D, CE-QUAL-ICM can make water quality simulations in three spatial dimensions. This model is also known as the Chesapeake Bay model. Chesapeake Bay (United States of America) was modelled intensively from 80's up today. Many ecological modelling studies conducted for the Chesapeake Bay (Di Toro and Fitzpatrick, 1993; USACE, 2000; Schaffner, et al., 2002; Xu, 2005; Galgeos et al., 2006) contributed to the ecological modelling science and the literature. These models did not only consider pelagic nutrient cycles and primary production but also benthic fluxes, zooplankton and filtrating organisms.

COHERENCE (Luyten et al., 1999) is a three dimensional hydrodynamic ecological model which was developed by the Management Unit of the Mathematical Models of the North Sea (MUMM) to use it in North Sea. ERSEM (European Regional Seas Ecosystem Model) (Paetsch, 2001) is developed by European Union for applications in North Sea. It is an advanced model including detailed description of pelagic and benthic dynamics.

In mid 70's another branch of ecological modelling was initiated. First examples of food web models that are designed to mimicking the trophic networks (Jansson, 1974; Jansson, et al., 1982; Polovina, 1984a; Polovina, 1984b) were used for research purposes. Unlike the most of the biogeochemical or nutrient dynamics models, which consider the nutrient cycles and primary production more detailed, trophic network models use relatively simplified approaches to consider them, or they accept them as model input rather than state variables. Trophic network models are equipped with algorithms for dealing with higher trophic levels and balancing the energy and matter in a user defined trophic network. Organisms in higher trophic levels such as fishes and macro invertebrates are good environmental indicators to track environmental health and ecological changes as adaptive response to stress, especially in estuaries and lagoons (USEPA, 2000; Villanueva, et al. 2006) and therefore food network models that can simulate these organisms are valuable tools for ecological assessment of those ecosystems. These models have been applied to transitional aquatic ecosystem such as coastal lagoons (Hull, et al., 2000; Gamito and Erzini, 2005; Villanueva, et al. 2006).

Coupling the nutrient dynamics and trophic network models provides the opportunity to benefit from the advantages of both frameworks. This topic was discussed by Mergey, et al.

(2001) and the Royal Comission on Environmental Pollution (2004). Tillmann et al. (2006) coupled CE-QUAL-ICM (Cerco and Cole, 1994) with EwE (Christensen, et al. 2005) and applied the coupled models to the Cheasepeake Bay.

Development of these models took years of study and research efforts. Appendix-A gives an insight to the reader by illustrating how a simple eutrophication model could be developed from scratch.

Modelling of Transport for Inland and Coastal Waterbodies

Some aquatic ecosystems are either too large in lateral dimensions or too deep so that they should not be considered as completely mixed. If this is the case, a model, which assumes that the ecosystem is completely mixed (such as the simple eutrophication model discusses in the previous section) should not be applied directly. For partly mixed aquatic ecosystem, the advection-dispersion-reaction equation given below should be applied.

$$\frac{\partial C}{\partial t} = -u\cdot\frac{\partial C}{\partial x} + D_x\cdot\frac{\partial^2 C}{\partial x^2} - v\cdot\frac{\partial C}{\partial y} + D_y\cdot\frac{\partial^2 C}{\partial y^2} - w\cdot\frac{\partial C}{\partial z} + D_z\cdot\frac{\partial^2 C}{\partial z^2} + \sum k\cdot C - \upsilon_{sedimentation}\cdot\frac{\partial C}{\partial z} \pm \text{external sources and sinks}$$

(Equation 7)

The terms used in Equation 7 are given below

x, y, z : Spatial coordinates [L]

u, v, w : Flow velocities in x, y, z directions respectively [$L\cdot T^{-1}$]

Dx, Dy, Dz : Dispersion coefficients in x, y, z directions respectively [$L^2\cdot T^{-1}$]

C : Concentration [$M\cdot L^{-3}$]

$\sum k\cdot C$: Reaction kinetics partial derivative [$M\cdot L^{-3}\cdot T^{-1}$]

Figure 1 provides a detailed description of the advection-dispersion-reaction equation. The reaction kinetics partial derivative corresponds to the content of the kinetic function in the example model described in the next section. All the biogeochemical and ecological interactions to be modelled are written into this partial derivative. Some model variables such as phytoplankton or detritus can settle. This is handled by the partial derivative of settling. External sources and sinks are important to represent the effect of point and non-point sources of loads. The velocities u, v and w can be calculated using a hydrodynamic model. An example hydrodynamic model is given in Appendix B.

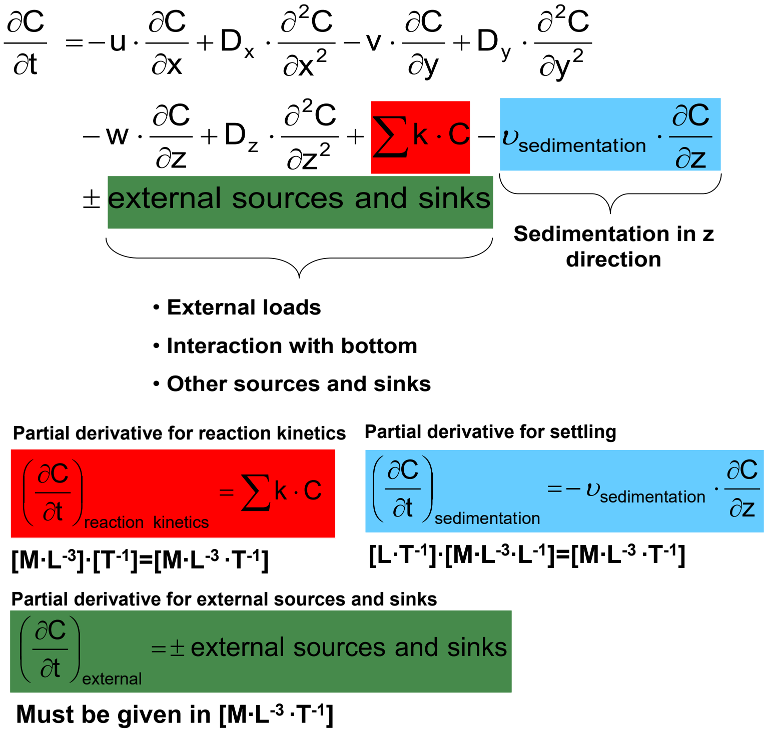

$\frac{\partial C}{\partial t} = -u\cdot\frac{\partial C}{\partial x} + D_x\cdot\frac{\partial^2 C}{\partial x^2} - v\cdot\frac{\partial C}{\partial y} + D_y\cdot\frac{\partial^2 C}{\partial y^2} - w\cdot\frac{\partial C}{\partial z} + D_z\cdot\frac{\partial^2 C}{\partial z^2} + \sum k\cdot C - \upsilon_{sedimentation}\cdot\frac{\partial C}{\partial z}$
± external sources and sinks
Sedimentation in z direction
• External loads
• Interaction with bottom
• Other sources and sinks
Partial derivative for reaction kinetics
$\left(\frac{\partial C}{\partial t}\right)_{reaction\ kinetics} = \sum k\cdot C$
$[M\cdot L^{-3}]\cdot[T^{-1}]=[M\cdot L^{-3}\cdot T^{-1}]$
Partial derivative for settling
$\left(\frac{\partial C}{\partial t}\right)_{sedimentation} = -\upsilon_{sedimentation}\cdot\frac{\partial C}{\partial z}$
$[L\cdot T^{-1}]\cdot[M\cdot L^{-3}\cdot L^{-1}]=[M\cdot L^{-3}\cdot T^{-1}]$
Partial derivative for external sources and sinks
$\left(\frac{\partial C}{\partial t}\right)_{external} = \pm$ external sources and sinks
Must be given in $[M\cdot L^{-3}\cdot T^{-1}]$

Figure 1. The advection-dispersion-reaction equation

Ecological Sub-Models for Prediction of the Progress and Effects of Eutrophication

Eutrophication is a complicated process that includes many ecological components and processes in addition to nutrients and primary production. A model designed for detailed and realistic eutrophication analysis should contain those components and processes. The model developed in Appendix A would be too simplified for such eutrophication analyses. Spatial variability and more advanced representation of transport processes should be incorporated into such a model as well. This section aims to instruct the reader how to construct this type of eutrophication models.

Development of Biogeochemical Cycle Sub-Models for Eutrophication Analyses

Biogeochemical cycle sub-models simulate processes that run among the biotic and abiotic components of the ecosystem.

These sub-models could be as simple incorporating free nutrients (N, P compounds), organic matter and nutrients bound to it and a single group of phytoplankton or as complicated as incorporating more detailed representation of nutrients (N, P, Si compounds) other inorganic compounds (S, Fe, Mn with different ionic states), detailed representation of detritus, multiple groups of phytoplankton, multiple groups of zooplankton and fish, benthic organisms, sediment diagenesis, macrophytes, bacteria, etc. In this section, a biogeochemical sub-model that is moderately complicated will be described. The model includes 22 state variables namely; NH4 and NO3 Nitrogen, PO4 Phosphorus, Available Silicon, Inorganic Carbon, Dissolved Oxygen, Diatoms, Cyanobacteria and Other Planktonic Algae Carbon, Zooplankton Carbon, External Labile Dissolved Org Carbon, External Labile Particulate Detritus Carbon, External Refractory Dissolved Organic Carbon, External Refractory Particulate Detritus Carbon, Diatoms based Dissolved Organic Carbon, Diatoms based Particulate Detritus Carbon, Other Planktonic Algae based Dissolved Organic Carbon, Other Planktonic Algae based Particulate Detritus Carbon, Cyanobacteria based Dissolved Organic Carbon, Cyanobacteria based Particulate Detritus Carbon, Zooplankton based Dissolved Organic Carbon and Zooplankton based Particulate Detritus Carbon.

A model with these state variables is classified as an NPZD (Nutrients Phytoplankton Zooplankton Detritus) model. The state variables and processes letting them interact with each other are illustrated in Figures 2 to 7. As seen in Figure 2, nitrogen is assumed to be in three main pools by the example NPZD model. The first of them is ammonia nitrogen, the second is the nitrate nitrogen and the third is nitrogen bound to molecules found in living (phytoplankton and zooplankton) and dead organic matter. Phosphorus is assumed to be in two main pools. The first of them is phosphate phosphorus and the second is phosphorus bound to molecules found in living (phytoplankton and zooplankton) and dead organic matter similar to nitrogen. Silicon is assumed to be in two main pools. The first of them is available silica silicon (dissolved inorganic silicon) and the second is silicon found with living (diatoms and zooplankton feeding on diatoms) and dead (diatoms and zooplankton based organic carbon and detritus) organic matter. As seen in Figure 3, Dissolved oxygen is dissolved oxygen is interacting with most of the other state variables in the example NPZD model. Carbon cycle is modelled extensively by the example NPZD model. External labile dissolved organic carbon, external labile particulate detritus carbon, external refractory dissolved organic carbon and external refractory particulate detritus carbon are used to model the allochtonous organic carbon and detritus carbon. The autochthonous detritus carbon is simulated using other planktonic algae based dissolved organic carbon, other planktonic algae based particulate detritus, diatoms based dissolved organic carbon, diatoms based particulate detritus, cyanobacteria based dissolved organic carbon, cyanobacteria based particulate detritus, zooplankton based dissolved organic carbon and zooplankton based particulate detritus. Representation of inorganic carbon cycle is illustrated in Figure 2. Three phytoplankton groups (diatoms, cyanobacteria and other planktonic algae) and one zooplankton group (resembling total zooplankton) are simulated by the NPZD model. The equations and other details of the NPZD model would be too space consuming to give here. The reader is referred to Erturk (2008) and Erturk et al (2015) for more detailed information and complete set of equations. As seen in Figures 2 to 7, the example NPZD model is designed to keep track from the inorganic nutrients up to the zooplankton biomass and back to inorganic nutrients via detritus and its decomposition. Organically bound nutrients are coupled within the detritus cycle so that they are no separate state variables representing them. The model can be used to identify the contribution of each plankton group to autochthonous organic matter hence analyse the eutrophication process in detail.

Development of Foodweb Sub-Models

Trophic network is defined as a set of interconnected food chains, by which energy is materials circulate within an ecosystem. The classical food web can be divided into two broad categories: the grazing web, which starts with primary producers and ends at top predators and the detrital web which starts with detritus, continues over decomposers (bacteria, fungi, etc.) and detrivores and ends at their predators. Unlike the biogeochemical sub-models, the foodweb sub-models are usually more specific to the system for which they are developed. This is because, each system has a different combination of complex behaving organisms on the higher levels of the trophic network. Depending on the aim of model development, a group of organisms, a particular species or a development stage within a species can be state variables of a food web model. To construct a food-web model, two components are needed: The basic knowledge about the food-web of the ecosystem for which the model is developed and the modelling tools. There are many tools for developing foodweb models.

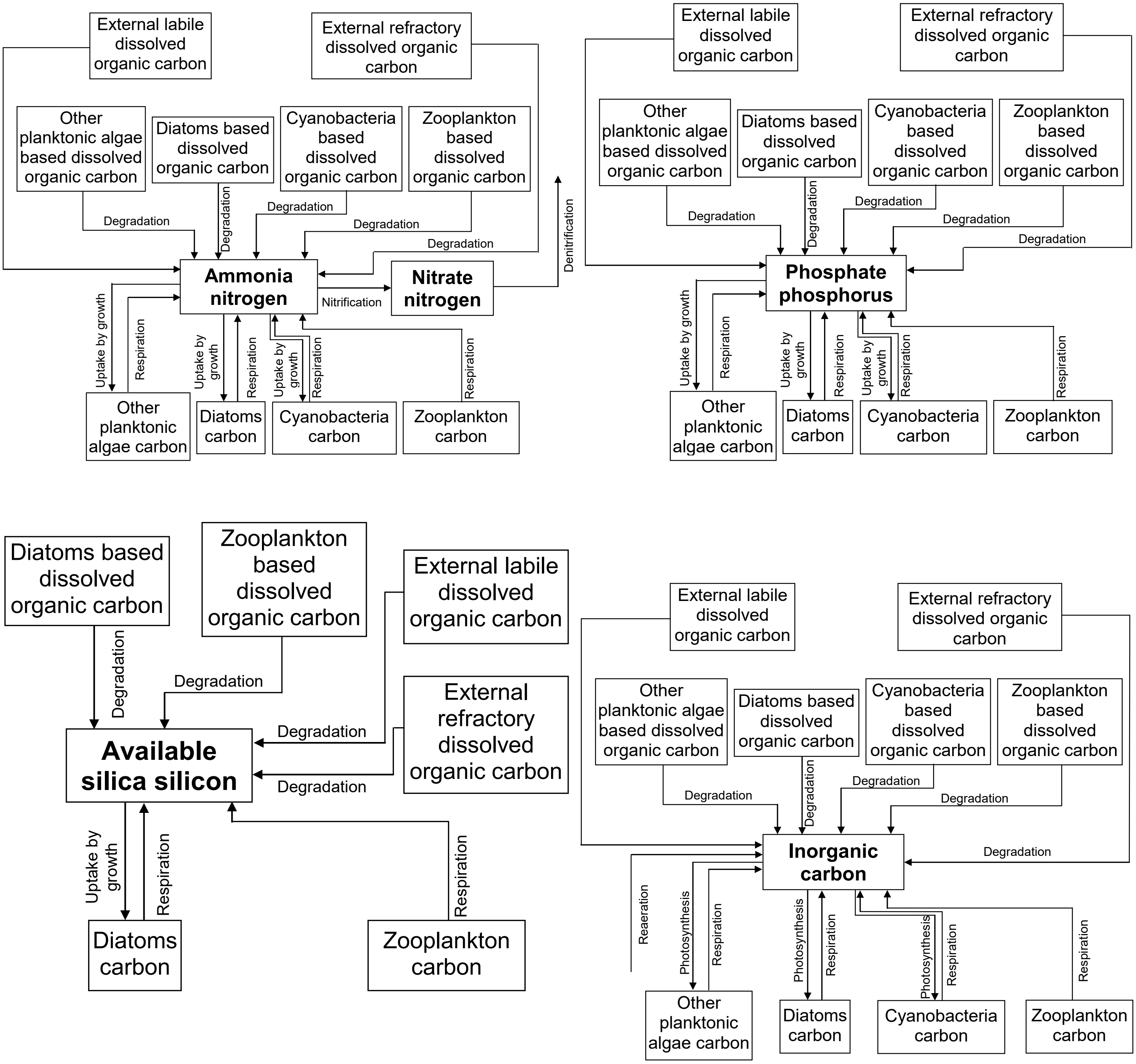

External labile dissolved organic carbon
External refractory dissolved organic carbon
Other planktonic algae based dissolved organic carbon
Diatoms based dissolved organic carbon
Cyanobacteria based dissolved organic carbon
Zooplankton based dissolved organic carbon
Degradation
Denitrification
Ammonia nitrogen
Nitrification
Nitrate nitrogen
Uptake by growth
Respiration
Other planktonic algae carbon
Diatoms carbon
Cyanobacteria carbon
Zooplankton carbon
Phosphate phosphorus
Available silica silicon
Inorganic carbon
Reaeration
Photosynthesis

Figure 2. Nutrient cycles in the NPZD model

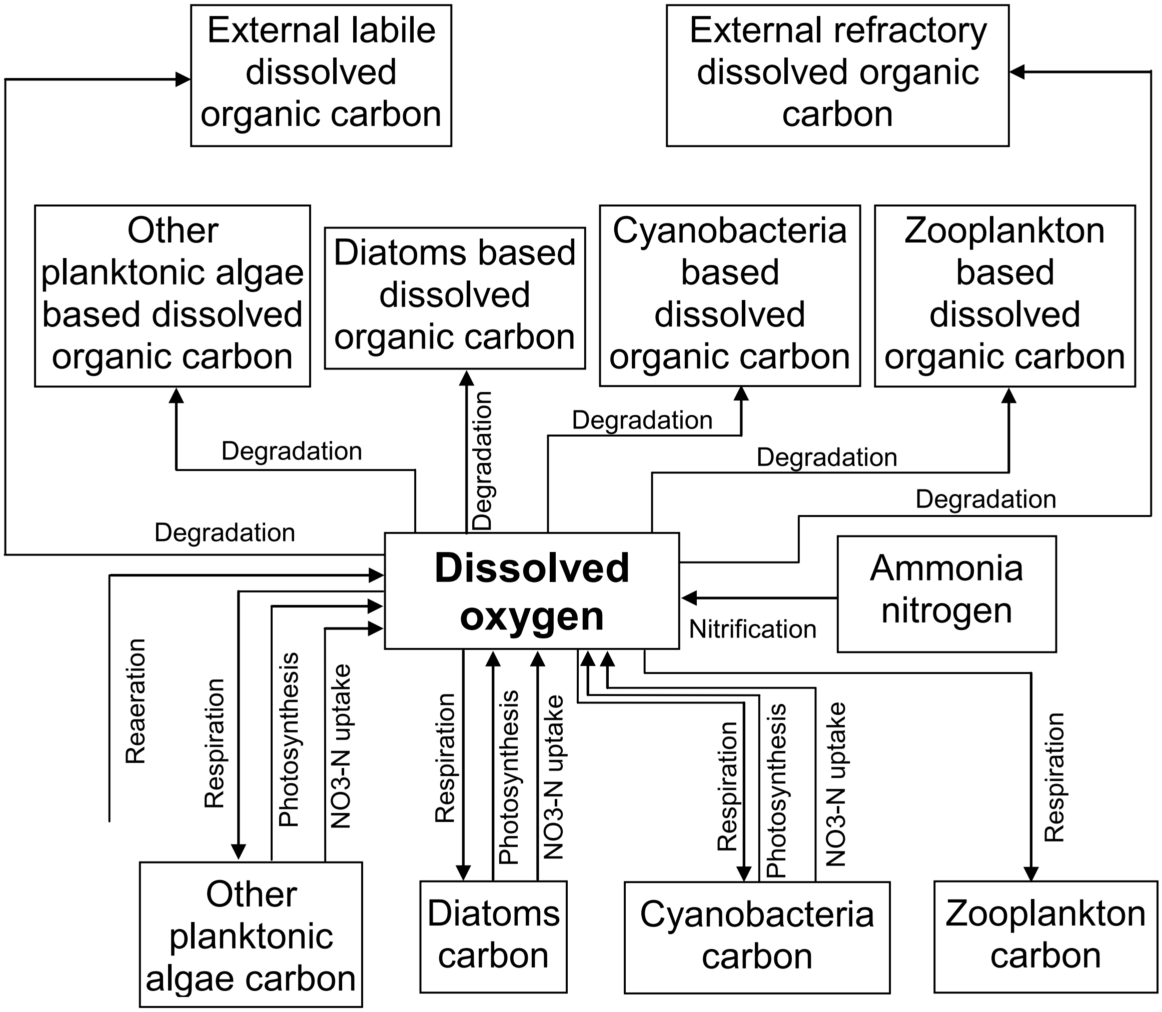


Figure 3. Dissolved oxygen cycle

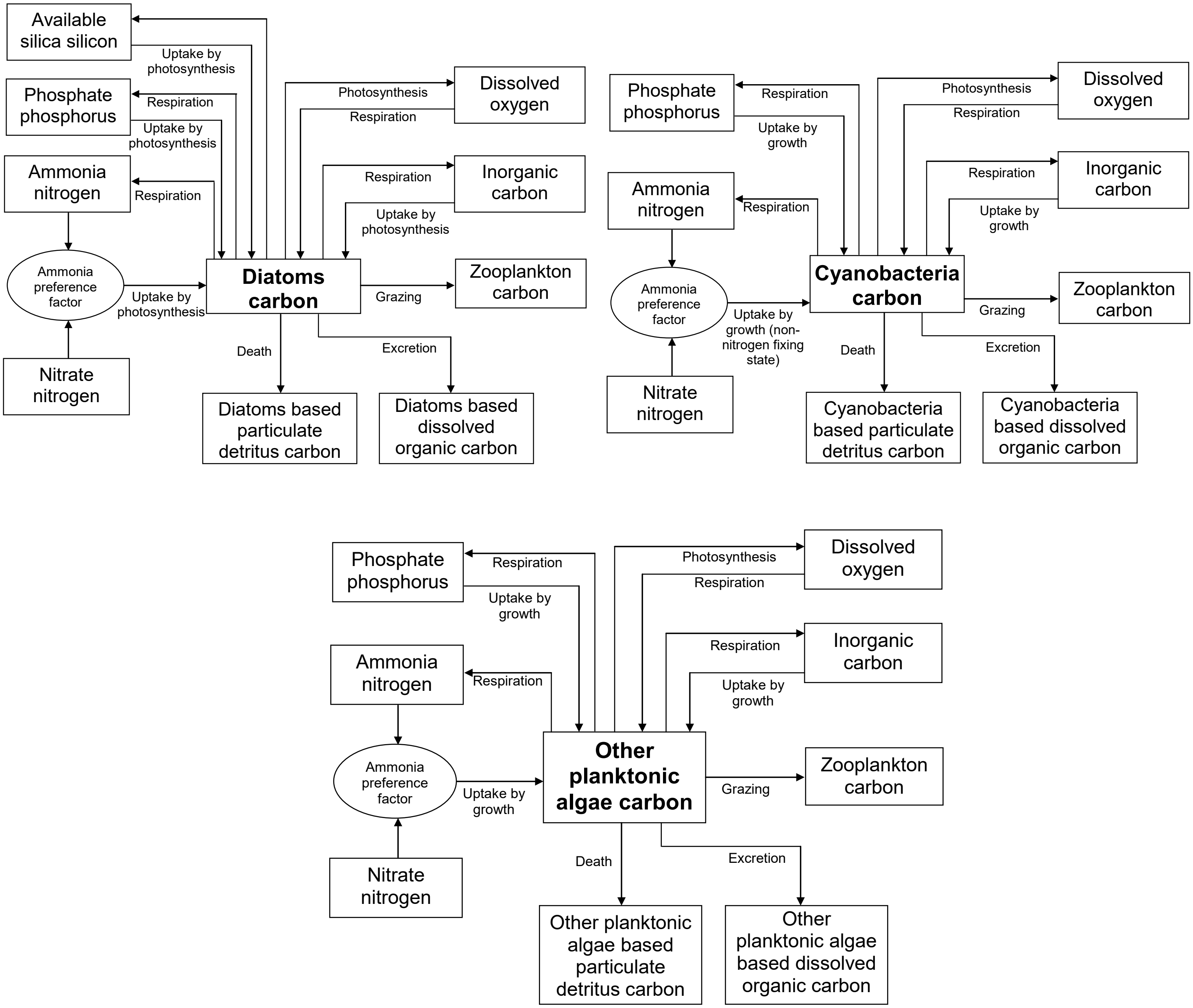


Figure 4. Phytoplankton in the NPZD model

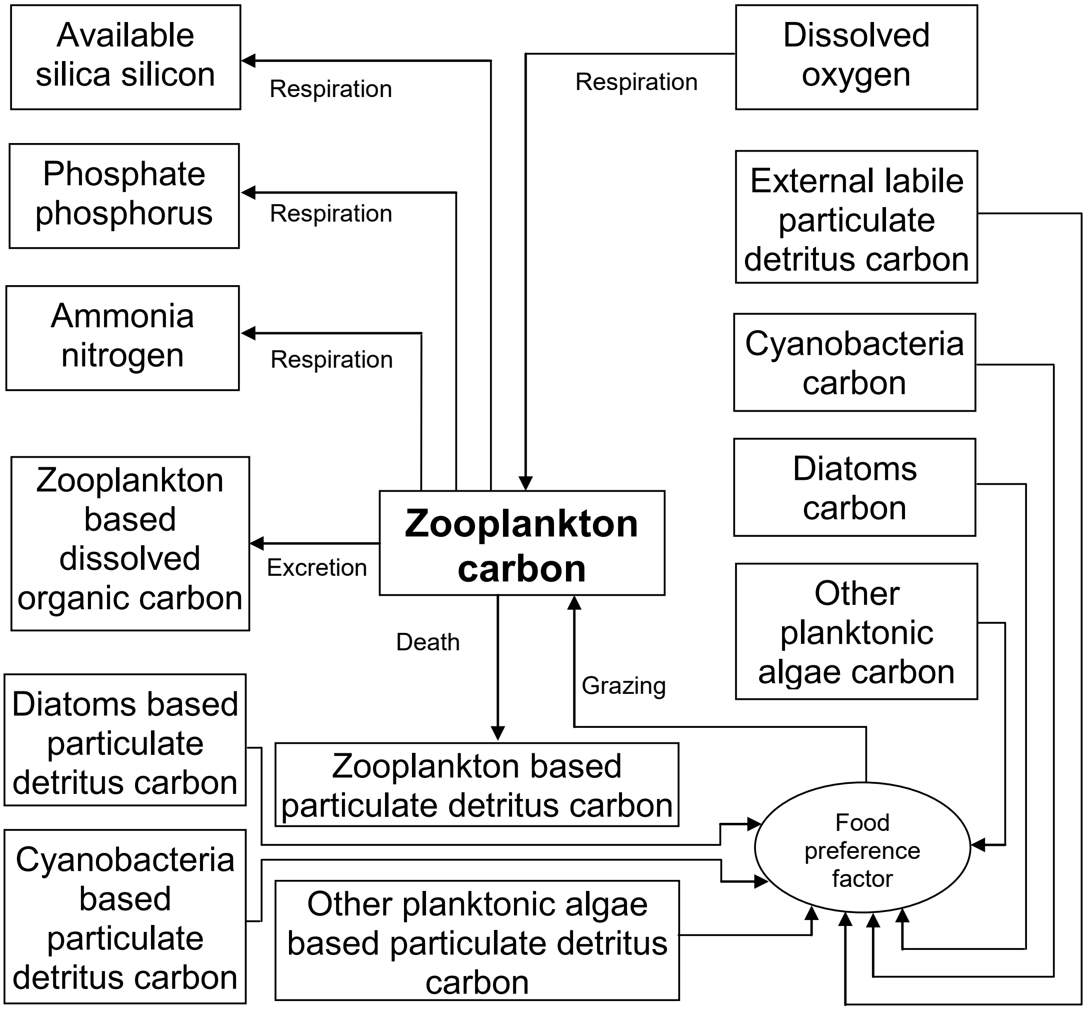

Available silica silicon
Respiration
Phosphate phosphorus
Respiration
Ammonia nitrogen
Respiration
Dissolved oxygen
Respiration
External labile particulate detritus carbon
Cyanobacteria carbon
Diatoms carbon
Other planktonic algae carbon
Zooplankton based dissolved organic carbon
Excretion
Zooplankton carbon
Death
Grazing
Diatoms based particulate detritus carbon
Zooplankton based particulate detritus carbon
Cyanobacteria based particulate detritus carbon
Other planktonic algae based particulate detritus carbon
Food preference factor

Figure 5. Zooplankton in the NPZD model

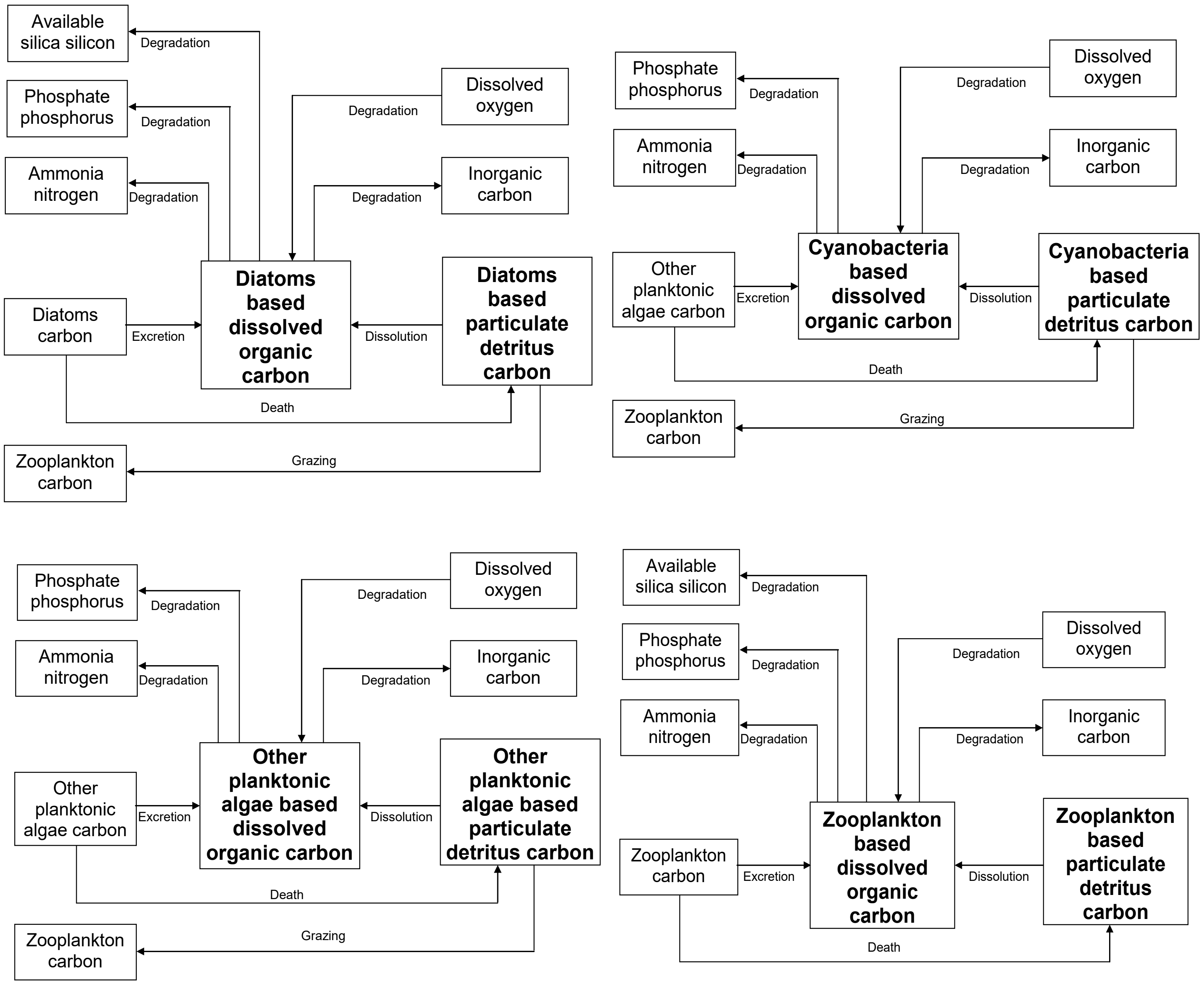


Figure 6. Authocthounus organic matter cycle in the NPZD model

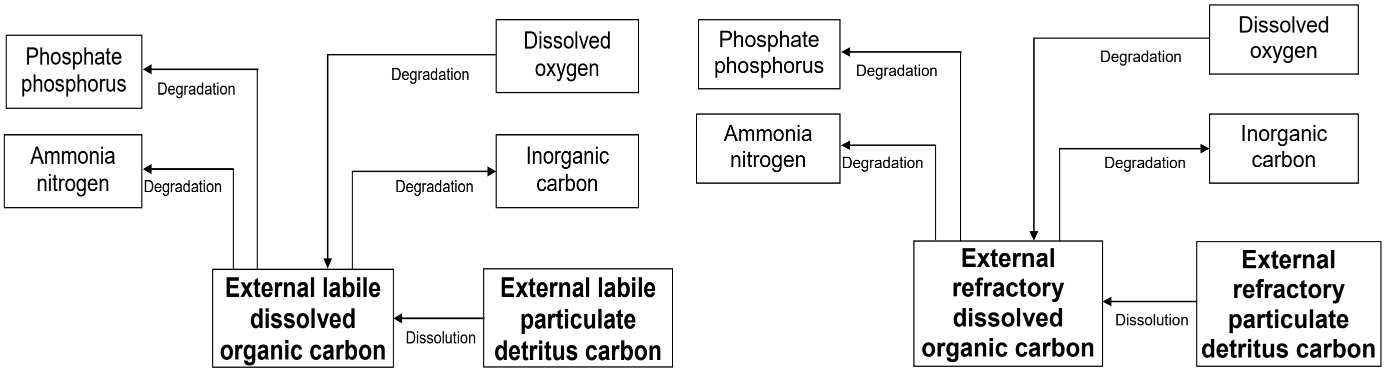


Figure 7. Allochtonous organic carbon and detritus cycles in the NPZD model

Ecopath with Ecosim that is optimized for aqutic ecosystems will be described in this section as an example. Ecopath with Ecosim is designed for straightforward construction, parameterization and analysis of mass-balance trophic models for various ecosystems. The core of Ecopath is derived from ECOPATH program developed by Polovnia and Ow (1983). However, Ecopath does not work under the steady state assumption any more. Instead, it is bases the parameterization on an assumption of mass balance of an arbitrary period (Chiristensen et al., 2005). This period is usually one year, but modelling an ecosystem seasonally is also possible. Ecopath allows the user to develop a generic model for any ecosystem, which can contain any number of state variables. In Ecopath terminology, a state variable is called as group or box. A box (group) in an Ecopath model can be a group of ecologically related species, a single species, or a single size/age group of given species. Since the original ECOPATH from early 1980s, Ecopath has undergone a long development process for both; the theory, ideas and as well as the software itself. The system has been optimized for direct use in fisheries assessment as well as for addressing other more general environmental questions through the inclusion of the temporal dynamic model Ecosim and spatial dynamic model Ecospace. Furthermore, tools such as Ecoranger (tool for addressing uncertainty), Ecoempire (tool for calculation of empirical relationships of production over biomass ratios), Flow diagram (tool for plotting the defined trophic network) or Ecowrite (reporting tool) ease and enhance the model development (Christensen et al., 2005). Different versions of Ecopath with Ecosim are used for various studies with topics such as analyses of trophic interactions (Opiz, 1996; Okey and Pauly, 1999; Harvey et al. 2003), trophic modelling for aquatic ecosystems (Aydin et al., 2003; Mohamed et al, 2005), fisheries management and fish stock assessment (Pauly, 1998; Fayram 2005) in different aquatic ecosystems. Being applied to different aquatic ecosystems from the tropics up to Arctics, Ecopath with Ecosim is proven to be reliable. Detailed information related to methods used in, Ecopath, Ecosim and Ecospace as well as capabilities and limitations of these models is given by Walters et al. (1999), Walters et al. (2000), Pauly et al. (2000), Christensen and Walters (2004), Kavanagah et al. (2004) and Christensen et al., (2005). Ecopath has two master equations. The first equation describes the production and second equation describes the energy balance for each modelled group the energy balance via consumption. The first master equation of Ecopath (Equation 8) describes how the production term for each group modelled can be split into components.

In mathematical terms, the first master equation is written as in Equation 9, where i is the index for the relevant group, P_i is the total production rate of group i, Y_i is the total fishery catch rate of group i, $M2_i$ is the total predation rate for group i, B_i the biomass of the group i, E_i the net migration rate (emigration – immigration), BA_i is the biomass accumulation rate for group i, while $M0_i = P_i\,(1\text{-}EE_i)$ is the ‘other mortality’ rate for group i and EE_i is the ecotrophic efficiency of group i. Equation 9 can be rearranged as Equation 10 and rewritten as Equation 11.

In Equation 11; j is the index for prey, P/B_i is the production/biomass ratio, Q/B_i is the consumption/biomass ratio and $DC_{j,i}$ is the fraction of prey j in the average diet of predator i (diet composition). A system of n linear equations (Equation 12) is obtained from Equation 12 for a trophic system with n groups.

Ecopath includes algorithms to solve this system of linear equation for one of following variables for each group: biomass (B), production/biomass ratio (P/B), consumption/biomass ratio (Q/B) or ecotrophic efficiency (EE). The energy input and output of all living groups must be balanced in a model. When balancing the energy for a living group additional terms, which do not exist in the first master equation, are needed and with their incorporation, the second master equation of Ecopath (Equation 13) is formed.

$$\begin{aligned}\text{Production} &= \text{catches} + \text{mortality by predation} + \text{biomass accumulation} + \\ &\quad \text{net migration} + \text{other mortality}\end{aligned} \qquad \text{(Equation 8)}$$

$$P_i = Y_i + B_i M2_i + E_i + BA_i + P_i(1 - EE_i) \qquad \text{(Equation 9)}$$

$$B_i\left(\frac{P}{B}\right)_i-\left(\sum_{j=1}^{n}\left(B_j\left(\frac{Q}{B}\right)_j\right)DC_{j,i}\right)-B_i\left(\frac{P}{B}\right)_i P_i(1-EE_i)-Y_i-E_i-BA_i=0$$

(Equation 10)

$$B_i\left(\frac{P}{B}\right)_i EE_i-\left(\sum_{j=1}^{n}\left(B_j\left(\frac{Q}{B}\right)_j\right)DC_{j,i}\right)-Y_i-E_i-BA_i=0$$

(Equation 11)

$$\begin{aligned}
&B_1\left(\frac{P}{B}\right)_1 EE_1-B_1\left(\frac{Q}{B}\right)_1 DC_{1,1}-B_2\left(\frac{Q}{B}\right)_2 DC_{2,1}-\cdots-B_n\left(\frac{Q}{B}\right)_n DC_{n,1}-Y_1-E_1-BA_1=0\\
&B_2\left(\frac{P}{B}\right)_2 EE_2-B_1\left(\frac{Q}{B}\right)_1 DC_{1,2}-B_2\left(\frac{Q}{B}\right)_2 DC_{2,2}-\cdots-B_n\left(\frac{Q}{B}\right)_n DC_{n,2}-Y_2-E_2-BA_2=0\\
&\vdots\\
&B_n\left(\frac{P}{B}\right)_n EE_n-B_1\left(\frac{Q}{B}\right)_1 DC_{1,n}-B_2\left(\frac{Q}{B}\right)_2 DC_{2,n}-\cdots-B_n\left(\frac{Q}{B}\right)_n DC_{n,n}-Y_n-E_n-BA_n=0
\end{aligned}$$

(Equation 12)

$$\text{Consumption}=\text{production}+\text{respiration}+\text{unassimilated food}$$

(Equation 13)

As stated previously, at least three of biomass (B), production/biomass ratio (P/B), consumption/biomass ratio (Q/B) and ecotrophic efficiency (EE) must be given as the basic input. Additionally; diet compositions as well as immigration and emigration rates must be given.

Fooweb models are usually presented as diagrams where the predators are put on the upper trophic levels of their preys. A line connecting two state variables means that the one in the lower trophic level is a food source for the one in the upper level. Figure 10 is an example food web model, developed for a coastal lagoon at the Baltic Sea using Ecopath. Food web models are good tools for simulating organism in the upper levels of the trophic network, however they sometimes lack the components to simulate the nutrients and phytoplankton as accurately as the biogeochemical models. A new emerging approach is to link (one way from the biogeochemical model to the foodweb model) or couple (two ways that both models send feedback to each other) them together. Figure 9 illustrates how the NPZD model described by Figures 2 to 7 could be linked with the food web model described in Figure 8. Figure 9 illustrates the linkage on state variable level. However, according to Equations 10-13 the foodweb model needs more information such as Production over Biomass and diet composition for the linked state variables of the NPZD model. This kind of information can only be extracted from the process rates internally calculated by the NPZD model.

Linking/Coupling of Ecological Models with Transport Models

As stated previously, some ecosystems are either too large in lateral dimensions or too deep so that they should not be considered as completely mixed. If this is the case, then the system must be spatially discretized into different compartments. There are several methods for spatial discretization; such as the finite difference method, finite element method or box modelling approach. In any case the biogeochemical and/or foodweb sub-models equations should be solved for each spatial compartment and the exchanges of material between these compartments should be considered. For this purpose, advection dispersion equation is extended with a reaction term, which includes the ecological sub-models (Figure 10).

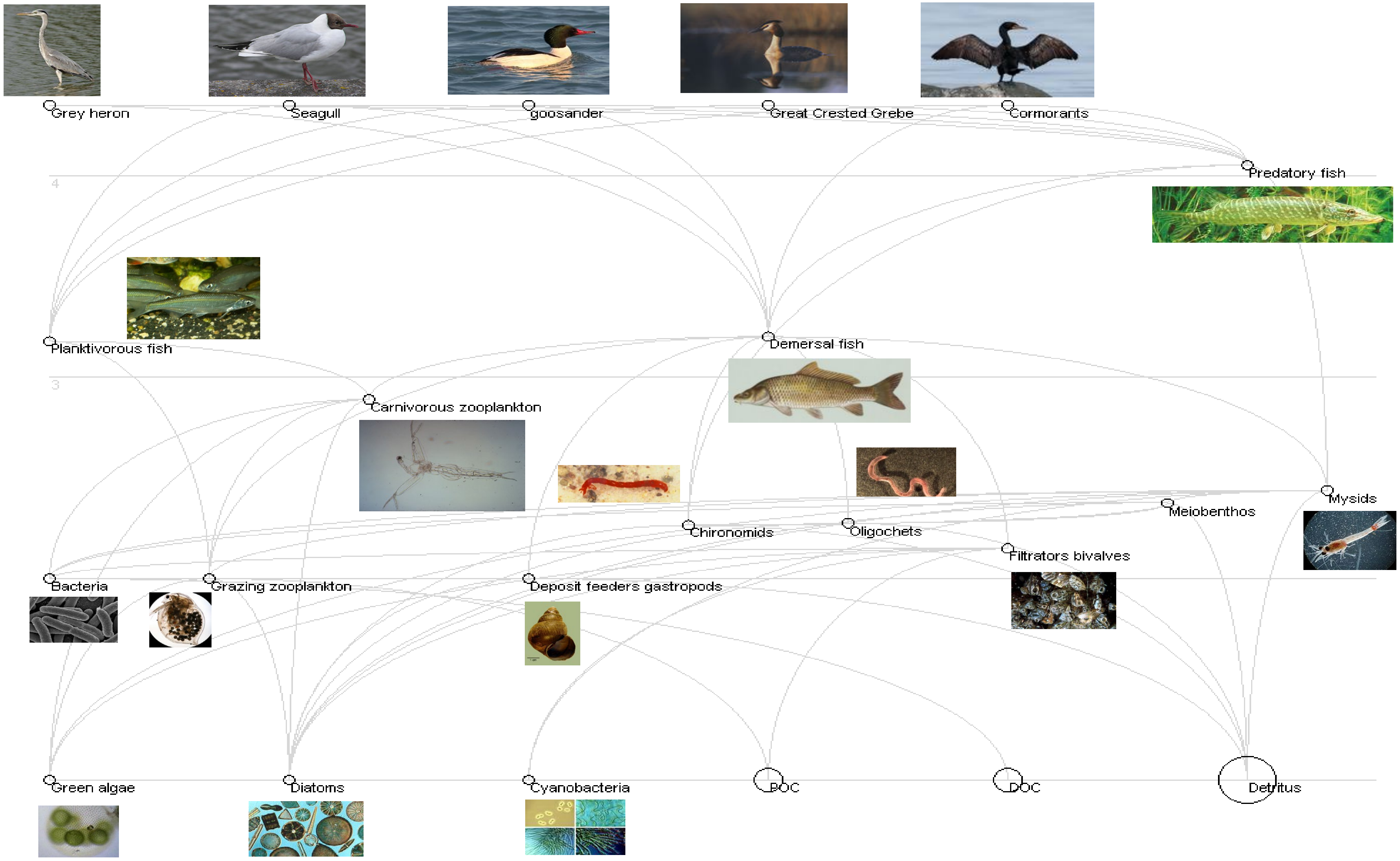


Figure 8. An example foodweb model

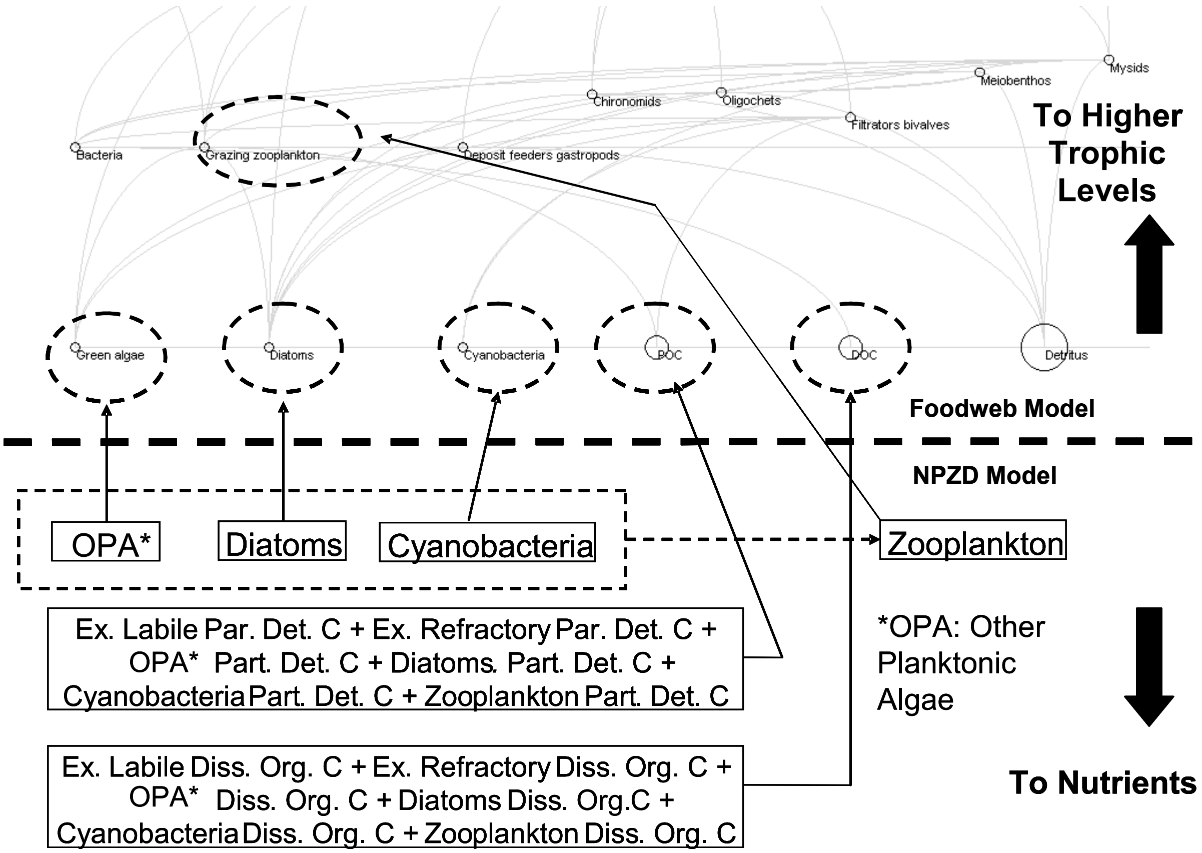
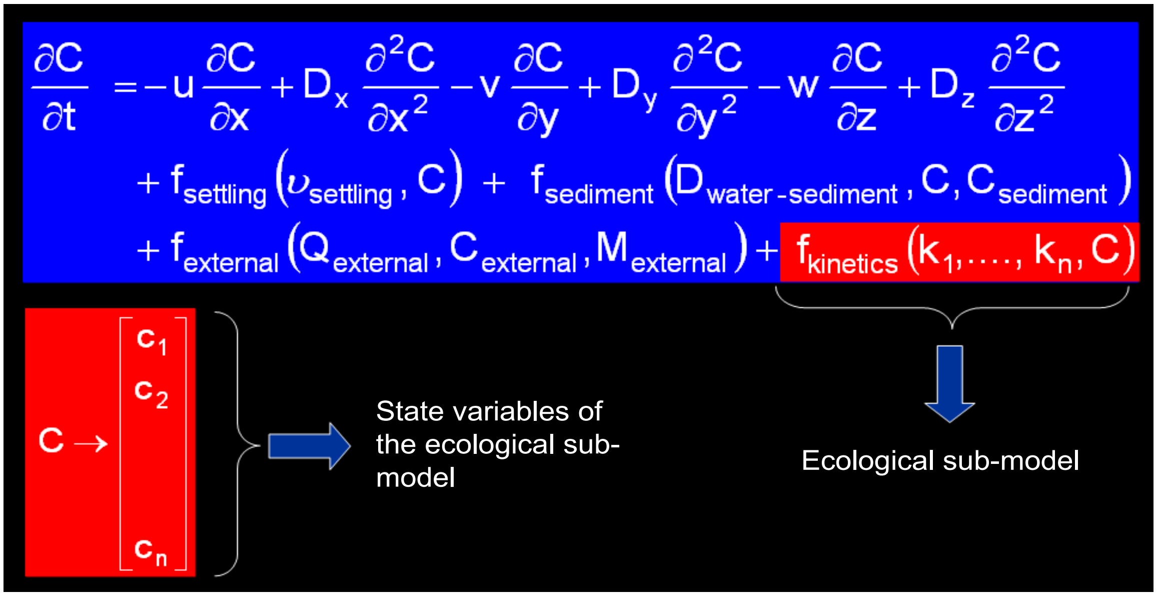

Mysids
Meiobenthos
Chironomids
Oligochets
Filtrators bivalves
Bacteria
Grazing zooplankton
Deposit feeders gastropods
To Higher Trophic Levels
Green algae
Diatoms
Cyanobacteria
POC
DOC
Detritus
Foodweb Model
NPZD Model
OPA*
Diatoms
Cyanobacteria
Zooplankton
Ex. Labile Par. Det. C + Ex. Refractory Par. Det. C + OPA* Part. Det. C + Diatoms. Part. Det. C + Cyanobacteria Part. Det. C + Zooplankton Part. Det. C
*OPA: Other Planktonic Algae
Ex. Labile Diss. Org. C + Ex. Refractory Diss. Org. C + OPA* Diss. Org. C + Diatoms Diss. Org.C + Cyanobacteria Diss. Org. C + Zooplankton Diss. Org. C
To Nutrients

Figure 9. Linking an NPZD model with a foodweb model



$$\frac{\partial C}{\partial t} = -u\frac{\partial C}{\partial x} + D_x\frac{\partial^2 C}{\partial x^2} - v\frac{\partial C}{\partial y} + D_y\frac{\partial^2 C}{\partial y^2} - w\frac{\partial C}{\partial z} + D_z\frac{\partial^2 C}{\partial z^2} + f_{settling}\left(\upsilon_{settling}, C\right) + f_{sediment}\left(D_{water-sediment}, C, C_{sediment}\right) + f_{external}\left(Q_{external}, C_{external}, M_{external}\right) + f_{kinetics}\left(k_1, \ldots, k_n, C\right)$$

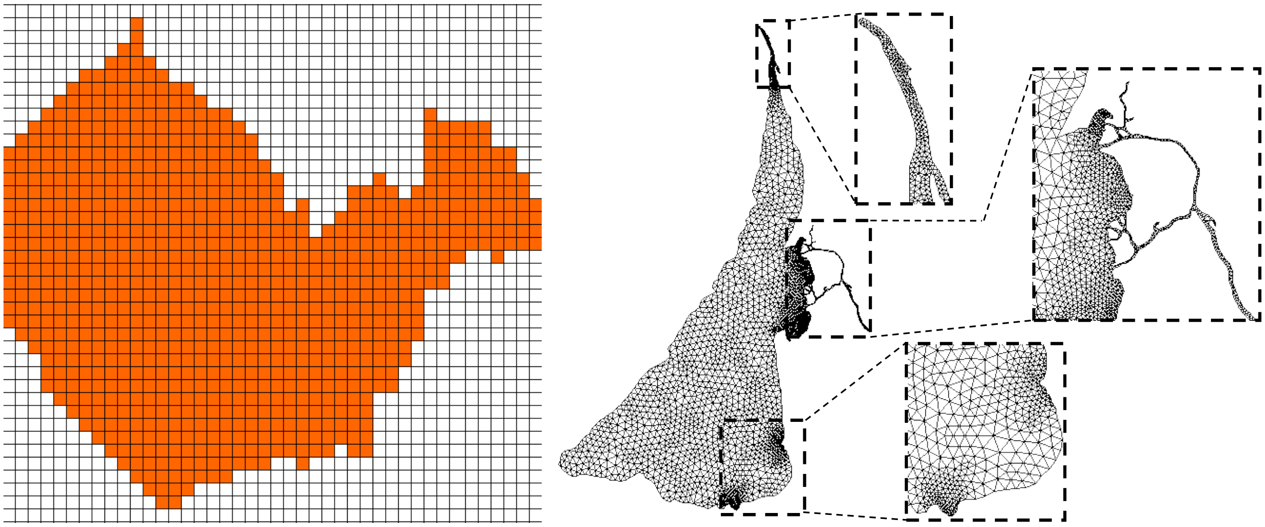
$$C \rightarrow \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}$$

State variables of the ecological sub-model

Ecological sub-model

Figure 10. Advection dispersion equation extended with ecological sub-model

This equation can be solved using different spatial discretization schemes such as the finite differences (Figure 11a), finite elements (Figure 11b) and box discretization (Figure 12). Finite elements are more difficult to handle mathematically than the finite differences, but provide the advantage of spatially variable resolution of discretization. A third commonly used spatial discretization method is the box modelling approach that is similar to finite differences. It is unstructured grid so that the exchanges between the model boxes have to be defined one by one. The advantage is that the boxes can be organized in one, two or three dimensional model domains easily and with a small number of computational elements. The advection diffusion equation extended with ecological sub-model can be rewritten as Equation 14 for a box model.



(a) (b)

Figure 11. Finite differences (a) and finite elements (b)

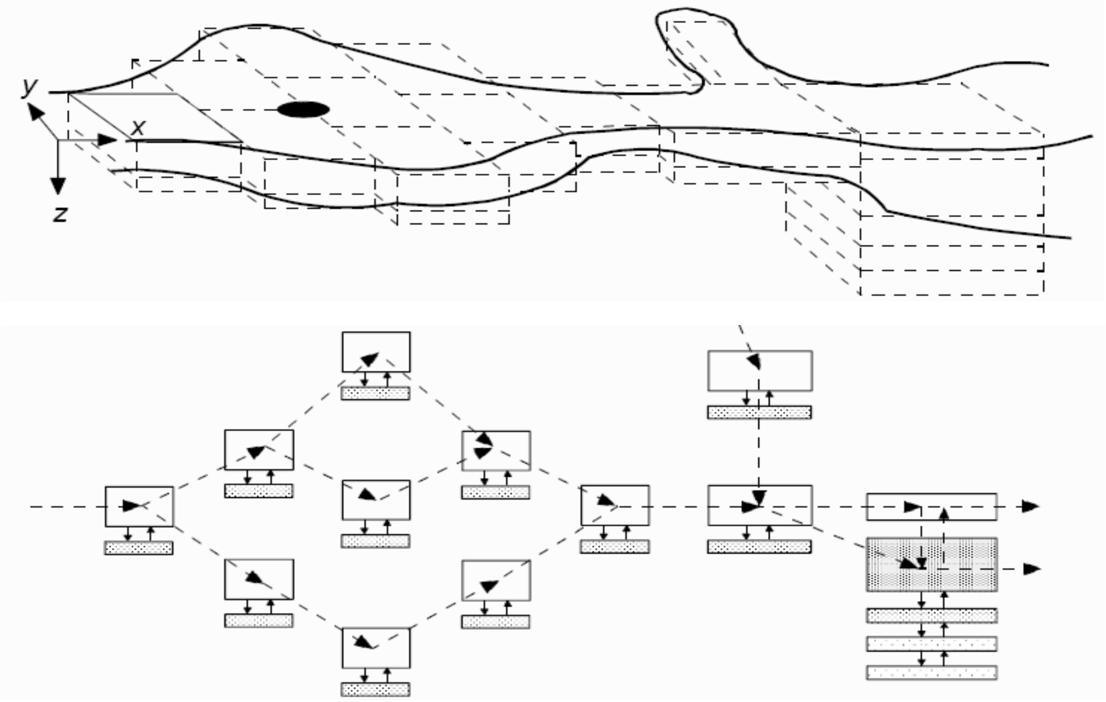


Figure 12. Discretization by box modelling approach

$$\frac{d}{dt}C_1^1 = \sum_{j=1}^{\substack{\text{number of}\\ \text{inflows for}\\ \text{box 1}}} \frac{Q_{j,1}}{V_1}\cdot C_j^1 - \sum_{j=1}^{\substack{\text{number of}\\ \text{outflows for}\\ \text{box 1}}} \frac{Q_{1,j}}{V_1}\cdot C_j^1 + \sum_{j=1}^{\substack{\text{number of}\\ \text{dispersive}\\ \text{exchanges}\\ \text{for box 1}}} \frac{A_{1,j}\cdot D_{1,j}}{\ell_{1,j}\cdot V_1}\cdot\left(C_j^1 - C_1^1\right) + \sum_{m=1}^{\substack{\text{number of}\\ \text{sources and}\\ \text{sinks for box 1}\\ \text{state variable 1}}} S_{1,m}^1 + \sum_{k=1}^{\substack{\text{number of}\\ \text{kinetic reaction}\\ \text{processes for box 1}\\ \text{state variable 1}}} R_{1,k}^1$$

$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots$$

$$\frac{d}{dt}C_i^1 = \sum_{j=1}^{\substack{\text{number of}\\ \text{inflows for}\\ \text{box i}}} \frac{Q_{j,i}}{V_i}\cdot C_j^1 - \sum_{j=1}^{\substack{\text{number of}\\ \text{outflows for}\\ \text{box i}}} \frac{Q_{i,j}}{V_{i1}}\cdot C_j^1 + \sum_{j=1}^{\substack{\text{number of}\\ \text{dispersive}\\ \text{exchanges}\\ \text{for box i}}} \frac{A_{i,j}\cdot D_{i,j}}{\ell_{i,j}\cdot V_i}\cdot\left(C_j^1 - C_i^1\right) + \sum_{m=1}^{\substack{\text{number of}\\ \text{sources and}\\ \text{sinks for box i}\\ \text{state variable 1}}} S_{i,m}^1 + \sum_{k=1}^{\substack{\text{number of}\\ \text{kinetic reaction}\\ \text{processes for box i}\\ \text{state variable 1}}} R_{i,k}^1$$

$$\frac{d}{dt}C_i^2 = \sum_{j=1}^{\substack{\text{number of}\\ \text{inflows for}\\ \text{box i}}} \frac{Q_{j,i}}{V_i}\cdot C_j^2 - \sum_{j=1}^{\substack{\text{number of}\\ \text{outflows for}\\ \text{box i}}} \frac{Q_{i,j}}{V_{i1}}\cdot C_j^2 + \sum_{j=1}^{\substack{\text{number of}\\ \text{dispersive}\\ \text{exchanges}\\ \text{for box i}}} \frac{A_{i,j}\cdot D_{i,j}}{\ell_{i,j}\cdot V_i}\cdot\left(C_j^2 - C_i^2\right) + \sum_{m=1}^{\substack{\text{number of}\\ \text{sources and}\\ \text{sinks for box i}\\ \text{state variable 2}}} S_{i,m}^2 + \sum_{k=1}^{\substack{\text{number of}\\ \text{kinetic reaction}\\ \text{processes for box i}\\ \text{state variable 2}}} R_{i,k}^2$$

$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots$$

$$\frac{d}{dt}C_i^{ns} = \sum_{j=1}^{\substack{\text{number of}\\ \text{inflows for}\\ \text{box i}}} \frac{Q_{j,i}}{V_i}\cdot C_j^{ns} - \sum_{j=1}^{\substack{\text{number of}\\ \text{outflows for}\\ \text{box i}}} \frac{Q_{i,j}}{V_{i1}}\cdot C_j^{ns} + \sum_{j=1}^{\substack{\text{number of}\\ \text{dispersive}\\ \text{exchanges}\\ \text{for box i}}} \frac{A_{i,j}\cdot D_{i,j}}{\ell_{i,j}\cdot V_i}\cdot\left(C_j^{ns} - C_i^{ns}\right) + \sum_{m=1}^{\substack{\text{number of}\\ \text{sources and}\\ \text{sinks for box i}\\ \text{state variable ns}}} S_{i,m}^{ns} + \sum_{k=1}^{\substack{\text{number of}\\ \text{kinetic reaction}\\ \text{processes for box i}\\ \text{state variable ns}}} R_{i,k}^{ns}$$

$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots$$

$$\frac{d}{dt}C_{nb}^{ns} = \sum_{j=1}^{\substack{\text{number of}\\ \text{inflows for}\\ \text{box nb}}} \frac{Q_{j,nb}}{V_{nb}}\cdot C_j^{ns} - \sum_{j=1}^{\substack{\text{number of}\\ \text{outflows for}\\ \text{box nb}}} \frac{Q_{nb,j}}{V_{nb}}\cdot C_j^{ns} + \sum_{j=1}^{\substack{\text{number of}\\ \text{dispersive}\\ \text{exchanges}\\ \text{for box nb}}} \frac{A_{nb,j}\cdot D_{i,j}}{\ell_{nb,j}\cdot V_{nb}}\cdot\left(C_j^{ns} - C_{nb}^{ns}\right) + \sum_{m=1}^{\substack{\text{number of}\\ \text{sources and}\\ \text{sinks for box nb}\\ \text{state variable ns}}} S_{nb,m}^{ns} + \sum_{k=1}^{\substack{\text{number of}\\ \text{kinetic reaction}\\ \text{processes for box nb}\\ \text{state variable ns}}} R_{nb,k}^{ns}$$

(Equation 14)

In Equation 14; nb is the number of boxes, ns is the number of state variables, index i corresponds to the actual box, index j corresponds to any neighbouring box, $Q_{i,}$ is the flow rate between boxes i and j [$L^3 \cdot T^{-1}$], $D_{i,j}$ is the dispersion coefficient between boxes i and j [$L^2 \cdot T^{-1}$], $\ell_{i,j}$ is the mixing length between boxes i and j [L], $A_{i,j}$ the interface area between boxes i and j [L^2], V_i the volume of box i [L^2], $S_{i,m}^s$ the external source m related to state variable s for box i [$M \cdot L^{-3} \cdot T^{-1}$] and the $R_{i,k}^s$ is Kinetic reaction rate k for state variable s in for box i [$M \cdot L^{-3} \cdot T^{-1}$]. The water exchanges between boxes can be calculated using a hydrodynamic model such as the one given in Appendix B.

A Case Study

The model described in "Development of Biogeochemical Cycle Sub-Models for Eutrophication Analyses" Section was applied to Curonian Lagoon (Figure 13), which is a shallow estuarine lagoon located in Lithuania at the south-eastern coast of the Baltic Sea. Curonian lagoon is a eutrophic estuarine lagoon downstream the Nemunas River. During cyanobacterial blooms, chlorophyll-a concentrations exceeding 200 $mg \cdot m^{-3}$ were measured on monitoring studies. Peak total organic carbon concentrations exceeding 30 $g \cdot m^{-3}$ are common.

The lagoon was previously modelled by Erturk (2008) and Erturk et al (2015) using the NPZD model described in "Development of biogeochemical cycle sub-models for eutrophication analyses" Sub-section incorporated into Equation 14. The model then was successfully linked to a foodweb model as illustrated in "Development of foodweb sub-models" Sub-section and used for nutrient management scenarios in Nemunas River Basin. The water exchanges between boxes are calculated using the finite element hydrodynamic model SHYFEM. The model setup and linkage is illustrated in Figure 14.

The model was used to simulate the effects of possible warming of the Curonian Lagoon due to climate change. The scenarios here are fictive just to test the behaviour of the model at increased lagoon water temperature. Forcing factors except the temperatures were not changed. The spatially and temporally (yearly) averaged results are summarized in Figure 15 and Figure 16.

As seen from the figures, the total phytoplankton biomass increases first with temperature, but then decreases. This is because of the temperature stress effects considered by the model where the death rate constant is increasing with the temperature. Dead organic carbon is increasing with increasing temperature indicating that the total primary production is increasing, however with decreased net primary production so that dead organic matter is accumulating in the system even though the total phytoplankton concentration is decreasing after an increase of 4ºC in water temperature. Figure 15b illustrates the response of production over biomass ratio to the increase in temperature. Basically, diatoms that prefer coder water are not affected by temperature increase since they dominate the phytoplankton community on the colder seasons and do not peak in warmer seasons. Therefore, their yearly average biomass does not change considerably. Consequently, the main competition is between the cyanobacteria and the greens. As seen in figure 15b, production over biomass ratio is increasing by cyanobacteria and decreasing by other planktonic algae. Since cyanobacteria are less available as food source, the ecotrophic efficiency of the Curonian Lagoon can be expected to decrease if the temperature increases, because there would be less of available phytoplankton biomass to upper levels of the food web. This effect is reproduced by the model as well by the continually decrease of zooplankton when the temperature increases (Figure 16).

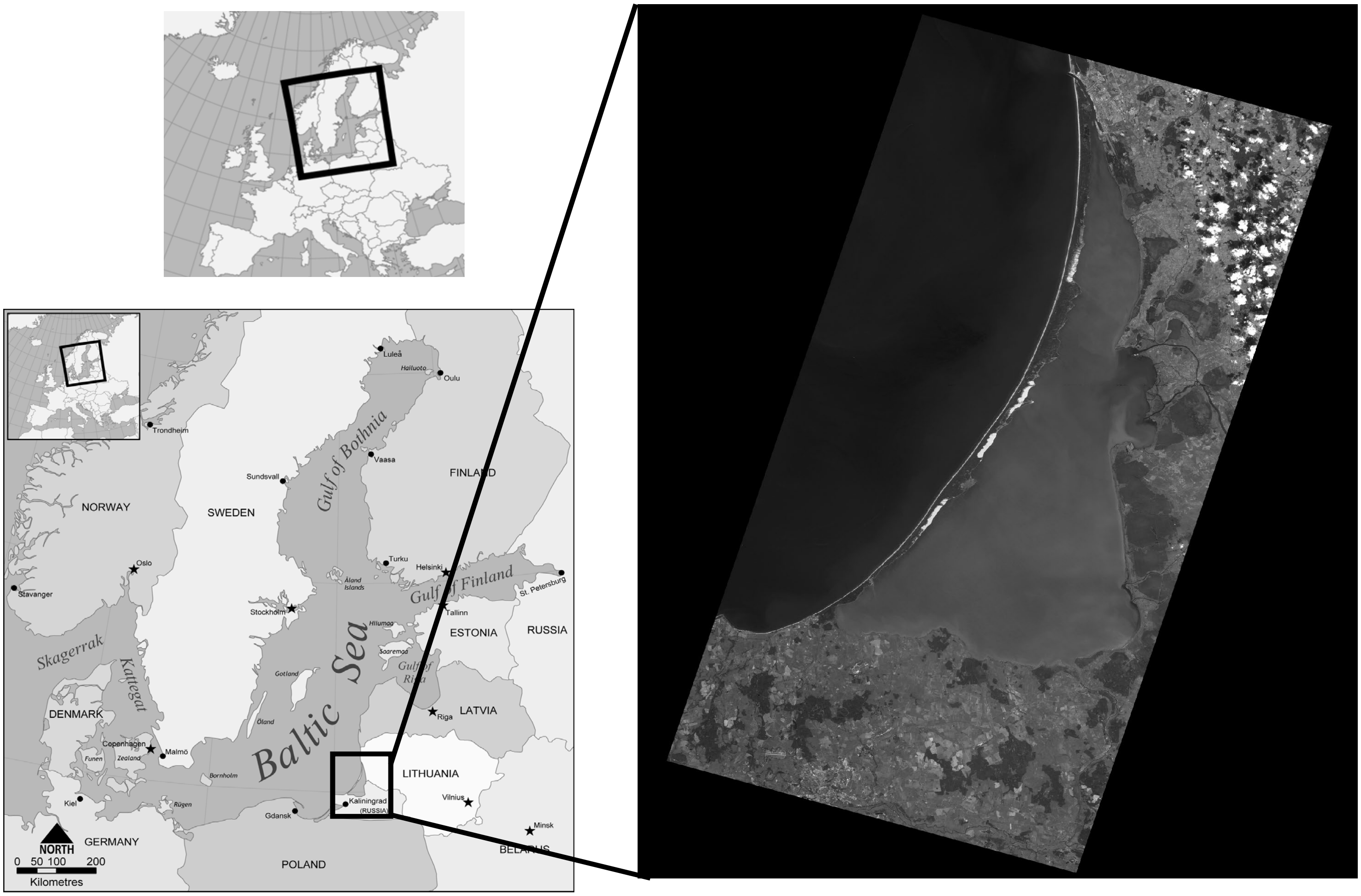
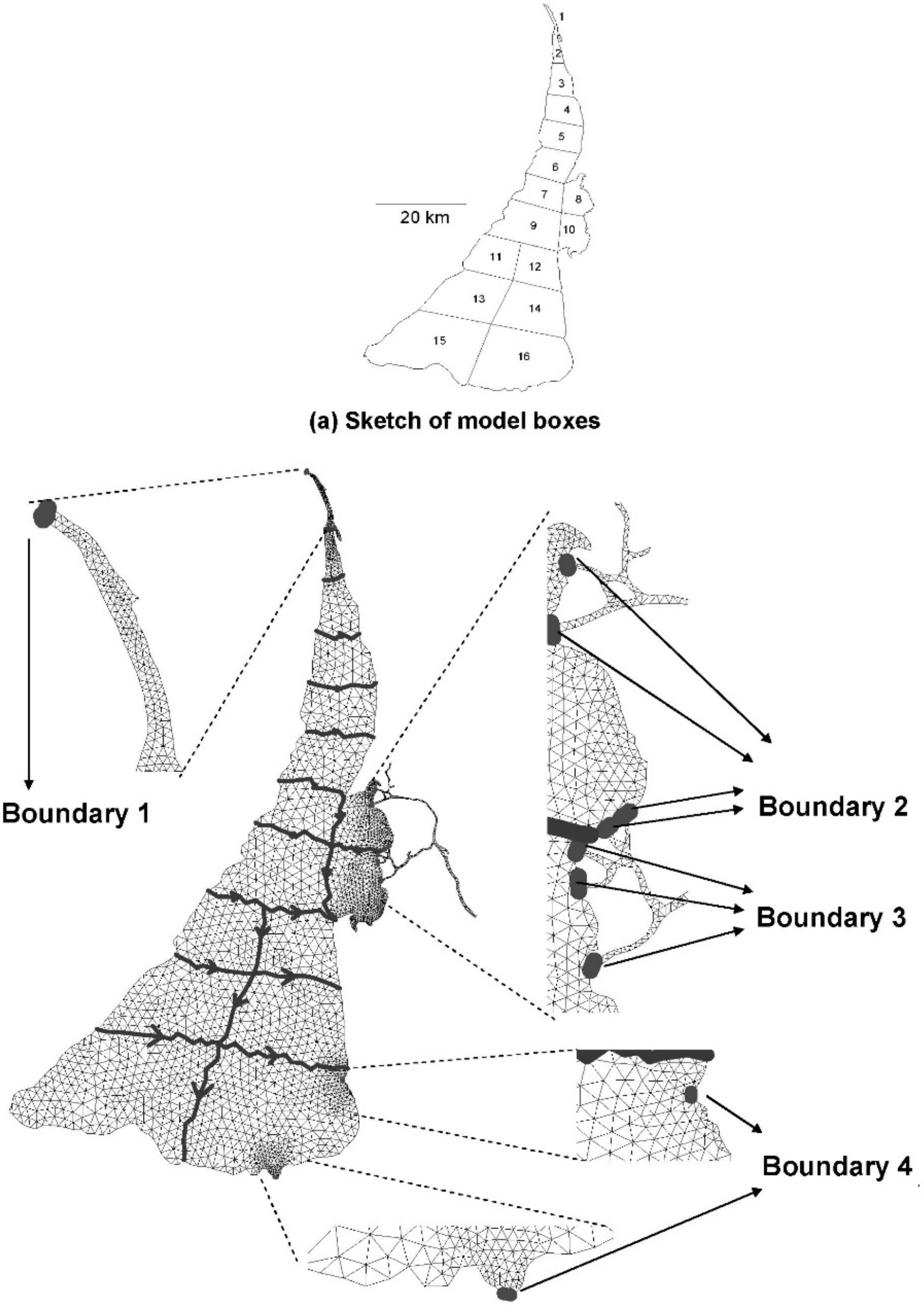


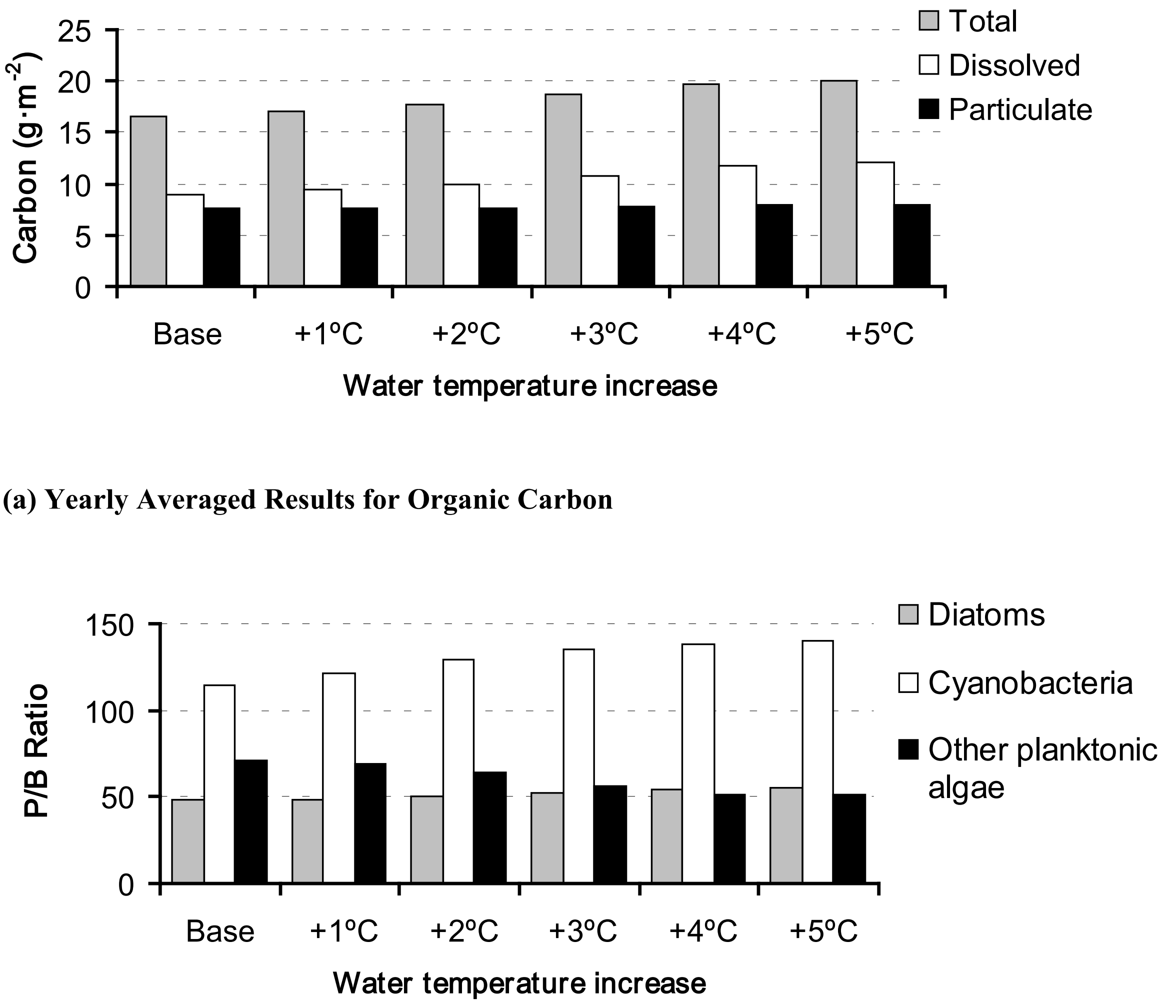
Figure 13. The Curonian Lagoon



(a) Sketch of model boxes

(b) Linking model boxes with hydrodynamic model elements and boundaries

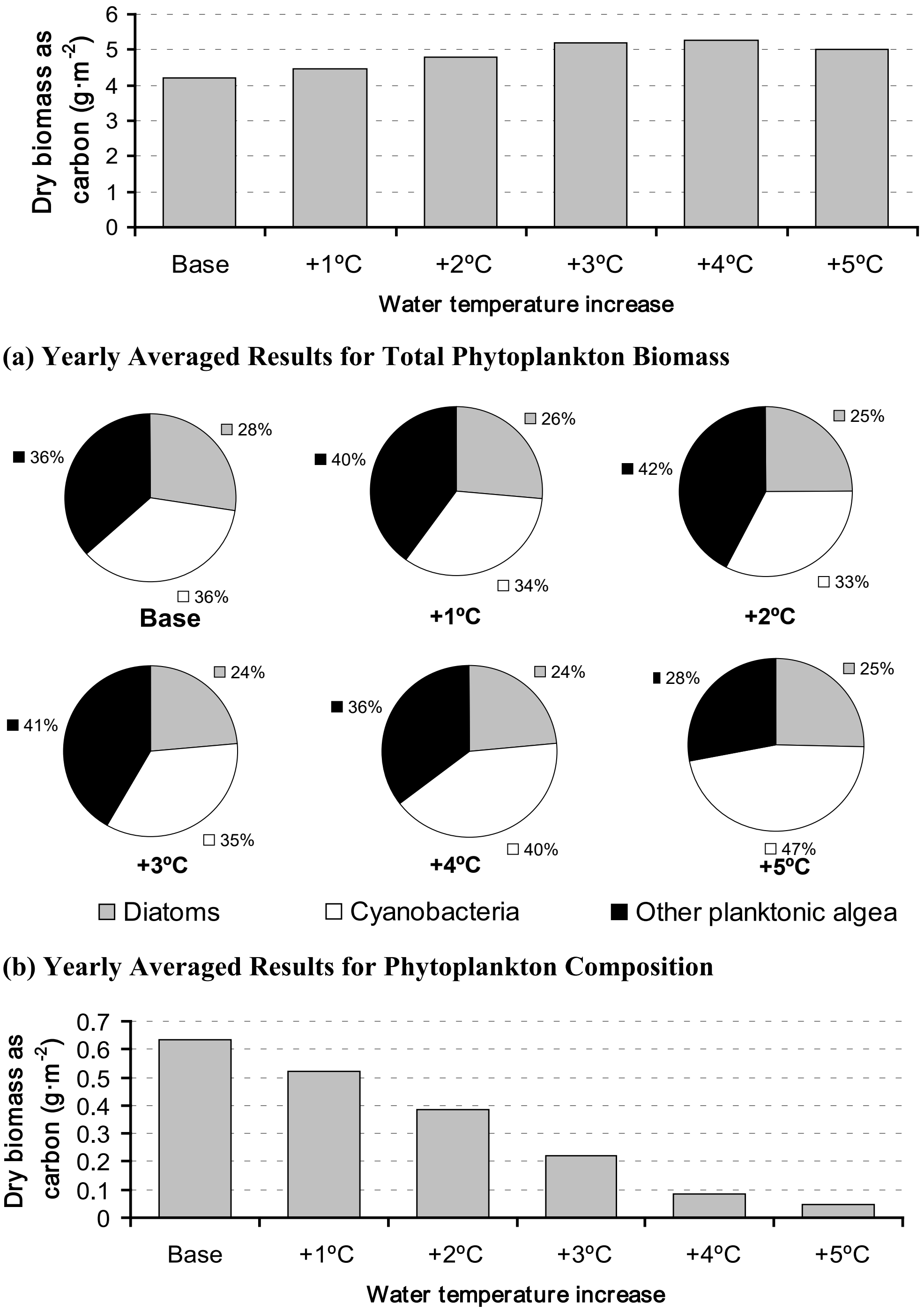
Figure 14. Model setup (Erturk, 2008; Erturk et al., 2015)



(a) Yearly Averaged Results for Organic Carbon

(b) Yearly Production over Biomass Results for Organic Carbon

Figure 15. Simulation results related to organic matter and primary production (Erturk et al., 2015)



(a) Yearly Averaged Results for Total Phytoplankton Biomass

(b) Yearly Averaged Results for Phytoplankton Composition

(c) Yearly Averaged Results for Total Zooplankton Biomass

Figure 16. Simulation results related to phytoplankton and zooplankton (Erturk et al., 2015)

Conclusions

Eutrophication is a complicated process and its predictive modelling may involve many tools applied in an interdisciplinary manner. Such a modelling effort could seem overwhelming for many researchers new to the topic. This paper however shows that building such models even from scratch is really not "rocket science" and most of the aquatic scientists already have the necessary mathematical background.

Once a simple model such as the one illustrated in Appendix A, it is quite easy to extend it into more comprehensive frameworks, such as a combined ecological model linked to higher trophic compartments as described in "Ecological Sub-Models for Prediction of the Progress and Effects of Eutrophication" Section.

Mathematical models are not only useful to predict the progress of eutrophication but they are also valuable tools for system identification. The model presented in "Development of Foodweb Sub-Models" Section is such an example, where in internals such as ecotrophic efficiency of the foodweb is estimated rather than the biomasses of individual trophic compartments.

Compliance with Ethical Standard

Conflict of interests: The authors declare that for this article they have no actual, potential or perceived conflict of interests.

References

Ambrose, B, Jr., Wool, T.A. Martin, J.L. (1993). *The Water Quality Analysis Simulation Program, WASP5; Part A: Model Documentation*, U.S. Environmental Protection Agency, Center for Exposure Assessment Modeling, Athens, GA.,

Arhonditsis, G. B., Brett, M. T. (2004). Evaluation of the current state of mechanistic aquatic biogeochemical modelling. *Marine Ecology Progress Series,* 271, 13-26.

Aydin, K.Y., McFarlane, G.A., King, J.R., Megrey, B.A. (2003). *The BASS/MODEL Report on Trophic Models of the Subarctic Pacific Basin Ecosystems*, PICES Scientific Report No. 25, https://www.pices.int/publications/scientific_reports/Report25/default.aspx, (accessed 8.4.2019).

Bartsch, A.F., Gakstatter, J.H. (1978). *Management Decision for Lake Systems on a Survey of Trophic Status, Limiting Nutrients, and Nutrient Loadings*, American-Soviet Symposium on Use of Mathematical Models to Optimize Water Quality Management, 1975, U.S. Environmental Protection Agency Office of Research and Development, Environmental Research Laboratory, Gulf Breeze, FL, pp 372-394, EPA-600/9-78-024

Bloesch, J, Stadelmann, P., Bührer, H. (1977). Primary production, mineralization and sedimentation in the euphotic zone of two swiss lakes. *Limnology and Oceanography*, 22(3), 511-526.

Cerco, C.F., Cole, T. (1994). *Three Dimensional Eutrophication Model of Chesapeake Bay; Volume 1, Main Report*, Technical Report EL 94-4, U.S. Army Corps of Engineers Waterways Experiment Station, Vicksburg, MS.

Cerco, C.F., Cole, T. (1995). *User's Guide to the CE-QUAL-ICM Three Dimensional Eutrophicaiton Model, Release Version 1.0.* Technical Report EL-95-15, US Army Corps of Engineers Waterways Experiment Station, Vicksburg, MS.

Chapra, S.C. (1997). *Surface Water-Quality Modeling*, WCB McGraw-Hill Publisher, ISBN 0-07-024186-4

Christensen, V., Walters, C.J. (2004). Ecopath with ecosim: methods capabilities and limitations. *Ecological Modelling*, 172, 109-139.

Christensen, V., Walters, C.J., Pauly, D. (2005). *Ecopath with Ecosim: A User's Guide*, Fisheries Centre University of British Columbia Vancouver, Canada.

Cole, T.M., Wells, S.A. (2006). *CE-QUAL-W2 A Two dimensional, Laterally Averaged Hydrodynamic and Water Quality Model, Version 3.5*, Instruction Report EL-2006-1, U.S. Army Engineering and Research Development Center, Vicksburg, MS.

Dillon, P.J., Rigler, F.H. (1974). The phosphorus-chlorophyll relationship for lakes. *Limnology and Oceanograpgy*, 19, 767-773.

Di Toro, D.M., O'Connor, D.L., Thomann, R.V. (1971). *A Dynamic Model of the Phytoplankton Population in the Sacramento-San Joaquin Delta*. Advances in Chemistry Series 106, Nonequilibrium Systems in Natural Water Chemistry, 131.

Di Toro, D.M., Connolly, J.P. (1980). *Mathematical Models of Large Lakes, Part 2. Ontario Lake*. National Environmental Research Center, Office of Research and Development, U.S. Environmental Protection Agency, Ecological Research Series, EPA-600/3-80-065.

Di Toro, D.M., Fitzpatrick, J.J., Thomann, R.V. (1983). *Water Quality Analysis Simulation Program (WASP) and Model Verification Program (MVP) – Documentation*. Contract No 68-01-3872, Hydroscience, Inc., USA.

Di Toro, D.M., Fitzpatrick, J.J. (1993). *Chesapeake Bay Sediment Flux Model*. Contract Report EL-93-2, Environmental Laboratory U.S. Army Engineer Waterways Experiment Station.

Edmonson, W.T. (1979). Phosphorus, nitrogen and algae in lake washington after diversion of sewage. *Science*, 169, 690-691.

Environmental Laboratory, (1995). CE-QUAL-R1: *A Numerical One Dimensional Model of Reservoir Water Quality; User's Manual*. Instruction Report E-82-1, Rev. Ed., US Army Engineer Waterways Experiment Station, Vicksburg MS.

Erturk, A. (2008). *Modelling the Response of an Estuarine Lagoon to External Nutrient Inputs*. Dissertation, Klaipeda University.

Erturk, A., Razinkovas-Baziukas, A., Zemlys, P., Umgiesser, G. (2015). Linking carbon-nitrogen-phosphorus cycle and foodweb models of an estuarine lagoon ecosystem. *Computational Science and Tecniques*, 3(1), 350-412.

Fayram, A.H. (2005). *Walleye Stocking in Wisconsin Lakes: Species Interactions, Changes in Angler Effort, Optimal Stocking Rates and Effects on Community Maturity,* Dissertation in Biological Sciences at the University of Wisconsin, Milwaukee, USA.

Gamito, S., Erzini, K. (2005). Trophic food web and ecosystem attributes of a water reservoir of the Ria Formosa (south portugal), *Ecological Modelling*, 181, 509-520.

Hamrick, J.M. (1996). *User's Manual for the Environmental Fluid Dynamics Computer Code*, Special Report No. 331 in Applied Marine Science and Ocean Engineering Virginia Institute of Marine Science School of Marine Science, The College of William and Mary Gloucester Point, VA 23062.

Harvey, C.J., Cox, S.P., Essington, T.E., Hansson, S, Kitchell, J.F. (2003). An ecosystem model of food web and fisheries interactions in the Baltic Sea. *ICES Journal of Marine Science*, 60, 939-950.

HEC (1978). *Generalized Computer Program, Water Quality for River-Reservoir Systems*, The Hydrologic Engineering Center, United States Army Corps of Engineers.

Hossenipour, E.Z., Martin J.L. (1990). T*he One-Dimensional Riverline Hydrodynamic Model*, RIVMOD-H Model Documentation and User's Manual, Environmental Laboratory Office of Research and Development USEPA Athens, Georgia 30605-2700.

Hull, V., Mocenni, C., Falcucci, M., Marchettini, N. (2000). A trophodynamic model for the lagoon of Fogliano (Italy) with ecological dependent modifying parameters. *Ecological Modelling*, 134, 153-167.

Kavanagah, P., Newlands, N., Christensen, V., Pauly, D. (2004). Automated parameter optimization for ecopath ecosystem models. *Ecological Modelling*, 172, 141-149.

Luyten, P.J., Jones, J.H., Proctor, R., Tabor, A., Tett, P., Wild-Allen, K. (1999). *COHERENS – A Coupled Hydrodynamical – Ecological Model for Regional and Shelf Seas: User Documentation*. MUMM Report, Management Unit of the Mathematical Models of the North Sea.

Megrey, B.A., Taft, B.A., Peterson, W.T. (Eds.) (2001). *PICES-GLOBEC International Program on Climate Change and Carrying Capacity*. Report of the 2000 BASS, MODEL, MONITOR and REX Workshops, and the 2001 BASS/MODEL Workshop. PICES Sci. Rep. No. 17.

Mohamed, K.S., Zacharia, P.U., Muthiah, C., Abdurahiman, K.P., Nayak, T.H. (2005). *A Trophic Model of the Arabian Sea Ecosystem off Karnataka and Simulation of Fishery Yields for its Multigear Marine Fisheries*. Research Centre of Central Marine Fisheries Research Institute, Karnataka, India.

O'Connor D. J., John, P. St., Di Toro, M. (1968). Water quality analyses of the Delaware River Estuary. *Journal of the Sanitary Engineering Division,* 94(SA6), 1225-1252.

Okey, T.A., Pauly, D.A. (1999). *Mass-Balanced Model of Trophic Flows in Prince William Sound: Decompartmentalizing Ecosystem Knowledge*. Ecosystem for Fisheries Management, Alaska Sea Grant College Program, AL-SG-99-01, pp 621-635.

Opiz, S. (1996). *Trophic Interactions in Caribbean Coral Reefs*, International Center for Living Aquatic Resources, Makati City, Philippines.

Pauly, D. (1998). *Use Ecopath with Ecosim to Evaluate Strategies for Sustainable Exploitation of Multi-Species Resources*: Proceedings of a Workshop held at the Fisheries Centre of University of British Columbia, Vancouver, B.C., Canada, edited by Pauly, D., Fisheries Centre Research Reports, Volume 6(2), ISSN 1198-6727

Pauly, D., Christensen, V., Walters, C. (2000). Ecopath ecosim and ecospace as tools for evaluating ecosystem impacts of fisheries. *ICES Journal of Marine Science*, 57, 1-10.

Polovina, J.J., Ow, M.D. (1983). *ECOPATH: A user's manual and program listings*, Administrative report, H-83-23, Sothwest Fisheries Center, Honolulu Laboratory, Honolulu, Hawaii, USA.

Rast, W., Lee, G.F. (1978). *Summary Analyses of the North American Project (US Portion) OECD Eutrophication Project: Nutrient Loading-Lake Response Relationships and Trophic State Indices*, USEPA Corvallis Environmental Research Laboratory, Corvallis, OR, EPA-600/3-78-008.

Royal Comission on Environmental Pollution, (2004). *Turning the Tide: Addressing the Impact of Fisheries on the Marine Environment*, 25th Repot, Chairman: Tom Blundell, FMedSci.

Sheng, Y.G. Eliason, D.E., Chen, X.J., Choi, J.K. (1991). A *Three Dimensional Numerical Model of Hydrodynamics and Sediment Transport in Lakes and Estuaries: Theory, Model Development and Documentation*. USEPA Athens, GA, USA.

Smith, V.H., Shapiro, J. (1981). A *Retroperspective Look at the Effects of Phosphorus Removal in Lakes, in Restoration of Lakes and Inland Waters*. USEPA, Office of Water Regulations and Standards, Washington, DC. EPA-440/5-81-010.

Thomann, R.V. Di Toro, D.M., Winfield, R.P. O'Connor, D.O. (1975). *Mathematical Modeling of Phytoplankton in Lake Ontario, 1. Model Development and Verification*, National Environmental Research Center, Office of Research and Development, USEPA, Ecological Research Series, EPA-660/3-75-005.

Thomann, R.V., Mueller, J.A., (1987). *Principles of Surface Water Quality Modeling and Control*, Harper Collins Publishers Inc., USA.

Tillman, D.H., Cerco, C.F., Noel, M.R. (2006). *Conceptual Processes for Linking Eutrophication and Network Models*. ERDC TN-SWWRP-06-9, United States Army Corps of Engineers.

USEPA, (2000). *Estuarine and coastal marine waters: Bioassessment and biocriteria technical guidance*. USEPA Report EPA-822-B00-024, Washington, DC.

Villanueva, M.C. Laleyeb, P., Albaret J.J., Lae, R., de Moraise, L. Tito, Moreaua J. (2006). Comparative analysis of trophic structure and interactions of two tropical lagoons. *Ecological Modelling*, 197, 461-477.

Walters, C., Pauly, D., Christensen, V. (1999). Ecospace: prediction of mesoscale spatial patterns in trophic relationships of exploited ecosystems with emphasis on the impact of marine protected areas. *Ecosystems* (1999)2, 539-554.

Walters, C., Pauly, D., Christensen, V., Kitchell, J.F. (2000). Representing density dependent consequences of life history strategies in aquatic ecosystems: Ecosim II. *Ecosystems* (2000)3, pp 70-83.

Wool, T.A., Ambrose, R.B.Jr., Martin, J.L., Comer, E.A. 2001. *The Water Quality Analysis Simulation Program, WASP* USEPA, Centre for Exposure Assessment Modeling, Athens, GA.

APPENDIX A

Development and Implementation of Simplified Eutrophication Modelling Tools from Scratch

The aim of this section is to illustrate the reader how to develop own modelling tools that can simulate the progress of the eutrophication process on simple but complete examples. Before starting to read this section, be advised that the development of an eutrophication model from scratch is not a simple process and consists of several tasks listed below:

- Development of a conceptual model
- Writing the equations that form the mathematical construct of the model
- Development of solution schemes for the equations
- Implementation of the model as a tool
- Development of the supporting environment and tools for the model

A.1. Development of a conceptual model

Development of a conceptual model is the first and most important step for developing a complete modelling tool. The conceptual model is the first level of modelling and describes the simplified system (actually aquatic ecosystem since our aim is to develop a eutrophication model) as it will be assumed by our model based analysis. The conceptual model "glues" the models state variables (the variables that are calculated by the model to describe the state of the aquatic ecosystems in terms of eutrophication), the auxiliary variables needed by the model itself and the processes (here the ecological processes related to eutrophication within the framework of the model) together. The conceptual model for the example in this subsection is illustrated in Figure A.1 is used as the conceptual model.

The aquatic ecosystem that is assumed to be a lake in this example is considered as a fully mixed reactor. The three state variables are unavailable phosphorus that includes all the dead and organically bound phosphorus, dissolved reactive phosphorus that can be utilized as nutrient and the phytoplankton chlorophyll representing the primary produces. The loads shown in Figure 1 are examples of auxiliary variables. The conceptual model includes the processes listed below:

- Settling of unavailable phosphorus
- Release of unavailable phosphorus by phytoplankton death
- Conversion of unavailable phosphorus to dissolved reactive phosphorus by decomposition
- Uptake of soluble reactive phosphorus by photosynthesis
- Release of unavailable phosphorus by phytoplankton
- Death of phytoplankton
- Settling of phytoplankton
- Inflow and outflow of soluble reactive phosphorus
- Inflow and outflow of unavailable phosphorus
- Inflow and outflow of phytoplankton

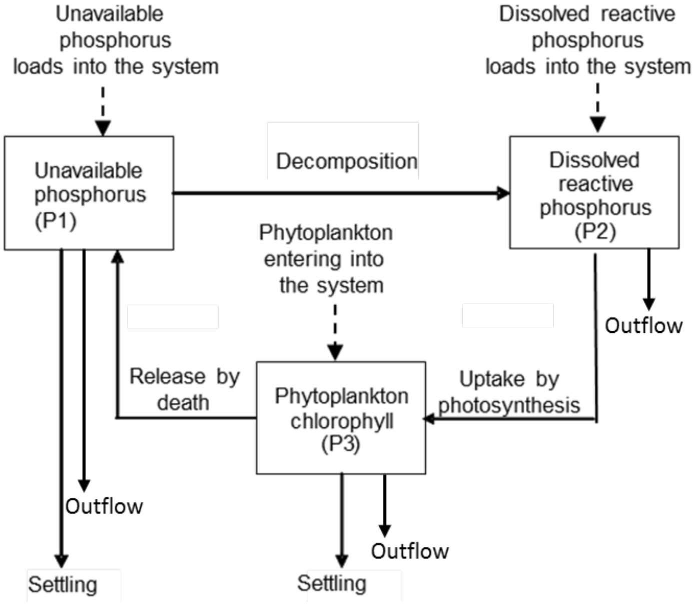

Unavailable phosphorus loads into the system
Dissolved reactive phosphorus loads into the system
Unavailable phosphorus (P1)
Decomposition
Dissolved reactive phosphorus (P2)
Phytoplankton entering into the system
Outflow
Release by death
Phytoplankton chlorophyll (P3)
Uptake by photosynthesis
Outflow
Outflow
Settling
Settling

Figure A.1. A simple, process based nutrient cycle model

A.2. Writing the equations that form the mathematical construct of the model

The next step is the writing the equations that describe the lake ecosystem mathematically. Since the aim of the model in this example is to describe the progress of eutrophication, it must be a dynamic model, where time (t) is the independent variable, whereas the state variables (P1, P2 and P3) are dependent variables. The processes as well as loads force the values of state variables to change. The state variables are in concentration dimensions ([$M \cdot L^{-3}$]- mass over the third power of length or mass of volume), whereas the processes are in reaction rate dimensions ([$M \cdot L^{-3} \cdot T^{-1}$]-change of concentration per unit time). To put the state variables and processes on the same equation, state variables should be rewritten in reaction rate dimensions as well, so mathematically divided by time. To describe the change in a moment (indefinitely small time) differential equations are needed. Since there are three state variables where the rate of change of a state variable depends on itself and on other state variables, we end up with system with three unknowns (the state variable) and three differential equations.

$$\frac{dP1}{dt}=\underbrace{\frac{W_{Chl\text{-}A}}{V}}_{\substack{\text{Incomming}\\ \text{mass of}\\ \text{unavailable}\\ \text{phosphorus}}}-\underbrace{\frac{Q_{OUT}}{V}}_{\substack{\text{Outgoing}\\ \text{mass of}\\ \text{unavailable}\\ \text{phosphorus}}}\cdot P1-\underbrace{\left(K_{12}\cdot\underbrace{\theta_{Decomp}^{\;TEMP-20}}_{\text{Temperature effect}}+\frac{v_{S,P1}}{H}\right)}_{\substack{\text{Settling rate of unavailable}\\ \text{phosphorus plus its conversion}\\ \text{rate to dissolved reavtive phosphorus}\\ \text{by decomposition}}}\cdot P1+\underbrace{D_P\cdot a_P\cdot Chl\text{-}A}_{\substack{\text{Death rate}\\ \text{of phytoplankton}}}\qquad\text{(Equation A.1)}$$

$$\frac{dP2}{dt}=\underbrace{\frac{W_{P2}}{V}}_{\substack{\text{Incomming}\\ \text{mass of}\\ \text{dissolved}\\ \text{reactive}\\ \text{phosphorus}}}-\underbrace{\frac{Q_{OUT}}{V}}_{\substack{\text{Outgoing}\\ \text{mass of}\\ \text{dissolved}\\ \text{reactive}\\ \text{phosphorus}}}\cdot P2+\underbrace{K_{12}\cdot\theta_{Decomp}^{\;TEMP-20}\cdot P1}_{\substack{\text{Conversion rate of}\\ \text{unavailable}\\ \text{phosphorus to dissolved}\\ \text{reactive phosphorus by}\\ \text{decomposition}}}-\underbrace{G_P\cdot a_P\cdot Chl\text{-}A}_{\substack{\text{Uptake rate of dissloved}\\ \text{reactive phosphorus by}\\ \text{phytoplankton}}}\qquad\text{(Equation A.2)}$$

$$\frac{dChl\text{-}A}{dt}=\underbrace{\frac{W_{Chl\text{-}A}}{V}}_{\substack{\text{Incomming}\\ \text{mass of}\\ \text{phytoplankton}}}-\underbrace{\frac{Q_{OUT}}{V}\cdot Chl\text{-}A}_{\substack{\text{Outgoing}\\ \text{mass of}\\ \text{phytoplankton}}}-\underbrace{\frac{v_{S,Phyto}}{H}\cdot Chl\text{-}A}_{\substack{\text{Settling rate of}\\ \text{phytoplankton}}}+\underbrace{\left(G_P-D_P\right)\cdot Chl\text{-}A}_{\substack{\text{Net growth rate}\\ \text{of phytoplankton}}}\qquad\text{(Equation A.3)}$$

where V is the volume of the lake [L^3]; P1, P2, Chl-A are unavailable (organically bound) phosphorus, available phosphorus and chlorophyll-a respectively [$M\cdot L^{-3}$], Q_{OUT} is the outflow [$L^3\cdot T^{-1}$], H is the depth of the lake [L], $v_{S,P1}$ is the settling velocity for unavailable phosphorus [$L\cdot T^{-1}$], $v_{S,Chl\text{-}A}$ is the settling velocity for phytoplankton [$L\cdot T^{-1}$]; W_1, W_2, $W_{Chl\text{-}A}$ are the loads for unavailable phosphorus, available phosphorus and chlorophyll-a respectively[$M\cdot T^{-1}$], K_{12} is the decomposition rate constant [T^{-1}]. G_P and D_P are the growth and death rate coefficients of phytoplankton respectively [T^{-1}]. As seen in below, these coefficients are calculated using three relatively long algebraic equations.

$$G_P=G_{MAX}\cdot\underbrace{\theta_{Phyto,G}^{\;TEMP-20}}_{\substack{\text{Limiting effect of}\\ \text{temperature on}\\ \text{phytoplankton}\\ \text{growth}}}\cdot\min\left(\underbrace{\frac{P2}{P2+K_{HS,Phyto,P2}}}_{\substack{\text{Limiting effect of}\\ \text{dissolved reactive}\\ \text{phosphorus on}\\ \text{phytoplankton}\\ \text{growth}}},\underbrace{\frac{2.718\cdot f_{day}}{K_e\cdot H}\cdot\left(\exp\left(-\frac{I_A}{I_S}\cdot\exp(-K_e\cdot H)\right)-\exp\left(-\frac{I_A}{I_S}\right)\right)}_{\substack{\text{Limiting effect of}\\ \text{light on}\\ \text{phytoplankton}\\ \text{growth}}}\right)\qquad\text{(Equation A.4)}$$

$$K_e=K_{b,e}+0.0088\cdot Chl\text{-}A+0.054\cdot Chl\text{-}A^{\frac{2}{3}}\qquad\text{(Equation A.5)}$$

$$D_P = \underbrace{\mu_R \cdot \theta_{Phyto,D}^{TEMP-20}}_{\substack{\text{non-predatory death}\\ \text{of phytoplankton}}} + \underbrace{C_G \cdot a_{Chl} \cdot Z}_{\substack{\text{predatory death}\\ \text{of phytoplankton}\\ \text{by zooplankton}\\ \text{grazing}}} \qquad \text{(Equation A.6)}$$

where TEMP is the temperature, K_e [L^{-1}] is the total light extinction coefficient, $K_{b,e}$ [L^{-1}] is the background light extinction coefficient, f_{day} is the fraction of time with day light, G_{max} is the growth rate constant for phytoplankton at optimum conditions, $K_{HS,Phyto,P2}$ is the half saturation concentration of dissolved reactive phosphorus for phytoplankton growth where a Monod-type relation is assumed [$M{\cdot}L^{-3}$]; I_A and I_S are the available light intensity and saturating light intensity respectively, □$_R$ is the death rate constant of phytoplankton, C_G is the grazing rate of zooplankton [T^{-1}]; $\theta_{Phyto,G}$ and $\theta_{Phyto,D}$ are the temperature coefficients for phytoplankton growth and non-predatory death respectively [T^{-1}] where both processes are assumed to accelerate with increasing temperature and Z is the zooplankton concentration [$M{\cdot}L^{-3}$].

Equations A.1, A.2 and A.3 may seem like linear differential equations, however the term G_P contains non-linear terms making the entire system non-linear since this term is substituted into the differential equation system. Equation A.4 states that the primary production that is the key process for eutrophication is under the influence of two limiting factors: the nutrient availability and the light availability. Equation A.5 states that the more phytoplankton grow, the more light extinction will increase a process known as “algal self shading” so that the model prevents indefinite growth of phytoplankton even if unlimited amount of dissolved reactive phosphorus would be available.

The non-linear structure of Equations A.4 and A.5 make the analytical integration (exact solution) of the differential equation system formed by Equations A.1, A.2 and A.3 impossible. This is usually the case in ecological models. Therefore they have to be solved numerically and approximate solutions will be obtained instead of exact solutions. The next sub-section gives more details on these topics for the example eutrophication model. However, before starting to develop the solution let us classify the terms in equations as given in Table A.1.

The forcing factors and auxiliary variables are changing with time. The forcing factors are external variables that are given to the equations from outside. Auxiliary variables are computed using state variables, forcing factors and other auxiliary variables during the model calculations. Model constants are used during the model calibration and model validation process.

Table 1 Terms in the model equations

Term	Description	Type	Unit
P1	Unavailable (organically bound) phosphorus	State variable	gP·m^{-3}
P2	Available (dissolved reactive) phosphorus	State variable	gP·m^{-3}
Chl-A	Chlorophyll-a	State variable	g Chl-A·m^{-3}
V	Volume	Auxiliary variable	m^{3}
H	Depth of the lake	Auxiliary variable	M
G_P	Growth rate of phytoplankton	Auxiliary variable	day^{-1}
D_P	Death rate of phytoplankton	Auxiliary variable	day^{-1}
K_e	Total light extinction coefficient	Auxiliary variable	m^{-1}
Q_{OUT}	Outflow rate	Forcing factor	m^{3} day^{-1}
$v_{S,P1}$	Settling velocity for unavailable phosphorus	Forcing factor	m day^{-1}
$v_{S,Chl-A}$	Settling velocity for phytoplankton	Forcing factor	m day^{-1}
W_1	Unavailable phosphorus load	Forcing factor	kg day^{-1}
W_2	Available phosphorus load	Forcing factor	kg day^{-1}
W_{Chl-A}	Chlorophyll-a load	Forcing factor	kg day^{-1}
I_A	Available light intensity	Forcing factor	Watt·m^{-2}
TEMP	Temperature	Forcing factor	ºC
f_{day}	Fraction of sunlight hours	Forcing factor	Unitless
Z	Zooplankton concentration	Forcing factor	g·m^{-3}
a_p	Phosphorus to chlorophyll-a ratio	Model constant	Unitless
a_{chl}	Chlorophyll a to zooplankton grazing ratio	Model constant	Unitless
$K_{b,e}$	Background light extinction coefficient	Model constant	m^{-1}
K_{12}	Decomposition rate constant	Model constant	day^{-1}
θ_{Decomp}	Temperature correction coefficient for decomposition	Model constant	Unitless
G_{max}	Maximum growth rate costant of phytoplankton	Model constant	day^{-1}
$K_{HS,Phyto,P2}$	Half saturation concentration of dissolved reactive phosphorus for phytoplankton growth	Model constant	gP·m^{-3}
I_S	Saturation light intensity	Model constant	Watt·m^{-2}
μ_R	Death rate constant of phytoplankton	Model constant	day^{-1}
C_G	Grazing rate constant of zooplankton	Model constant	day^{-1}
$\theta_{Phyto,G}$	Temperature correction coefficient for phytoplankton growth and non-predatory death	Model constant	Unitless
$\theta_{Phyto,D}$	Temperature correction coefficient for non-predatory phytoplankton death	Model constant	Unitless

A.3 Development of solution schemes for the equations

Numerical integration methods convert a system of differential equation into a system algebraic equation that can be solved easily. The differential terms will simply be converted to difference terms, so that the model will have limited number of points in time and/or space. This process is called discretization. There are many discretization algorithms each of which having its own advantages and disadvantages. There are many excellent numerical methods books that contain details related to discretization algorithms. In this example we will use the simplest discretization method, the explicit Euler algorithm. Since the lake is considered as completely mixed in our model, there is no discretization in space and we only need to discretize in time. In other words, time is the only independent variable. Using the Euler algorithm, Equation A.1 can be transformed into Equation A.8. In these equations; Δt which is a small time interval, where everything is assumed to be constant is called the time step. The superscript t means that the value at time point t of the relevant variable ***(not to be confused with the t-th power operation)*** will be used and superscript t+Δt means that the value at time point t+Δt of the relevant variable will be used. In any case, one important assumption is made that within each time steps, some of the variables are assumed to be frozen on time point t and some on time point t+Δt. This will always lead to some errors, since we know that in the real world none of the eutrophication related variables are frozen in time. On the other hand, this is the only way to obtain solutions to the model equations. If we take Δt small enough, the errors will be small and we will end up with an acceptably accurate approximate solution. If Δt is too small, then the calculations will take unnecessary long, if Δt is too large than the solutions will contain larger errors leading to inaccurate results which will diverge from the real case. Further increase of Δt may lead to physically unacceptable results such as negative concentrations and eventually to mathematical errors that may lead the calculation algorithm to destabilize and collapse. This situation is called numerical instability. Unfortunately, for non-linear systems there is no way to tell what could be an optimum value for Δt, but experience shows that for ecological models of completely mixed shallow lakes it is on the order of hours. After the temporal discretization and substation of all terms, and reorganization of terms, the model equations are as follows:

$$\frac{P1^{t+\Delta t}-P1^{t}}{\Delta t}=\frac{W_{P1}{}^{t}}{V^{t}}-\frac{Q_{OUT}{}^{t}}{V^{t}}\cdot P1^{t}-\left(K_{12}{}^{t}+\frac{v_{S,P1}{}^{t}}{H^{t}}\right)\cdot P1^{t}+D_{P}{}^{t}\cdot a_{P}\cdot \mathrm{Chl\text{-}A}^{t} \qquad \text{(Equation A.7)}$$

$$P1^{t+\Delta t}=P1^{t}+\left(\frac{W_{P1}{}^{t}}{V^{t}}-\frac{Q_{OUT}{}^{t}}{V^{t}}\cdot P1^{t}-\left(K_{12}{}^{t}+\frac{v_{S,P1}{}^{t}}{H^{t}}\right)\cdot P1^{t}+D_{P}{}^{t}\cdot a_{P}\cdot \mathrm{Chl\text{-}A}^{t}\right)\cdot\Delta t \qquad \text{(Equation A.8)}$$

$$P1^{t+\Delta t}=P1^{t}+\left(\underbrace{\frac{W_{P1}{}^{t}}{\overline{V}^{t}}-\left(\frac{Q_{OUT}{}^{t}}{\overline{V}}+\frac{v_{S,P1}{}^{t}}{\overline{H}}\right)\cdot P1^{t}}_{\text{Transport derivative}}+\underbrace{\left(\mu_{R}\cdot\theta_{\mathrm{Phyto,D}}{}^{\mathrm{TEMP}-20}+C_{G}{}^{t}\cdot Z^{t}\right)\cdot a_{P}\cdot \mathrm{Chl\text{-}A}^{t}-K_{12}\cdot\theta_{\mathrm{Phyto,D}}{}^{\mathrm{TEMP}-20}\cdot P1^{t}}_{\text{Kinetic derivative}}\right)\cdot\Delta t \qquad \text{(Equation A.9)}$$

$$P2^{t+\Delta t}=P2^{t}+\left(\underbrace{\frac{W_{P2}{}^{t}}{\overline{V}}-\frac{Q_{OUT}{}^{t}}{\overline{V}}\cdot P2^{t}}_{\text{Transport derivative}}+\underbrace{K_{12}{}^{t}\cdot\theta_{\mathrm{Phyto,D}}{}^{\mathrm{TEMP}-20}\cdot P1^{t}-G_{P}{}^{t}\cdot a_{P}\cdot \mathrm{Chl\text{-}A}^{t}}_{\text{Kinetic derivative}}\right)\cdot\Delta t \qquad \text{(Equation A.10)}$$

$$\mathrm{Chl\text{-}A}^{t+\Delta t}=\mathrm{Chl\text{-}A}^{t}+\left(\underbrace{\frac{W_{\mathrm{Chl\text{-}A}}{}^{t}}{\overline{V}}-\left(\frac{Q_{OUT}{}^{t}}{\overline{V}}+\frac{v_{S,\mathrm{Phyto}}{}^{t}}{\overline{H}}\right)\cdot \mathrm{Chl\text{-}A}^{t}}_{\text{Transport derivative}}+\underbrace{\left(G_{P}{}^{t}-\mu_{R}\cdot\theta_{\mathrm{Phyto,D}}{}^{\mathrm{TEMP}-20}-C_{G}{}^{t}\cdot Z^{t}\right)\cdot \mathrm{Chl\text{-}A}^{t}}_{\text{Kinetic derivative}}\right)\cdot\Delta t \qquad \text{(Equation A.11)}$$

$$G_{P}{}^{t}=G_{MAX}\cdot\theta_{\mathrm{Phyto,G}}{}^{\mathrm{TEMP}-20}\cdot\min\left(\frac{P2^{t}}{P2^{t}+K_{\mathrm{HS,Phyto,P2}}},\frac{2.718\cdot f_{\mathrm{day}}^{t}}{K_{e}{}^{t}\cdot\overline{H}}\cdot\left(\exp\left(-\frac{I_{A}{}^{t}}{I_{S}}\cdot\exp\left(-K_{e}{}^{t}\cdot\overline{H}\right)\right)-\exp\left(-\frac{I_{A}{}^{t}}{I_{S}}\right)\right)\right) \qquad \text{(Equation A.12)}$$

$$K_{e}{}^{t}=K_{b,e}+0.0088\cdot \mathrm{Chl}-A^{t}+\left(0.054\cdot\left(\mathrm{Chl}-A^{t}\right)^{\frac{2}{3}}\right) \qquad \text{(Equation A.13)}$$

$$\overline{V}=\frac{V^{t}+V^{t+\Delta t}}{2}=\frac{V^{t}}{2}+\frac{V^{t}+\left(Q_{in}^{t}-Q_{out}^{t}\right)\cdot\Delta t}{2}=\frac{2\cdot V^{t}+\left(Q_{in}^{t}-Q_{out}^{t}\right)\cdot\Delta t}{2} \qquad \text{(Equation A.14)}$$

$$\overline{H}=f\left(\overline{V}\right) \qquad \text{(Equation A.15)}$$

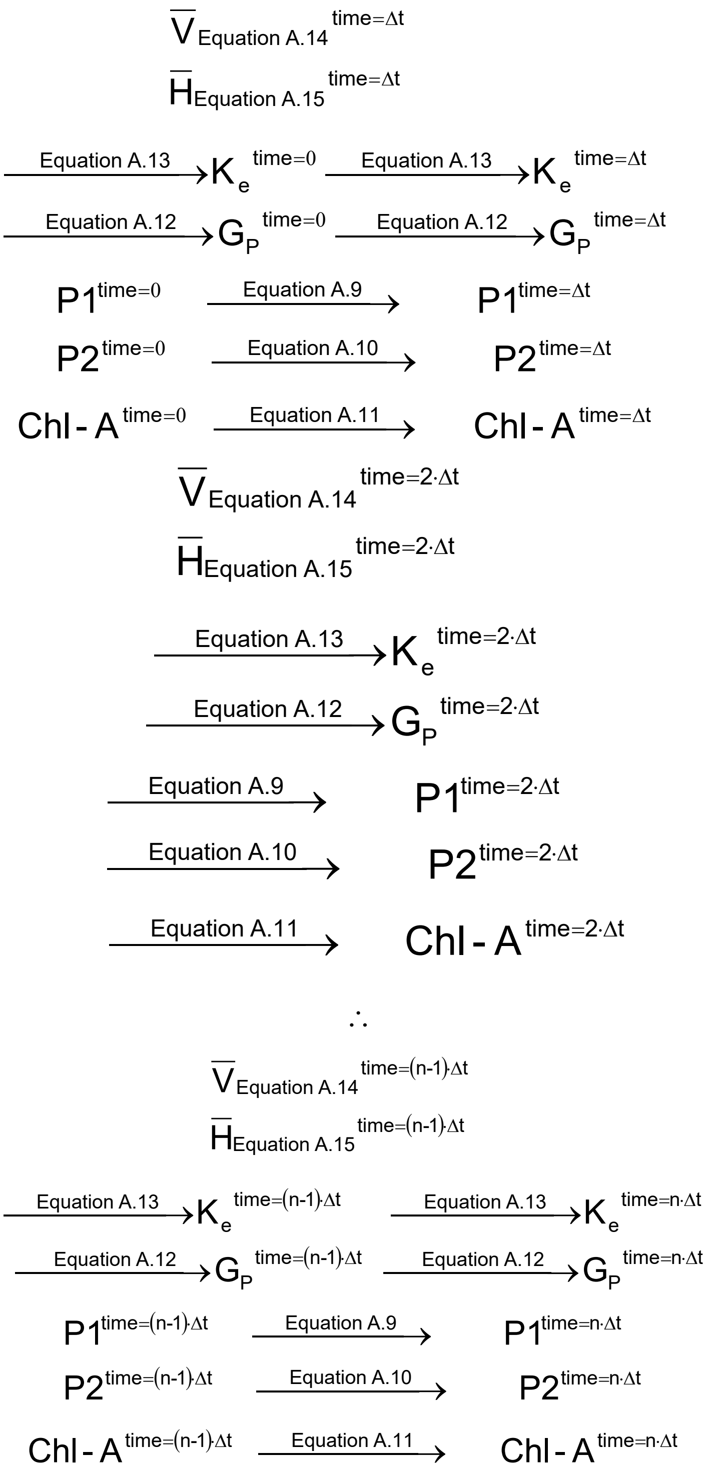
where the terms $\overline{V}$ and $\overline{H}$ represent the volume and spatially mean depth averaged over the time step. The spatially mean depth is assumed to be a function of volume. The derivative terms in equations A.9 to A.11 are divided into transport and kinetic derivatives. This is a good model development practice since transport equations do not change usually, but the kinetic relations may need to be changed if the model has to be extended to include different eutrophication mechanisms.

It is important to consider during the implementation of the model as a tool is that, any variable regardless a state variable, auxiliary variable, forcing factor with a temporal superscript (t+Δt or t) should change in time. The values of P1, P2 and Chl-A must be calculated for each time step from the values of all variables that are superscripted as t. In other words, if the time step is one hour and the model has to be run for a year, Equations A.9 to A.12 must be solved 24· 365 = 8760 times (8760 + 24 = 8784 times in case of a leap year). From this figures it is clear that computers will be needed to implement the model developed in this section.

A.4. Implementation of the model as a tool

The implementation algorithm of the solution illustrated in Figure 2 is simple and straightforward, where n is the number of time steps for the entire simulation. However; the solution algorithm itself as implemented directly is not a complete modelling tool.

Referring to Section A.2, there are terms in equations that have to be updated every time step. The forcing factors are given to the model from outside so they have to be organized as time series. One option is to read one element for each time step. In the example model however, we will use daily time series for the sake of simplicity. The auxiliary variables should be updated every time step. Considering these necessities, the flowchart of all operations to be conducted by the model is as illustrated in Figure A.3.



V̄ Equation A.14 time=Δt
H̄ Equation A.15 time=Δt
Equation A.13 → Ke time=0
Equation A.13 → Ke time=Δt
Equation A.12 → GP time=0
Equation A.12 → GP time=Δt
P1 time=0
Equation A.9
P1 time=Δt
P2 time=0
Equation A.10
P2 time=Δt
Chl - A time=0
Equation A.11
Chl - A time=Δt
V̄ Equation A.14 time=2·Δt
H̄ Equation A.15 time=2·Δt
Equation A.13 → Ke time=2·Δt
Equation A.12 → GP time=2·Δt
Equation A.9 → P1 time=2·Δt
Equation A.10 → P2 time=2·Δt
Equation A.11 → Chl - A time=2·Δt
∴
V̄ Equation A.14 time=(n-1)·Δt
H̄ Equation A.15 time=(n-1)·Δt
Equation A.13 → Ke time=(n-1)·Δt
Equation A.13 → Ke time=n·Δt
Equation A.12 → GP time=(n-1)·Δt
Equation A.12 → GP time=n·Δt
P1 time=(n-1)·Δt
Equation A.9
P1 time=n·Δt
P2 time=(n-1)·Δt
Equation A.10
P2 time=n·Δt
Chl - A time=(n-1)·Δt
Equation A.11
Chl - A time=n·Δt

Figure A.2. The implementation algorithm of the model

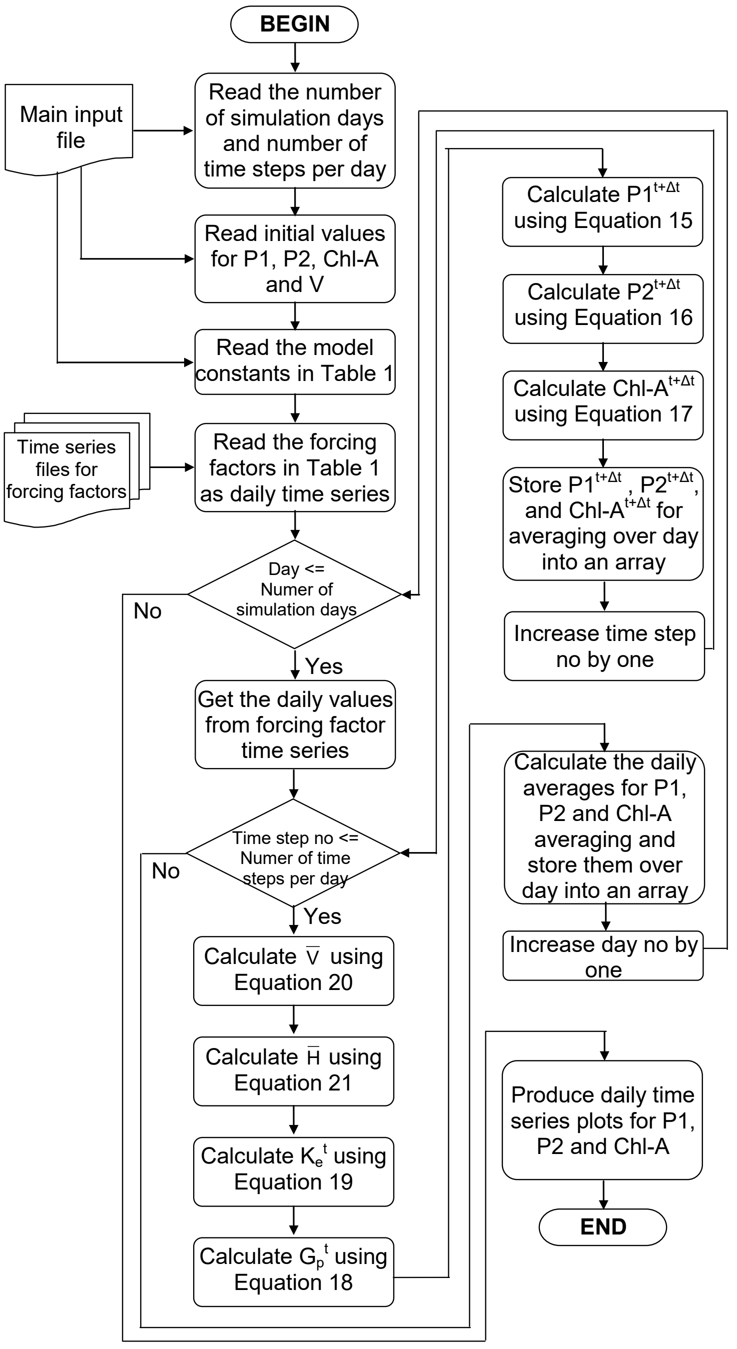

BEGIN
Main input file
Read the number of simulation days and number of time steps per day
Read initial values for P1, P2, Chl-A and V
Read the model constants in Table 1
Time series files for forcing factors
Read the forcing factors in Table 1 as daily time series
Day <= Numer of simulation days
No
Yes
Get the daily values from forcing factor time series
Time step no <= Numer of time steps per day
No
Yes
Calculate $\overline{V}$ using Equation 20
Calculate $\overline{H}$ using Equation 21
Calculate K_e^t using Equation 19
Calculate G_p^t using Equation 18
Calculate $P1^{t+\Delta t}$ using Equation 15
Calculate $P2^{t+\Delta t}$ using Equation 16
Calculate Chl-$A^{t+\Delta t}$ using Equation 17
Store $P1^{t+\Delta t}$, $P2^{t+\Delta t}$, and Chl-$A^{t+\Delta t}$ for averaging over day into an array
Increase time step no by one
Calculate the daily averages for P1, P2 and Chl-A averaging and store them over day into an array
Increase day no by one
Produce daily time series plots for P1, P2 and Chl-A
END

Figure A.3. Flowchart of the operations to be conducted by the model

The flowchart given in Figure A.3. can be coded in any programming language. For this model, we will use MATLAB because

- MATLAB is easy to program
- MATLAB gives a good post processing environment
- MATLAB has many tools that ease to develop a supporting environment for the model

The source code of the model is given below. The program consists of four MATLAB functions listed below:

- Phos_ChlA_Model.m: It is the main function that forms the body of the model following the algorithm that is given in flowchart illustrated in Figure 3.
- TRANSPORT.m: It handles the transport related terms in Equations A.9, A.10 and A.11. If the effects of additional inputs or outflows of nutrients and phytoplankton such as new inflowing or out flowing rivers, then this function will be changed by the user.
- KINETICS.m: It handles the kinetics related terms in Equations A.9, A.10 and A.11. New processes have to be added to the model, then this function will be changed by the user.
- GET_SPATIALLY_AVERAGED_DEPTH.m: It calculates the depth from the volume. According to the source code given in Appendix A, it is designed to calculate the depth of a rectangular prismatic basin with a surface area of 166 km^2. However; it can be extended to handle a variety of situations with the capabilities provided by MATLAB.

The reader should know the basics of MATLAB before starting to explore and modify the code. To keep the example code simple, advanced numerical integration techniques available in MATLAB were not used.

A.5. Development of the supporting environment and tools for the model

Many advanced eutrophication models such as WASP, AQUATOX, EFDC, Delft3D-DELWAQ, MOHID and MIKE have their own graphical model input editors, post processors and even data exchange capabilities with geographical information systems. The example eutrophication model here reads text files or csv files created/edited with text editors or spreadsheet programs and it produces a simple time series plot as output, so one can say that the model has no supporting environment that would ease the modelling process. However, MATLAB itself can be considered as supporting environment for the model developed in this example:

- The main body of the model (function Phos_ChlA_Model) is designed to produce a matrix output for daily results with time index. MATLAB has very powerful functions to process such matrices. For example, one can easily do statistical analysis of the columns. MATLAB has also many options for plotting so that within few commands it would be possible to write powerful graphs. All this commands could be written as scripts each of them not exceeding ten lines of code.
- MATLAB has powerful tools that are useful for general modelling steps such as autocalibration, optimization or more advanced modelling steps such as uncertainty analysis or parameter identification.

Source code of the simple eutrophication model

Phos_ChlA_Model.m

```
function [DAILY_P1, DAILY_P2, DAILY_CHLA] = Phos_ChlA_Model()
    %Read the main input file
    MAIN_INPUT_DATA        = load('MAIN_INPUT.inp');
    NUM_DAYS               = MAIN_INPUT_DATA(1);
    NUM_TIME_STEPS_PER_DAY = MAIN_INPUT_DATA(2);

    %Read the initial conditions
    INITIAL_CONDIITONS     = load('INITIAL_CONDITIONS.inp');
    V_T                    = INITIAL_CONDIITONS(4);

    %Read the model constants file
    MODEL_CONSTANTS        = load('MODEL_CONSTANTS.inp');

    %Read the forcing factors as the daily time series files
    Q_IN_TS                = load('Q_IN.inp');
    Q_OUT_TS               = load('Q_OUT.inp');
    VS_P1_TS               = load('VS_P1.inp');
    VS_CHLA_TS             = load('VS_CHLA.inp');
    W_P1_TS                = load('W_P1.inp');
    W_P2_TS                = load('W_P2.inp');
    W_CHLA_TS              = load('W_CHLA.inp');
    I_A_TS                 = load('I_A.inp');
    TEMP_TS                = load('TEMP.inp');
    FDAY_TS                = load('FDAY.inp');
    Z_TS                   = load('Z.inp');

    STATE_VARIABLES(1)     = INITIAL_CONDIITONS(1);
    STATE_VARIABLES(2)     = INITIAL_CONDIITONS(2);
    STATE_VARIABLES(3)     = INITIAL_CONDIITONS(3);
    DAILY_P1   = zeros(NUM_DAYS);
    DAILY_P2   = zeros(NUM_DAYS);
    DAILY_CHLA = zeros(NUM_DAYS);

    for DAY = 1:NUM_DAYS
        %Get daily values from forcing time series
        FORCING_FACTORS = ...
            [interp1(Q_IN_TS   (:,1), Q_IN_TS   (:,2), DAY_NO + 0.5), ...
             interp1(Q_OUT_TS  (:,1), Q_OUT_TS  (:,2), DAY_NO + 0.5), ...
             interp1(VS_P1_TS  (:,1), VS_P1_TS  (:,2), DAY_NO + 0.5), ...
             interp1(VS_CHLA_TS(:,1), VS_CHLA_TS(:,2), DAY_NO + 0.5), ...
             interp1(W_P1_TS   (:,1), W_P1_TS   (:,2), DAY_NO + 0.5), ...
             interp1(W_P2_TS   (:,1), W_P2_TS   (:,2), DAY_NO + 0.5), ...
             interp1(W_CHLA_TS (:,1), W_CHLA_TS (:,2), DAY_NO + 0.5), ...
             interp1(I_A_TS    (:,1), I_A_TS    (:,2), DAY_NO + 0.5), ...
             interp1(TEMP_TS   (:,1), TEMP_TS   (:,2), DAY_NO + 0.5), ...
             interp1(FDAY_TS   (:,1), FDAY_TS   (:,2), DAY_NO + 0.5), ...
             interp1(Z_TS      (:,1), Z_TS      (:,2), DAY_NO + 0.5)];

        P1_ARRAY   = zeros(NUM_TIME_STEPS_PER_DAY);
        P2_ARRAY   = zeros(NUM_TIME_STEPS_PER_DAY);
        CHLA_ARRAY = zeros(NUM_TIME_STEPS_PER_DAY);
        for TIME_STEP_NO = 1:NUM_TIME_STEPS_PER_DAY
            %Calculate V using Equation 20
            V = (2 * V_T + ((Q_IN_T - Q_OUT_T) * DT)) / 2;

            %Calculate H using Equation 21
            H = GET_SPATIALLY_AVERAGED_DEPTH(V);

            AUXILLARY_VARIABLES = [V, H];

            %Calculate the transport derivatives
            TRANSPORT_DERIVATIVES = ...
                KINETICS(STATE_VARIABLES, AUXILLARY_VARIABLES,FORCING_FACTORS);
            P1_TRANSPORT_DERIV   = TRANSPORT_DERIVATIVES(1);
            P2_TRANSPORT_DERIV   = TRANSPORT_DERIVATIVES(2);
            CHLA_TRANSPORT_DERIV = TRANSPORT_DERIVATIVES(3);

            %Calculate the kinetic derivatives
            KINETIC_DERIVATIVES = ...
                KINETICS(STATE_VARIABLES, AUXILLARY_VARIABLES, FORCING_FACTORS, MODEL_CONSTANTS);

            P1_KINETIC_DERIV   = KINETIC_DERIVATIVES(1);
            P2_KINETIC_DERIV   = KINETIC_DERIVATIVES(2);
            CHLA_KINETIC_DERIV = KINETIC_DERIVATIVES(3);
            P1_T               = STATE_VARIABLES(1);
            P2_T               = STATE_VARIABLES(2);
            CHLA_T             = STATE_VARIABLES(3);

            %Calculate P1_T_PLUS_DT using Equation 15
            P1_T_PLUS_DT   = P1_T + (P1_TRANSPORT_DERIV + P1_KINETIC_DERIV) * DELTA_T;

            %Calculate P2_T_PLUS_DT using Equation 16
            P2_T_PLUS_DT   = P2_T + (P2_TRANSPORT_DERIV + P2_KINETIC_DERIV) * DELTA_T;

            %Calculate CHLA_T_PLUS_DT using Equation 17
            CHLA_T_PLUS_DT = CHLA_T + (CHLA_TRANSPORT_DERIV + CHLA_KINETIC_DERIV) * DELTA_T;
            P1_ARRAY(TIME_STEP_NO)   = P1_PLUS_DT;
            P2_ARRAY(TIME_STEP_NO)   = P2_PLUS_DT;
            CHLA_ARRAY(TIME_STEP_NO) = CHLA_T_PLUS_DT;
            STATE_VARIABLES(1) = P1_T_PLUS_DT;
            STATE_VARIABLES(2) = P2_T_PLUS_DT;
            STATE_VARIABLES(3) = CHLA_T_PLUS_DT;
            V_T = V;
        end

        DAILY_P1(DAY_NO)   = mean(P1_ARRAY);
        DAILY_P2(DAY_NO)   = mean(P1_ARRAY);
        DAILY_CHLA(DAY_NO) = mean(P1_ARRAY);
    end

    DAYS = 1:NUM_DAYS;
    plot(DAYS, DAILY_P1(DAY_NO), DAYS, DAILY_P2(DAY_NO), DAYS, DAILY_CHLA(DAY_NO));
end
```

KINETICS.m

```
function [P1_KIN_DERIV, P2_KIN_DERIV, CHLA_KIN_DERIV] = KINETICS ...
         (STATE_VARIABLES, AUXILLARY_VARIABLES, FORCING_FACTORS, ...
          MODEL_CONSTANTS)
    P1            = STATE_VARIABLES(1);
    P2            = STATE_VARIABLES(2);
    CHLA          = STATE_VARIABLES(3);
    H             = AUXILLARY_VARIABLES(1);
    I_A_T         = FORCING_FACTORS(8);
    TEMP_T        = FORCING_FACTORS(9);
    FDAY_T        = FORCING_FACTORS(10);
    Z_T           = FORCING_FACTORS(11);
    A_P           = MODEL_CONSTANTS(1);
    K_B_E         = MODEL_CONSTANTS(2);
    K_12          = MODEL_CONSTANTS(3);
    THETA_DECOMP  = MODEL_CONSTANTS(4);
    G_MAX         = MODEL_CONSTANTS(5);
    K_HS_PHYTO_P2 = MODEL_CONSTANTS(6);
    I_S           = MODEL_CONSTANTS(7);
    MU_R          = MODEL_CONSTANTS(8);
    C_G           = MODEL_CONSTANTS(9);
    THETA_PHYTO_G = MODEL_CONSTANTS(10);
    THETA_PHYTO_D = MODEL_CONSTANTS(11);

    %Calculate K_E_T using Equation 19
    K_E_T = K_B_E + (0.088 * CHLA) + (0.054 * (CHLA.^(2/3)));

    %Calculate G_P_T using Equation 18
    G_P_T = G_MAX * (THETA_PHYTO_G.^(TEMP - 20)) * ...
            min((P2 / (P2 + K_HS_PHYTO_P2)),(((2.718 * FDAY_T) / (K_E_T * H)) * ...
                 (exp(-(I_A_T / I_S) * exp(-K_E_T * H)) - exp(-(I_A_T / I_S)))));

    %Calculate the kinetic derivative for P1
    P1_KIN_DERIV   = (MU_R * THETA_PHYTO_D.^(TEMP_T - 20)) * A_P * CHLA - ...
                     (K_12 * THETA_DECOMP.^(TEMP_T - 20)) * P1;

    %Calculate the kinetic derivative for P2
    P2_KIN_DERIV   = (K_12 * THETA_DECOMP.^(TEMP_T - 20)) * P1 - G_P_T * A_P * CHLA;

    %Calculate the kinetic derivative for CHLA
    CHLA_KIN_DERIV =  (G_P_T - MU_R * THETA_PHYTO_D.^(TEMP_T - 20) - C_G * Z_T) * A_P * CHLA;
end
```

TRANSPORT.m

```
function [P1_TRANS_DERIV, P2_TRANS_DERIV, CHLA_TRANS_DERIV] = TRANSPORT ...
         (STATE_VARIABLES, AUXILLARY_VARIABLES, FORCING_FACTORS)
    P1        = STATE_VARIABLES(1);
    P2        = STATE_VARIABLES(2);
    CHLA      = STATE_VARIABLES(3);
    V         = AUXILLARY_VARIABLES(1);
    H         = AUXILLARY_VARIABLES(2);
    Q_OUT_T   = FORCING_FACTORS(2);
    VS_P1_T   = FORCING_FACTORS(3);
    VS_CHLA_T = FORCING_FACTORS(4);
    W_P1_T    = FORCING_FACTORS(5);
    W_P2_T    = FORCING_FACTORS(6);
    W_CHLA_T  = FORCING_FACTORS(7);

    %Calculate the kinetic derivative for P1
    P1_TRANS_DERIV   = (W_P1_T / V) - ((Q_OUT_T / V) + (VS_P1_T / H)) * P1;

    %Calculate the kinetic derivative for P2
    P2_TRANS_DERIV   = (W_P2_T / V) - (Q_OUT_T / V) * P2;

    %Calculate the kinetic derivative for CHLA
    CHLA_TRANS_DERIV =  (W_CHLA_T / V) - .((Q_OUT / V) + (VS_CHLA_T / H)) * CHLA;
end
```

GET_SPATIALLY_AVERAGED_DEPTH.m

```
function [DEPTH] = GET_SPATIALLY_AVERAGED_DEPTH(V)
    % A very simple representation of bathymetry. It is the users
    % to program this function to represent the bathymetry of his/her lake
    % realistically
    AREA  = 166000000;
    DEPTH = V / AREA;
end
```

APPENDIX B

Solving the Hydrodynamic Equations

Hydrodynamic equations are relatively more difficult to solve. The task becomes more challenging if they have to be solved in three dimensions. Here, the solution of simplified version of two dimensional hydrodynamic equations is shown. This type of hydrodynamic models can be applied to deep and narrow waterbodies that are well mixed laterally but not longitudinally or vertically.

There are two basic equations in hydrodynamics, the continuity equation and the momentum equation. The simplified versions of these equations for the so called x-z model are given below,

Continuity equation in longitudinal and vertical dimensions:

$$\frac{\partial V}{\partial t} + \frac{\partial U}{\partial x} = 0$$

(Equation B.1)

Momentum equation in longitudinal and vertical dimensions:

$$B \cdot \frac{\partial u}{\partial t} = -g \cdot B \cdot \frac{\partial \eta}{\partial x} + \frac{\partial}{\partial z}\left(A_z \cdot B \cdot \frac{\partial u}{\partial z} \right) - \text{friction terms}$$

(Equation B.2)

where, U is the velocity integrated over the depth therefore in volumetric flow rate dimensions [$L^3 \cdot T^{-1}$], V is the control volume over which the continuity equation is solved [L^3], B is the width the channel as a function of depth [L], g is the gravitational acceleration [$L \cdot T^{-2}$] u is the velocity in x direction [L], η is the water surface elevation [L] and A_z is the eddy viscosity. Equation B2 is derived from the Navier-Stokes equations, however it is simplified so that the non-linear advective inertia terms are neglected.

Even though the hydrodynamic equations are in simplified form, there is no direct analytical solution for them. Therefore, numerical solution schemes have to be applied. To apply a numerical solution scheme it is necessary to discretize the hydrodynamic equation on time and space.

There are many spatial discretization schemes for hydrodynamic equations. In this example, we will use the cell-link scheme. This scheme is efficient and easy to apply. If is more compatible with box models, if the results from the hydrodynamic model are to be imported into a box model described by Equation 14. In this discretization scheme, a waterbody is assumed to consist of basins that exchange water with each other through channels. An example model domain is given in Figure B.1. Each basin is represented with a cell and each channel is represented with a link. A link is between two cells. The flow directions are assumed to be from one cell (begin) to another (end). If the actual flow direction is opposite or the flow reveres during the simulation, then a simply a negative velocity will be returned by the model.

The surface area of a channel is calculated by Equation B.3

$$A_{surf} = \sum_{\text{channel}=1}^{\substack{\text{Number of}\\ \text{channels}}} \frac{\Delta x_{channel} \cdot B_{channel}}{2}$$

(Equation B.3)

Where A_{surf} is the surface area [L^2], $\Delta x_{channel}$ is the length of the channel that is connecting a cell with another cell [L]. If Equation B.1 is divided by the surface area, then Equation B.4 is obtained,

$$\frac{\partial \eta}{\partial t} + \frac{1}{A_{surf}} \cdot \frac{\partial U}{\partial x} = 0$$

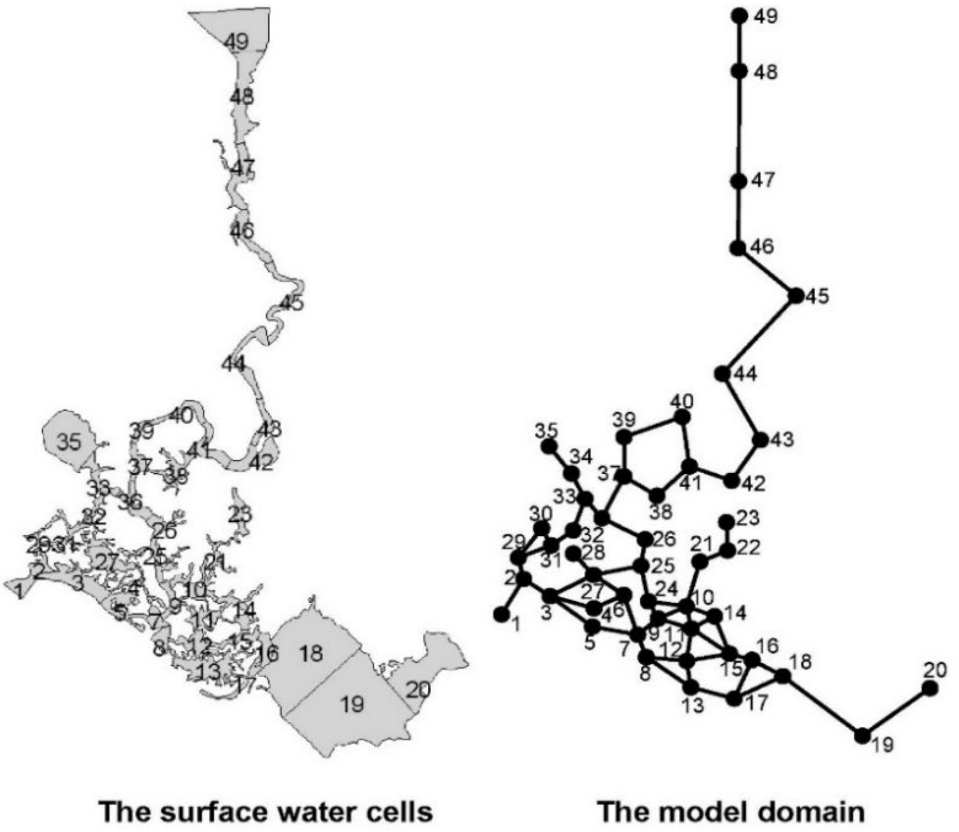
(Equation B.4)

where the term $\frac{\partial \eta}{\partial t}$ would correspond to the change of depth over time. Since it is a derivative and chance of depth over time is the same as change of water surface elevation over time, we can use the term as change of water surface elevation over time. The latter definition is more convenient to solve the momentum equation. Discretizing Equation B.4 spatially according to cell, node scheme yields Equation B.5.

$$\frac{\partial \eta_{Cell}}{\partial t} = \frac{1}{A_{surf}} \cdot \left(\sum_{\text{Inflow}=1}^{\substack{\text{Number of}\\ \text{inflowing}\\ \text{channels}}} \sum_{k=1}^{\substack{\text{Number of}\\ \text{layers}}} u_{Inflow,k} \cdot h_k \cdot B_k - \sum_{\text{Outflow}=1}^{\substack{\text{Number of}\\ \text{outflowing}\\ \text{channels}}} \sum_{k=1}^{\substack{\text{Number of}\\ \text{layers}}} u_{Outflow} \cdot h_k \cdot B_k \right)$$

(Equation B.5)

where u is the velocity in a layer of a channel [$L \cdot T^{-1}$], h is the layer depth [L] and B is the layer width [L]. Discretizing Equation B.5 over time yields Equation B.6.



The surface water cells

The model domain

Channel No	Cell 1	Cell 2
1	2	1
2	3	2
3	5	3
4	4	3
5	27	3
6	7	5
7	6	4
8	27	6
9	28	27
10	25	27
11	6	7
12	25	24
13	24	9
14	24	10
15	10	9
16	9	7
17	8	7
18	12	8
19	9	11
20	10	11
21	10	14

Channel No	Cell 1	Cell 2
22	14	15
23	11	12
24	14	15
25	11	15
26	15	12
27	12	13
28	13	8
29	17	13
30	15	16
31	16	17
32	16	18
33	18	17
34	19	18
35	20	19
36	21	10
37	22	21
38	23	22
39	29	2
40	30	29
41	31	29
42	31	30

Channel No	Cell 1	Cell 2
43	32	31
44	33	32
45	33	34
46	34	35
47	36	33
48	36	26
49	26	25
50	37	36
51	38	37
52	41	38
53	39	37
54	40	39
55	41	40
56	42	41
57	43	42
58	44	43
59	45	44
60	46	45
61	47	46
62	48	47
63	49	48

Channel No	Cell 1	Cell 2
64	Koycegiz Lake Boundary	49
65	1	Mediterranean Sea Boundary

Figure B.1. The cell-node scheme discretization

The discretization of the momentum equation (Equation B.2) is more complicated. The momentum equation must be discretized horizontally on channels and vertically for each layer, where each layer is interacting with the layers above and below. The first layer is interacting with the atmosphere and the second layer and the lowest layer is interacting with its upper layer and the bottom. Each laver has an upper and lower interaction. The general form of discretized momentum equation for a channel is given in Equation B.7 to B.9.

$$\eta_{Cell}^{t+\Delta t} = \eta_{Cell}^{t} + \frac{\Delta t}{A_{surf}} \cdot \left(\sum_{Inflow=1}^{\substack{\text{Number of}\\ \text{inflowing}\\ \text{channels}}} \sum_{k=1}^{\substack{\text{Number of}\\ \text{layers}}} u_{Inflow,k} \cdot h_k \cdot B_k - \sum_{Outflow=1}^{\substack{\text{Number of}\\ \text{outflowing}\\ \text{channels}}} \sum_{k=1}^{\substack{\text{Number of}\\ \text{layers}}} u_{Outflow} \cdot h_k \cdot B_k \right) \quad \text{(Equation B.6)}$$

$$-B_k \cdot \Delta t \cdot C_{upper} \cdot u_{k-1}^{t+\Delta t} + \left(B_k \cdot h_k^{t+\Delta t} + B_k \cdot \Delta t \cdot C_{upper} + B_k \cdot \Delta t \cdot C_{lower}\right) \cdot u_k^{t+\Delta t} - B_k \cdot \Delta t \cdot C_{lower} \cdot u_{k+1}^{t+\Delta t} =$$
$$B_k \cdot h_k^{t} \cdot u_k^{t} - g \cdot h_k^{t+\Delta t} \cdot B_k \cdot \frac{\eta_{end}^{t+\Delta t} - \eta_{begin}^{t+\Delta t}}{\Delta x} \cdot \Delta t \quad \text{(Equation B.7)}$$

$$C_{upper} = \frac{2 \cdot A_z}{h_{k-1} + h_k} \quad \text{(Equation B.8)}$$

$$C_{lower} = \frac{2 \cdot A_z}{h_k + h_{k+1}} \quad \text{(Equation B.9)}$$

For the first layer, Equation B.10 and Equation B.11 are substituted into Equation B.7

$$C_{upper} = 0 \quad \text{(Equation B.10)}$$

k=1

$$\left(B_1 \cdot h_1^{t+\Delta t} + B_1 \cdot \Delta t \cdot \frac{2 \cdot A_z}{h_1^{t+\Delta t} + h_2^{t+\Delta t}}\right) \cdot u_1^{t+\Delta t} - B_1 \cdot \Delta t \cdot \frac{2 \cdot A_z}{h_1^{t+\Delta t} + h_2^{t+\Delta t}} \cdot u_2^{t+\Delta t} = B_1 \cdot h_1^{t+\Delta t} \cdot \left(u_1^{t} - g \cdot \frac{\eta_{end}^{t+\Delta t} - \eta_{begin}^{t+\Delta t}}{\Delta x} \cdot \Delta t\right) \quad \text{(Equation B.11)}$$

Equation B.10 indicates that there is no interaction with the atmosphere. If the interaction with the atmosphere such as the effect of wind shear stress is important, then Equation B.10 should be modified accordingly. For an intermediate layer Equation B.8 and Equation B.9 are substituted into Equation B.7

$$-B_k \cdot \Delta t \cdot \frac{2 \cdot A_z}{h_{k-1}^{t+\Delta t} + h_k^{t+\Delta t}} \cdot u_{k-1}^{t+\Delta t} + \left(B_k \cdot \left(h_k^{t+\Delta t} + 2 \cdot A_z \cdot \Delta t \left(\frac{1}{h_{k-1}^{t+\Delta t} + h_k^{t+\Delta t}} + \frac{1}{h_k^{t+\Delta t} + h_{k+1}^{t+\Delta t}} \right)\right)\right) \cdot u_k^{t+\Delta t} -$$
$$B_k \cdot \Delta t \cdot \frac{2 \cdot A_z}{h_k^{t+\Delta t} + h_{k+1}^{t+\Delta t}} \cdot u_{k+1}^{t+\Delta t} = B_k \cdot h_k^{t+\Delta t} \cdot \left(u_k^{t} - g \cdot \frac{\eta_{end}^{t+\Delta t} - \eta_{begin}^{t+\Delta t}}{\Delta x} \cdot \Delta t \right) \quad \text{(Equation B.12)}$$

For the last layer Equations B.13 to Equation B.16 are substituted into Equation B.7

$$C_{upper} = \frac{2 \cdot A_z}{h_{k-1} + h_k} \quad \text{(Equation B.13)}$$

$$C_{lower}=C_D\cdot\left|u_L^t\right| \qquad \text{(Equation B.14)}$$

$$u_{k+1}^{t+\Delta t}=0 \qquad \text{(Equation B.15)}$$

$$k=L \qquad \text{(Equation B.16)}$$

where L is the number of layers

$$-B_k\cdot\Delta t\cdot\frac{2\cdot A_z}{h_{k-1}^{t+\Delta t}+h_k^{t+\Delta t}}\cdot u_{L-1}^{t+\Delta t}+\left(B_L\cdot\left(h_L^{t+\Delta t}+\Delta t\cdot\left(\frac{2\cdot A_z}{h_{k-1}^{t+\Delta t}+h_k^{t+\Delta t}}+C_D\cdot\left|u_L^t\right|\right)\right)\right)\cdot u_L^{t+\Delta t}= B_L\cdot h_L^{t+\Delta t}\cdot\left(u_L^t-g\cdot\frac{\eta_{end}^{t+\Delta t}-\eta_{begin}^{t+\Delta t}}{\Delta x}\cdot\Delta t\right) \qquad \text{(Equation B.17)}$$

Equation B.13 represents the friction whereas Equation B.14 is the boundary condition for the velocity. The eddy viscosity A_Z can be calculated by a number of formulae. The steps of the solution algorithm are simple and straightforward:

- Step 1: Solve equation B.6 for each cell to calculate $\eta_{Cell}^{t+\Delta t}$
- Step 2: Solve equation B.7 for each cell to channel in each layer to calculate $u_L^{t+\Delta t}$

As seen by the general form of discretized momentum equation (Equation B.7), $u_L^{t+\Delta t}$ depends on $u_{k-1}^{t+\Delta t}$ and $u_{k+1}^{t+\Delta t}$. In other words, the velocity of a layer at the next time steps depends on the velocities on the upper and lower layers. This means that the momentum equations for all the layers in a channel have to be solved simultaneously. Another issue is that, the horizontal density differences should be taken into account. If the density currents are important then, the water surface elevation terms $\eta_{begin}^{t+\Delta t}$ and $\eta_{end}^{t+\Delta t}$ must be corrected with pressure gradient terms calculated at the centre of each layer. A free form Fortran source code listing, that would construct the coefficient matrix and constants vector a for a channel is given below:

```
do k = 1, NUM_LAYERS
    C_UP   = 2.0 * A_Z(i,k-1) / (H(i,k-1,2) + H(i,k,2))
    C_DOWN = 2.0 * A_Z(i,k)   / (H(i,k,2) + H(i,k+1,2))
    if(k .eq. 1) C_UPPER = 0.0D0
    if(k .eq. NUM_LAYERS) C_LOWER = 0.0D0

    !Bands of the coefficient matrix C:Subdiagonal, D:Diagonal, E:Super diagonal
    C(k) = (-B(i,k) * DT * C_UPPER)
    D(k) = B(i,k) * H(i,k,2) +  DT * B(i,k) * (C_UPPER + C_LOWER)
    E(k) = (-B(i,k) * DT * C_LOWER)
    LEVEL_GRAD = ITA  (C_END,2) - ITA  (C_BEGIN,2)
    PRESS_GRAD = PRESS(C_END,k) - PRESS(C_BEGIN,k)

    !Constant vector
    F(k) = (B(i,k) * H(i,k,2) * U(i,k,1)) - &
           (B(i,k) * H(i,k,2) * G * ((LEVEL_GRAD + PRESS_GRAD) / DX) * DT)
    if(k.eq.NUM_LAYERS) D(k) = D(k) + (B(i,k) * DT * C_D * dabs(U(i,k,1)))
end do
```

The three-banded matrix can then easily be solved using the Thomas algorithm.

As one can see in the source code listing, pressure gradient terms should be known before solving the momentum equation. Pressure depends on density that depends on salinity and temperature. On the other hand, salinity and temperature depend on the hydrodynamic transport. This means that a mass balance equation must be solved for salinity (Equations B.18 to B.20). Unlike the continuity equation, a negative flow direction is much more important here, because the flow from opposite direction will completely change the amount of salt mass inflow and outflow, since salinity is expected to change in horizontal and vertical directions. Therefore, before solving Equation B.18 for a time step, the upstream and downstream channels for a cell must be determined by checking the velocities. Any channel that causes an inflow is an upstream channel (subscripted as ups), and any channel that causes an outflow is a downstream channel (subscripted as dws).

$$\frac{\partial S_{i,k}}{\partial t} = \frac{1}{V_{i,k}} \left(\begin{aligned} & \sum_{ups=1}^{\substack{\text{Number of}\\ \text{inflowing}\\ \text{channels}}} u_{ups,k} \cdot h_{ups,k} \cdot B_{ups,k} \cdot S_{ups,k} - \sum_{dws=1}^{\substack{\text{Number of}\\ \text{outflowing}\\ \text{channels}}} u_{dws,k} \cdot h_{dws,k} \cdot B_{dws,k} \cdot S_{i,k} + \\ & w_{upper,ups,k} \cdot A_{surf,k} \cdot S_{i,k-1} - w_{upper,dws,k} \cdot A_{surf,k} \cdot S_{i,k} + \\ & w_{lower,ups,k} \cdot A_{surf,k+1} \cdot S_{i,k+1} - w_{lower,dws,k} \cdot A_{surf,k+1} \cdot S_{i,k} + Svdisp + Shdisp \end{aligned} \right) \qquad \text{(Equation B.18)}$$

$$Svdisp = \frac{\partial}{\partial z}\left(K_z \cdot A_{int,v} \cdot \frac{\partial S}{\partial z} \right) \qquad \text{(Equation B.19)}$$

$$Shdisp = \frac{\partial}{\partial x}\left(K_x \cdot A_{int,h} \cdot \frac{\partial S}{\partial x} \right) \qquad \text{(Equation B.20)}$$

The terms used in Equations B.18 to B.20 are listed below:

i : Index for cell no
k : Index for layer no
S : Salinity [$M \cdot L^{-3}$]
$w_{upper,ups,k}$: Downwelling velocity from upper layer (considered as inflow) [$L \cdot T^{-1}$]
$w_{upper,dws,k}$: Upwelling velocity to upper layer (considered as outflow) [$L \cdot T^{-1}$]
$w_{lower,ups,k}$: Upwelling velocity from upper layer (considered as inflow) [$L \cdot T^{-1}$]
$w_{lower,dws,k}$: Downwelling velocity to lower layer (considered as outflow) [$L \cdot T^{-1}$]
Svdisp : Exchange of salt mass by vertical dispersion [$M \cdot T^{-1}$]
Shdisp : Exchange of salt mass by horizontal dispersion [$M \cdot T^{-1}$]
K_z : Vertical eddy diffusivity [$L^2 \cdot T^{-1}$]
K_x : Horizontal eddy diffusivity [$L^2 \cdot T^{-1}$]
$A_{int,v}$: Vertical interface area [L^2]
$A_{int,H}$: Horizontal interface area [L^2]

The vertical dispersion term is discretized as below:

$$\mathrm{Svdiff} = \left(\mathrm{DS}_{\mathrm{upper}} \cdot \left(S_{i,k\text{-}1} - S_{i,k}\right) - \mathrm{DS}_{\mathrm{lower}} \cdot \left(S_{i,k} - S_{i,k+1}\right)\right) \qquad \text{(Equation B.21)}$$

$$\mathrm{DS}_{\mathrm{upper}} = \frac{2 \cdot K_z \cdot A_{\mathrm{surf},i,k}}{h_{k\text{-}1} + h_k} \qquad \text{(Equation B.22)}$$

$$\mathrm{DS}_{\mathrm{lower}} = \frac{2 \cdot K_z \cdot A_{\mathrm{surf},i,k+1}}{h_k + h_{k+1}} \qquad \text{(Equation B.23)}$$

Substitution of Equation 21 into Equation 18 and rearrangements yield Equation B.24 that is the spatially and temporally discretized salt mass balance

$$\begin{aligned}
&-\left(\frac{\Delta t \cdot \left(w_{\mathrm{upper,ups},k} \cdot A_{\mathrm{surf},k}\right) + \mathrm{DS}_{\mathrm{upper}}}{V_{i,k}}\right) \cdot S^{t+\Delta t}_{i,k\text{-}1} + \\
&\left(\frac{\Delta t \cdot \left(w_{\mathrm{upper,dws},k} \cdot A_{\mathrm{surf},k} + w_{\mathrm{lower,dws},k} \cdot A_{\mathrm{surf},k+1} + \mathrm{DS}_{\mathrm{upper}} + \mathrm{DS}_{\mathrm{lower}}\right)}{V_{i,k}} + 1\right) \cdot S^{t+\Delta t}_{i,k} \\
&-\left(\frac{\Delta t \cdot \left(\mathrm{DS}_{\mathrm{lower}} + w_{\mathrm{lower,ups},k} \cdot A_{\mathrm{surf},k+1}\right)}{V_{i,k}}\right) \cdot S^{t+\Delta t}_{i,k+1} = \qquad \text{(Equation B.24)} \\
&S^t_{i,k} + \frac{\Delta t}{V_{i,k}} \cdot \left(\sum_{\mathrm{ups}=1}^{\substack{\text{Number of}\\ \text{inflowing}\\ \text{channels}}} u_{\mathrm{ups},k} \cdot h_{\mathrm{ups},k} \cdot B_{\mathrm{ups},k} \cdot S^t_{\mathrm{ups},k} - \sum_{\mathrm{dws}=1}^{\substack{\text{Number of}\\ \text{outflowing}\\ \text{channels}}} u_{\mathrm{dws},k} \cdot h_{\mathrm{dws},k} \cdot B_{\mathrm{dws},k} \cdot S^t_{i,k} + \sum_{n=1}^{\substack{\text{Number of}\\ \text{neighbouring}\\ \text{cells}}} \mathrm{Shdiff}_n \right)
\end{aligned}$$

As seen in Equation B.24, all layers in a cell must be solved simultaneously resulting in a linear system of equations to be solved. Like for the momentum equations, a three-banded coefficient matrix is solved. The free formatted Fortran source code listing illustrates the generation and solution of the equation system is given below.

The detailed solution algorithm is given in Figure B.1. The full version of the algorithm as a Fortran program can be made available upon request from the author

```
!Generate right hand sides
do i = 1, NUM_CELLS
    do k = 1, NUM_LAYERS
        SALT_FS(i,k) = SALT(i,k,1)
    end do
end do

do i = 1, NUM_LINKS
    C_BEGIN = BEGIN_CELL_NOS(i)
    C_END   = END_CELL_NOS  (i)

    do k = 1, NUM_LAYERS
        if (U(i,k,2) >= 0.0D0) then
            H_UPSTREAM   = C_BEGIN
            H_DOWNSTREAM = C_END
        else
            H_DOWNSTREAM = C_BEGIN
            H_UPSTREAM   = C_END
        end if

        FLOW_RATE = dabs(U(i,k,2) * H(i,k,2) * B(i,k)) !Horizontal advection

        !Horizontal dispersion
        HS_DISP   = ((K_X * H(i,k,2) * B(i,k)) / LENGTHS(i)) * &
                    (SALT(H_UPSTREAM,k,1) - SALT(H_DOWNSTREAM,k,1))

        if (H_UPSTREAM > 0) SALT_FS(H_UPSTREAM,k) = &
            SALT_FS(H_UPSTREAM,k) - ((DT/VOLS(H_UPSTREAM   , k)) * (FLOW_RATE * &
                  SALT(H_UPSTREAM,k,1) + HS_DISP))

        if (H_DOWNSTREAM > 0) SALT_FS(H_DOWNSTREAM,k) = &
            SALT_FS(H_DOWNSTREAM,k) + ((DT/VOLS(H_DOWNSTREAM, k)) * (FLOW_RATE * &
                  SALT(H_UPSTREAM,k,1) + HS_DISP))
    end do
end do

do i = 1, NUM_CELLS   !Generate the coefficient matrix
    C = 0.0D0
    D = 0.0D0
    E = 0.0D0
    F = 0.0D0
    W(i,1) = 0.0D0
    W(i,NUM_LAYERS+1) = 0.0D0

    do k = 1, NUM_LAYERS !Generate the matrix for salt mass balance equation
        DS_UPPER = 2.0 * K_Z(i,k-1) / (H(i,k-1,2) + H(i,k,2))
        DS_LOWER = 2.0 * K_Z(i,k)   / (H(i,k,2)   + H(i,k+1,2))

        if (W(i,k+1) >= 0.0D0) then
            W_LOWER_UPS = dabs(W(i,k+1))
            W_LOWER_DWS = 0.0D0
        else
            W_LOWER_UPS = 0.0D0
            W_LOWER_DWS = dabs(W(i,k+1))
        end if

        if (W(i,k) >= 0.0D0) then
            W_UPPER_DWS = dabs(W(i,k))
            W_UPPER_UPS = 0.0D0
        else
            W_UPPER_DWS = 0.0D0
            W_UPPER_UPS = dabs(W(i,k))
        end if

        if(k.eq.1) DS_UPPER = 0.0D0
        if(k.eq.NUM_LAYERS) DS_LOWER = 0.0D0
        C(k) = (-DT * (W_UPPER_UPS * A_SURFS(i,k)) + DS_UPPER) / VOLS(i,k)

        D(k) = ((DT * ((W_UPPER_DWS * A_SURFS(i,k))    + &
                       (W_LOWER_DWS * A_SURFS(i,k+1))+ &
                       DS_UPPER + DS_LOWER)) / VOLS(i,k)) + 1.0D0

        E(k) = (-DT * (W_LOWER_UPS * A_SURFS(i,k+1)) + DS_LOWER) / VOLS(i,k)
        F(k) = SALT_FS(i,k)
    end do

    INFO = 0
    call SOLVE_MATRIX_THOMAS(NUM_LAYERS, C, D, E, F, INFO)     !Solve the matrix

    do k = 1, NUM_LAYERS    !Get the solution vector
        SALT(i,k,2) = F(k)
    end do
end do
```

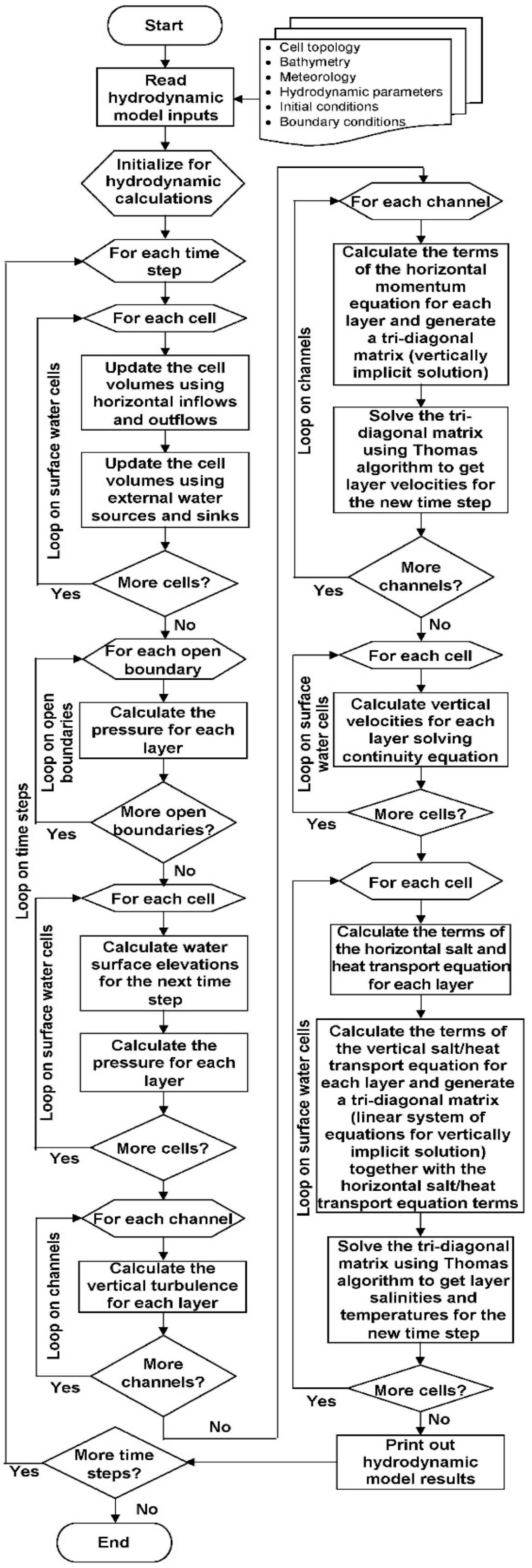

Start
Read hydrodynamic model inputs
Cell topology
Bathymetry
Meteorology
Hydrodynamic parameters
Initial conditions
Boundary conditions
Initialize for hydrodynamic calculations
For each time step
Loop on time steps
For each cell
Loop on surface water cells
Update the cell volumes using horizontal inflows and outflows
Update the cell volumes using external water sources and sinks
More cells?
Yes
No
For each open boundary
Loop on open boundaries
Calculate the pressure for each layer
More open boundaries?
Yes
No
For each cell
Loop on surface water cells
Calculate water surface elevations for the next time step
Calculate the pressure for each layer
More cells?
Yes
For each channel
Loop on channels
Calculate the vertical turbulence for each layer
More channels?
Yes
No
For each channel
Loop on channels
Calculate the terms of the horizontal momentum equation for each layer and generate a tri-diagonal matrix (vertically implicit solution)
Solve the tri-diagonal matrix using Thomas algorithm to get layer velocities for the new time step
More channels?
Yes
No
For each cell
Loop on surface water cells
Calculate vertical velocities for each layer solving continuity equation
More cells?
Yes
For each cell
Loop on surface water cells
Calculate the terms of the horizontal salt and heat transport equation for each layer
Calculate the terms of the vertical salt/heat transport equation for each layer and generate a tri-diagonal matrix (linear system of equations for vertically implicit solution) together with the horizontal salt/heat transport equation terms
Solve the tri-diagonal matrix using Thomas algorithm to get layer salinities and temperatures for the new time step
More cells?
Yes
No
Print out hydrodynamic model results
More time steps?
Yes
No
End

Figure B.1. Solution algorithm for the example hydrodynamic model