SHYFEM
Finite Element Model for Coastal Seas

User Manual

Georg Umgiesser
ISDGM-CNR, S. Polo 1364
30125 Venezia, Italy

Version 4.93

December 2, 2005

Contents

Disclaimer

Comments and additions should be sent to the author:

```
Georg Umgiesser
ISDGM/CNR
S. Polo 1364
30125 Venezia
Italy

Tel.   : ++39-41-5216875
Fax    : ++39-41-2602340
E-Mail : georg@lagoon.isdgm.ve.cnr.it
```

Chapter 1

Introduction

The finite element program `SHYFEM` is a program package that can be used to resolve the hydrodynamic equations in lagoons, coastal seas, estuaries and lakes. The program uses finite elements for the resolution of the hydrodynamic equations. These finite elements, together with an effective semi-implicit time resolution algorithm, makes this program especially suitable for application to a complicated geometry and bathymetry.
This version of the program `SHYFEM` resolves the depth integrated shallow water equations. It is therefore recommended for the application of very shallow basins or well mixed estuaries. Storm surge phenomena can be investigated also. This two-dimensional version of the program is not suited for the application to baroclinic driven flows or large scale flows where the the Coriolis acceleration is important.
Finite elements are superior to finite differences when dealing with complex bathymetric situations and geometries. Finite differences are limited to a regular outlay of their grids. This will be a problem if only parts of a basin need high resolution. The finite element method has an advantage in this case allowing more flexibility with its subdivision of the system in triangles varying in form and size.
This model is especially adapted to run in very shallow basins. It is possible to simulate shallow water flats, i.e., tidal marshes that in a tidal cycle may be covered with water during high tide and then fall dry during ebb tide. This phenomenon is handled by the model in a mass conserving way.
Finite element methods have been introduced into hydrodynamics since 1973 and have been extensively applied to shallow water equations by numerous authors [3, 9, 5, 4, 6].
The model presented here [10, 11] uses the mathematical formulation of the semi-implicit algorithm that decouples the solution of the water levels and velocity components from each other leading to smaller systems to solve. Models of this type have been presented from 1971 on by many authors [7, 2, 1].

Chapter 2

Equations and resolution techniques

2.1 Equations and Boundary Conditions

The equations used in the model are the well known vertically integrated shallow water equations in their formulation with water levels and transports.

$$\frac{\partial U}{\partial t} + gH\frac{\partial \zeta}{\partial x} + RU + X = 0 \tag{2.1}$$

$$\frac{\partial V}{\partial t} + gH\frac{\partial \zeta}{\partial y} + RV + Y = 0 \tag{2.2}$$

$$\frac{\partial \zeta}{\partial t} + \frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} = 0 \tag{2.3}$$

where ζ is the water level, u, v the velocities in x and y direction, U, V the vertical integrated velocities (total or barotropic transports)

$$U = \int_{-h}^{\zeta} u\,dz \qquad V = \int_{-h}^{\zeta} v\,dz$$

g the gravitational acceleration, $H = h + \zeta$ the total water depth, h the undisturbed water depth, t the time and R the friction coefficient. The terms X, Y contain all other terms that may be added to the equations like the wind stress or the nonlinear terms and that need not be treated implicitly in the time discretization. following treatment.
The friction coefficient has been expressed as

$$R = \frac{g\sqrt{u^2 + v^2}}{C^2 H} \tag{2.4}$$

with C the Chezy coefficient. The Chezy term is itself not retained constant but varies with the water depth as

$$C = k_s H^{1/6} \tag{2.5}$$

where k_s is the Strickler coefficient.
In this version of the model the Coriolis term, the turbulent friction term and the nonlinear advective terms have not been implemented.
At open boundaries the water levels are prescribed. At closed boundaries the normal velocity component is set to zero whereas the tangential velocity is a free parameter. This corresponds to a full slip condition.

2.2 The Model

The model uses the semi-implicit time discretization to accomplish the time integration. In the space the finite element method has been used, not in its standard formulation, but using staggered finite elements. In the following a description of the method is given.

2.2.1 Discretization in Time - The Semi-Implicit Method

Looking for an efficient time integration method a semi-implicit scheme has been chosen. The semi-implicit scheme combines the advantages of the explicit and the implicit scheme. It is unconditionally stable for any time step Δt chosen and allows the two momentum equations to be solved explicitly without solving a linear system.
The only equation that has to be solved implicitly is the continuity equation. Compared to a fully implicit solution of the shallow water equations the dimensions of the matrix are reduced to one third. Since the solution of a linear system is roughly proportional to the cube of the dimension of the system the saving in computing time is approximately a factor of 30.
It has to be pointed out that it is important not to be limited with the time step by the CFL criterion for the speed of the external gravity waves

$$\Delta t < \frac{\Delta x}{\sqrt{gH}}$$

where Δx is the minimum distance between the nodes in an element. With the discretization described below in most parts of the lagoon we have $\Delta x \approx 500$m and $H \approx 1$m, so $\Delta t \approx 200$ sec. But the limitation of the time step is determined by the worst case. For example, for $\Delta x = 100$ m and $H = 40$ m the time step criterion would be $\Delta t < 5$ sec, a prohibitive small value.
The equations (1)-(3) are discretized as follows

$$\frac{\zeta^{n+1} - \zeta^n}{\Delta t} + \frac{1}{2}\frac{\partial(U^{n+1} + U^n)}{\partial x} + \frac{1}{2}\frac{\partial(V^{n+1} + V^n)}{\partial y} = 0 \tag{2.6}$$

$$\frac{U^{n+1} - U^n}{\Delta t} + gH\frac{1}{2}\frac{\partial(\zeta^{n+1} + \zeta^n)}{\partial x} + RU^{n+1} + X = 0 \tag{2.7}$$

$$\frac{V^{n+1} - V^n}{\Delta t} + gH\frac{1}{2}\frac{\partial(\zeta^{n+1} + \zeta^n)}{\partial y} + RV^{n+1} + Y = 0 \tag{2.8}$$

With this time discretization the friction term has been formulated fully implicit, X, Y fully explicit and all the other terms have been centered in time. The reason for the implicit treatment of the friction term is to avoid a sign inversion in the term when the friction parameter gets too high. An example of this behavior is given in Backhaus [1].
If the two momentum equations are solved for the unknowns U^{n+1} and V^{n+1} we have

$$U^{n+1} = \frac{1}{1+\Delta t R}\left(U^n - \Delta t g H \frac{1}{2}\frac{\partial(\zeta^{n+1} + \zeta^n)}{\partial x} - \Delta t X\right) \tag{2.9}$$

$$V^{n+1} = \frac{1}{1+\Delta t R}\left(V^n - \Delta t g H \frac{1}{2}\frac{\partial(\zeta^{n+1} + \zeta^n)}{\partial y} - \Delta t Y\right) \tag{2.10}$$

If ζ^{n+1} were known, the solution for U^{n+1} and V^{n+1} could directly be given. To find ζ^{n+1} we insert (2.9) and (2.10) in (2.6). After some transformations (2.6) reads

$$\zeta^{n+1} \quad - \quad (\Delta t/2)^2 \frac{g}{1+\Delta t R}\left(\frac{\partial}{\partial x}(H\frac{\partial \zeta^{n+1}}{\partial x}) + \frac{\partial}{\partial y}(H\frac{\partial \zeta^{n+1}}{\partial y})\right)$$

$$
\begin{aligned}
= \quad & \zeta^n + (\Delta t/2)^2 \frac{g}{1+\Delta t R} \left(\frac{\partial}{\partial x}(H\frac{\partial \zeta^n}{\partial x}) + \frac{\partial}{\partial y}(H\frac{\partial \zeta^n}{\partial y}) \right) \\
- \quad & (\Delta t/2) \left(\frac{2+\Delta t R}{1+\Delta t R} \right) \left(\frac{\partial U^n}{\partial x} + \frac{\partial V^n}{\partial y} \right) \\
+ \quad & \frac{\Delta t^2}{2(1+\Delta t R)} \left(\frac{\partial X}{\partial x} + \frac{\partial Y}{\partial y} \right)
\end{aligned}
\tag{2.11}
$$

The terms on the left hand side contain the unknown ζ^{n+1}, the right hand contains only known values of the old time level. If the spatial derivatives are now expressed by the finite element method a linear system with the unknown ζ^{n+1} is obtained and can be solved by standard methods. Once the solution for ζ^{n+1} is obtained it can be substituted into (2.9) and (2.10) and these two equations can be solved explicitly. In this way all unknowns of the new time step have been found.
Note that the variable H also contains the water level through $H = h + \zeta$. In order to avoid the equations to become nonlinear ζ is evaluated at the old time level so $H = h + \zeta^n$ and H is a known quantity.

2.2.2 Discretization in Space - The Finite Element Method

While the time discretization has been explained above, the discretization in space has still to be carried out. This is done using staggered finite elements. With the semi-implicit method described above it is shown below that using linear triangular elements for all unknowns will not be mass conserving. Furthermore the resulting model will have propagation properties that introduce high numeric damping in the solution of the equations.
For these reasons a quite new approach has been adopted here. The water levels and the velocities (transports) are described by using form functions of different order, being the standard linear form functions for the water levels but stepwise constant form functions for the transports. This will result in a grid that resembles more a staggered grid in finite difference discretizations.

Formalism

Let u be an approximate solution of a linear differential equation L. We expand u with the help of basis functions ϕ_m as

$$u = \phi_m u_m \qquad m = 1, K \tag{2.12}$$

where u_m is the coefficient of the function ϕ_m and K is the order of the approximation. In case of linear finite elements it will just be the number of nodes of the grid used to discretize the domain.
To find the values u_m we try to minimize the residual that arises when u is introduced into L multiplying the equation L by some weighting functions Ψ_n and integrating over the whole domain leading to

$$\int_\Omega \psi_n L(u)\, d\Omega = \int_\Omega \psi_n L(\phi_m u_m)\, d\Omega = u_m \int_\Omega \psi_n L(\phi_m)\, d\Omega \tag{2.13}$$

If the integral is identified with the elements of a matrix a_{nm} we can write (2.13) also as a linear system

$$a_{nm} u_m = 0 \qquad n = 1, K \quad m = 1, K \tag{2.14}$$

Once the basis and weighting functions have been specified the system may be set up and (2.14) may be solved for the unknowns u_m.

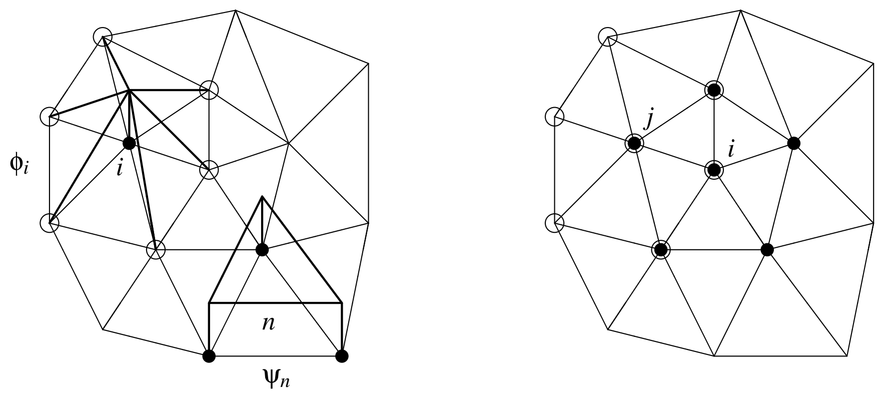

Figure 2.1: a) form functions in domain b) domain of influence of node i

Staggered Finite Elements

For decades finite elements have been used in fluid mechanics in a standardized manner. The form functions ϕ_m were chosen as continuous piecewise linear functions allowing a subdivision of the whole area of interest into small triangular elements specifying the coefficients u_m at the vertices (called nodes) of the triangles. The functions ϕ_m are 1 at node m and 0 at all other nodes and thus different from 0 only in the triangles containing the node m. An example is given in the upper left part of Fig. 1a where the form function for node i is shown. The full circle indicates the node where the function ϕ_i take the value 1 and the hollow circles where they are 0.

The contributions a_{nm} to the system matrix are therefore different from 0 only in elements containing node m and the evaluation of the matrix elements can be performed on an element basis where all coefficients and unknowns are linear functions of x and y.

This approach is straightforward but not very satisfying with the semi-implicit time stepping scheme for reasons explained below. Therefore an other way has been followed in the present formulation. The fluid domain is still divided in triangles and the water levels are still defined at the nodes of the grid and represented by piecewise linear interpolating functions in the internal of each element, i.e.

$$\zeta = \zeta_m \phi_m \qquad m = 1, K$$

However, the transports are now expanded, over each triangle, with piecewise constant (non continuous) form functions ψ_n over the whole domain. We therefore write

$$U = U_n \psi_n \qquad n = 1, J$$

where n is now running over all triangles and J is the total number of triangles. An example of ψ_n is given in the lower right part of Fig. 1a. Note that the form function is constant 1 over the whole element, but outside the element identically 0. Thus it is discontinuous at the element borders.

Since we may identify the center of gravity of the triangle with the point where the transports U_n are defined (contrary to the water levels ζ_m which are defined on the vertices of the triangles), the resulting grid may be seen as a staggered grid where the unknowns are defined on different locations. This kind of grid is usually used with the finite difference method. With the form functions used here the grid of the finite element model resembles very much an Arakawa B-grid that defines the water levels on the center and the velocities on the four vertices of a square.

Staggered finite elements have been first introduced into fluid mechanics by Schoenstadt [8]. He showed that the un-staggered finite element formulation of the shallow water equations has very poor geostrophic adjustment properties. Williams [12, 13] proposed a similar algorithm, the one actually used in this paper, introducing constant form functions for the velocities. He showed the excellent propagation and geostrophic adjustment properties of this scheme.

The Practical Realization

The integration of the partial differential equation is now performed by using the subdivision of the domain in elements (triangles). The water levels ζ are expanded in piecewise linear functions ϕ_m, $m = 1, K$ and the transports are expanded in piecewise constant functions ψ_n, $n = 1, J$ where K and J are the total number of nodes and elements respectively. As weighting functions we use ψ_n for the momentum equations and ϕ_m for the continuity equation. In this way there will be K equations for the unknowns ζ (one for each node) and J equations for the transports (one for each element).
In all cases the consistent mass matrix has been substituted with the their lumped equivalent. This was mainly done to avoid solving a linear system in the case of the momentum equations. But it was of use also in the solution of the continuity equation because the amount of mass relative to one node does not depend on the surrounding nodes. This was important especially for the flood and dry mechanism in order to conserve mass.

Finite Element Equations

If equations (2.9,2.10,2.11) are multiplied with their weighting functions and integrated over an element we can write down the finite element equations. But the solution of the water levels does actually not use the continuity equation in the form (2.11), but a slightly different formulation. Starting from equation (2.6), multiplied by the weighting function Φ_M and integrated over one element yields

$$\int_\Omega \Phi_N(\zeta^{n+1} - \zeta^n)\, d\Omega + (\tfrac{\Delta t}{2}) \int_\Omega \left(\Phi_N \frac{\partial (U^{n+1} + U^n)}{\partial x} + \Phi_N \frac{\partial (V^{n+1} + V^n)}{\partial y} \right) d\Omega = 0$$

If we integrate by parts the last two integrals we obtain

$$\int_\Omega \Phi_N(\zeta^{n+1} - \zeta^n)\, d\Omega - (\tfrac{\Delta t}{2}) \int_\Omega \left(\frac{\partial \Phi_N}{\partial x}(U^{n+1} + U^n) + \frac{\partial \Phi_N}{\partial y}(V^{n+1} + V^n) \right) d\Omega = 0$$

plus two line integrals, not shown, over the boundary of each element that specify the normal flux over the three element sides. In the interior of the domain, once all contributions of all elements have been summed, these terms cancel at every node, leaving only the contribution of the line integral on the boundary of the domain. There, however, the boundary condition to impose is exactly no normal flux over material boundaries. Thus, the contribution of these line integrals is zero.
If now the expressions for U^{n+1}, V^{n+1} are introduced, we obtain a system with again only the water levels as unknowns

$$\begin{aligned}
\int_\Omega \Phi_N \zeta^{n+1}\, d\Omega \quad &+ \quad (\Delta t/2)^2 \alpha g \int_\Omega H\Big(\frac{\partial \Phi_N}{\partial x}\frac{\partial \zeta^{n+1}}{\partial x} + \frac{\partial \Phi_N}{\partial y}\frac{\partial \zeta^{n+1}}{\partial y}\Big)\, d\Omega \\
&= \quad \int_\Omega \Phi_N \zeta^n\, d\Omega + (\Delta t/2)^2 \alpha g \int_\Omega H\Big(\frac{\partial \Phi_N}{\partial x}\frac{\partial \zeta^n}{\partial x} + \frac{\partial \Phi_N}{\partial y}\frac{\partial \zeta^n}{\partial y}\Big)\, d\Omega \\
&+ \quad (\Delta t/2)(1+\alpha) \int_\Omega \Big(\frac{\partial \Phi_N}{\partial x} U^n + \frac{\partial \Phi_N}{\partial y} V^n\Big)\, d\Omega \qquad (2.15) \\
&- \quad (\Delta t^2/2)\alpha \int_\Omega \Big(\frac{\partial \Phi_N}{\partial x} X + \frac{\partial \Phi_N}{\partial y} Y\Big)\, d\Omega
\end{aligned}$$

Here we have introduced the symbol α as a shortcut for

$$\alpha = \frac{1}{1 + \Delta t R}$$

The variables and unknowns may now be expanded with their basis functions and the complete system may be set up.

2.2.3 Mass Conservation

It should be pointed out that only through the use of this staggered grid the semi-implicit time discretization may be implemented in a feasible manner. If the Galerkin method is applied in a naive way to the resulting equation (2.11) (introducing the linear form functions for transports and water levels and setting up the system matrix), the model is not mass conserving. This may be seen in the following way (see Fig. 1b for reference). In the computation of the water level at node i, only ζ and transport values belonging to triangles that contain node i enter the computation (full circles in Fig. 1b). But when, in a second step, the barotropic transports of node j are computed, water levels of nodes that lie further apart from the original node i are used (hollow circles in Fig. 1b). These water levels have not been included in the computation of ζ_i, the water level at node i. So the computed transports are actually different from the transports inserted formally in the continuity equation. The continuity equation is therefore not satisfied.
These contributions of nodes lying further apart could in principle be accounted for. In this case not only the triangles Ω_i around node i but also all the triangles that have nodes in common with the triangles Ω_i would give contributions to node i, namely all nodes and elements shown in Fig. 1b. The result would be an increase of the bandwidth of the matrix for the ζ computation disadvantageous in terms of memory and time requirements.
Using instead the approach of the staggered finite elements, actually only the water levels of elements around node i are needed for the computation of the transports in the triangles Ω_i. In this case the model satisfies the continuity equation and is perfectly mass conserving.

2.2.4 Inter-tidal Flats

Part of a basin may consist of areas that are flooded during high tides and emerge as islands at ebb tide. These inter-tidal flats are quite difficult to handle numerically because the elements that represent these areas are neither islands nor water elements. The boundary line defining their contours is wandering during the evolution of time and a mathematical model must reproduce this features.
For reasons of computer time savings a simplified algorithm has been chosen to represent the inter-tidal flats. When the water level in at least one of the three nodes of an element falls below a minimum value (5 cm) the element is considered an island and is taken out of the system. It will be reintroduced only when in all three nodes the water level is again higher then the minimum value. Because in dry nodes no water level is computed anymore, an estimate of the water level has to be given with some sort of extrapolation mechanism using the water nodes nearby.
This algorithm has the advantage that it is very easy to implement and very fast. The dynamical features close to the inter-tidal flats are of course not well reproduced but the behavior of the method for the rest of the lagoon gave satisfactory results.
In any case, since the method stores the water levels of the last time step, before the element is switched off, introducing the element in a later moment with the same water levels conserves the mass balance. This method showed a much better performance than the one where the new elements were introduced with the water levels taken from the extrapolation of the surrounding nodes.

Chapter 3

Pre-Processing

The pre-processing routine `vp` is used to generate an optimized version of the file that describes the basin where the main program is to be run. In the following a short introduction in using this program is given.

3.1 The pre-processing routine `vp`

The main routine `hp` reads the basin file generated by the pre-processing routine `vp` and uses it as the description of the domain where the hydrodynamic equations have to be solved.
The program `vp` is started by typing `vp` on the command line. From this point on the program is interactive, asking you about the basin file name and other options. Please follow the online instructions.
The routine `vp` reads a file of type GRD. This type of file can be generated and manipulated by the program `grid` which is not described here. In short, the file GRD consists of nodes and elements that describe the geometrical layout of the basin. Moreover, the elements have a type and a depth.
The depth is needed by the main program `hp` to run the model. The type of the element is used by `hp` to determine the friction parameter on the bottom, since this parameter may be assigned differently, depending on the various situations of the bottom roughness.
This file GRD is read by `vp` and transformed into an unformatted file BAS. It is this file that is then read by the main routine `hp`. Therefore, if the name of the basin is `lagoon`, then the file GRD is called `lagoon.grd` and the output of the pre-processing routine `vp` is called `lagoon.bas`.
The program `vp` normally uses the depths assigned to the elements in the file GRD to determine the depth of the finite elements to use in the program `hp`. In the case that these depth values are not complete, and that all nodes have depths assigned in the GRD file, the nodal values of the depths are used and interpolated onto the elements. However, if also these nodal depth values are incomplete or are missing altogether, the program terminates with an error.

3.2 Optimization of the bandwidth

The main task of routine `vp` is the optimization of the internal numbering of the nodes and elements. Re-numbering the elements is just a mere convenience. When assembling the system matrix the contribution of one element after the other has to be added to the system matrix. If the elements are numbered in terms of lowest node numbers, then the access of the nodal pointers is more regular in computer memory and paging is more likely to be inhibited.

However, re-numbering the nodes is absolutely necessary. The system matrix to be solved is of band-matrix type. I.e., non-zero entries are all concentrated along the main diagonal in a more or less narrow band. The larger this band is, the larger the amount of cpu time spent to solve the system. The time to solve a band matrix is of order $n \cdot m^2$, where n is the size of the matrix and m is the bandwidth. Note that m is normally much smaller than n.
If the nodes are left with the original numbering, it is very likely that the bandwidth is very high, unless the nodes in the file GRD are by chance already optimized. Since the bandwidth m is entering the above formula quadratically, the amount of time spent solving the matrix will be prohibitive. E.g., halving the bandwidth will speed up computations by a factor of 4.
The bandwidth is equal to the maximum difference of node numbers in one element. It is therefore important to re-number the nodes in order to minimize this number. However, there exist only heuristic algorithms for the minimization of this number.
The two main algorithms used in the routine `vp` are the Cuthill McGee algorithm and the algorithm of Rosen. The first one, starting from one node, tries to number all neighbors in a greedy way, optimizing only this step. From the points numbered in this step, the next neighbors are numbered.
This procedure is tried from more than one node, possibly from all boundary nodes. The numbering resulting from this algorithm is normally very good and needs only slight enhancements to be optimum.
Once all nodes are numbered, the Rosen algorithm tries to exchange these node numbers, where the highest difference can be found. This normally gives only a slight improvement of the bandwidth. It has been seen, however, that, if the node numbers coming out from the Cuthill McGee algorithm are reversed, before feeding them into the Rosen algorithm, the results tend to be slightly better. This step is also performed by the program.
All these steps are performed by the program without intervention by the operator, if the automatic optimization of bandwidth is chosen in the program `vp`. The choices are to not perform the bandwidth optimization at all (GRD file has already optimized node numbering), perform it automatically or perform it manually. It is suggested to always perform automatic optimization of the bandwidth. This choice will lead to a nearly optimum numbering of the nodes and will be by all means good results.
If, however, you decide to do a manual optimization, please follow the online instructions in the program.

3.3 Internal and external node numbering

As explained above, the elements and nodes of the basin are re-numbered in order to optimize the bandwidth of the system matrix and so the execution speed of the program.
However, this re-numbering of the node and elements is transparent to the user. The program keeps pointers from the original numbering (external numbers) to the optimized numbering (internal numbers). The user has to deal only with external numbers, even if the program uses internally the other number system.
Moreover, the internal numbers are generated consecutively. Therefore, if there are a total of 4000 nodes in the system, the internal nodes run from 1 to 4000. The external node numbers, on the other side, can be anything the user likes. They just must be unique. This allows for insertion and deletion of nodes without having to re-number over and over again the basin.
The nodes that have to be specified in the input parameter file use again external numbers. In this way, changing the structure of the basin does not at all change the node and element numbers in the input parameter file. Except in the case, where modifications actually touch nodes and elements that are specified in the parameter file.

Chapter 4

The Model

In the following an overview is given on running the model `SHYFEM`. The model needs a parameter input file that is read on standard input. Moreover, it needs some external files that are specified in this parameter input file. The model produces several external files with the results of the simulation. Again, the name of this files can be influenced by the parameter input file

4.1 The Parameter Input File

The model reads one input file that determines the behavior of the simulation. All possible parameters can and must be set in this file. If other data files are to be read, here is the place where to specify them.
The model reads this parameter file from standard input. Thus, if the model binary is called `hp` and the parameter file `param.str`, then the following line starts the simulation

```
hp < param.str
```

and runs the model.

4.1.1 The General Structure of the Parameter Input File

The input parameter file is the file that guides program performance. It contains all necessary information for the main routine to execute the model. Nearly all parameters that can be given have a default value which is used when the parameter is not listed in the file. Only some time parameters are compulsory and must be present in the file.
The format of the file looks very like a namelist format, but is not dependent on the compiler used. Values of parameters are given in the form : `name = value` or `name = 'text'`. If `name` is an array the following format is used :

```
          name = value1 , value2, ... valueN
```

The list can continue on the following lines. Blanks before and after the equal sign are ignored. More then one parameter can be present on one line. As separator blank, tab and comma can be used.
Parameters, arrays and data must be given in between certain sections. A section starts with the character `$` followed by a keyword and ends with `$end`. The `$keyword` and `$end` must not contain any blank characters and must be the first non blank characters in the line. Other characters following the keyword on the same line separated by a valid separator are ignored.
Several sections of data may be present in the input parameter file. Further ahead all sections are presented and the possible parameters that can be specified are explained. The

sequence in which the sections appear is of no importance. However, the first section must always be section `\$title`, the section that determines the name of simulation and the basin file to use and gives a one line description of the simulation.
Lines outside of the sections are ignored. This gives the possibility to comment the parameter input file.
Figure 4.1 shows an example of a typical input parameter file and the use of the sections and definition of parameters.

4.1.2 The Single Sections of the Parameter Input File

Section `$title`

This section must be always the first section in the parameter input file. It contains only three lines. An example is given in figure 4.2.
The first line of this section is a free one line description of the simulation that is to be carried out. The next line contains the name of the simulation (in this case `name_of_simulation`). All created files will use this name in the main part of the file name with different extensions. Therefore the hydrodynamic output file (extension `out`) will be named `name_of_simulation.out`. The last line gives the name of the basin file to be used. This is the pre-processed file of the basin with extension `bas`. In our example the basin file `name_of_basin.bas` is used.
The directory where this files are read from or written to depends on the settings in section `$name`. Using the default the program will read from and write to the current directory.

Section `$para`

This section defines the general behavior of the simulation, gives various constants of parameters and determines what output files are written. In the following the meaning of all possible parameters is given.
Note that the only compulsory parameters in this section are the ones that chose the duration of the simulation and the integration time step. All other parameters are optional.

Compulsory time parameters This parameters are compulsory parameters that define the period of the simulation. They must be present in all cases.

`itanf`	Start of simulation. (Default 0)
`itend`	End of simulation.
`idt`	Time step of integration.

Output parameters The following parameters deal with the output frequency and start time to external files. The content of the various output files should be looked up in the appropriate section.
The default for the time step of output of the files is 0 which means that no output file is written. If the time step of the output files is equal to the time step of the simulation then at every time step the output file is written. The default start time of the output is 0.

`idtout` `itmout`	Time step and start time for writing to file OUT, the file containing the general hydrodynamic results.
`idtext` `itmext`	Time step and start time for writing to file EXT, the file containing hydrodynamic data of extra points. The extra points for which the data is written to this file are given in section `extra` of the parameter file.
`idtrst` `itmrst`	Time step and start time for writing the restart

```
$title
        benchmark test for test lagoon
        bench
        venlag
$end

$para
        itanf = 0  itend = 86400  idt = 300
        ireib = 2  ilin = 0
        href = 0.23  iczv = 1
$end

$bound1
        kbound = 73 74 76
        ampli = 0.50 period = 43200. phase = 10800. zref = 0.
$end

$bound2
        kbound = 353,350, 349
        iqual = 1
$end

$bound3
        kbound = 1374 1154 1160 1161
        iqual = 1
$end

$name
        wind='win18sep.win'
$end

-------------- MAREOGRAFI PER TARATURA--------------
13,133,99,259,328,772,419,1141,1195,1070,1064,942,468,1154
-----------------------------------------------------

$extra ------- MAREOGRAFI + SEZIONI PER TARATURA----------
13,133,99,259,328,772,419,1141,1195,1070,1064,942,468,1154
73,74,76,353,350,349,1374,1154,1160,1161,408,409,786,795
$end

$area ------- old chezy values ----------
         0  33.
         1  25.  27.    74    75
         2  21.  23.   350   346
         3  20.  25.  1154  1153
         4  27.
         5  27.
         6  27.
$end
```

Figure 4.1: Example of a parameter input file

```
$title
        free one line description of simulation
        name_of_simulation
        name_of_basin
$end
```

Figure 4.2: Example of section `$title`

file (extension RST). No restart file is written with `idtrst` equal to 0. `itrst` Time to use for the restart. If a restart is performed, then the file name containing the restart data has to be specified in `restrt` and the time record corresponding to `itrst` is used in this file.

`idtres` `itmres`	Time step and start time for writing to file RES, the file containing residual hydrodynamic data.
`idtrms` `itmrms`	Time step and start time for writing to file RMS, the file containing hydrodynamic data of root mean square velocities.
`idtflx` `itmflx`	Time step and start time for writing to file FLX, the file containing discharge data through defined sections. The transects for which the discharges are computed are given in section `flux` of the parameter file.

Model parameters The next parameters define the inclusion or exclusion of certain terms of the primitive equations.

`ilin`	Linearization of the momentum equations. If `ilin` is different from 0 the advective terms are not included in the computation. (Default 1)
`itlin`	This parameter decides how the advective (non-linear) terms are computed. The value of 0 (default) uses the usual finite element discretization over a single element. The value of 1 choses a semi-lagrangian approach that is theoretically stable also for Courant numbers higher than 1. It is however recommended that the time step is limited using `itsplt` and `coumax` described below. (Default 0)
`iclin`	Linearization of the continuity equation. If `iclin` is different from 0 the depth term in the continuity equation is taken to be constant. (Default 0)
`nrand`	Compute system matrix every `nrand` iterations. This parameter can be used to speed up computations if the system matrix does not depend crucially on the varying water depth (e.g., in deep waters). It is recommended to leave it to the default value of 1 (compute at every iteration). (Default 1)

The next parameters allow for a variable time step in the hydrodynamic computations. This is especially important for the non-linear model (`ilin=0`) because in this case the criterion for stability cannot be determined a priori and in any case the time integration will not be unconditionally stable.
The variable time steps allows for longer basic time steps (here called macro time steps) which have to be set in `idt`. It then computes the optimal time step (here micro time step) in order to not exceed the given Courant number. However, the value for the macro time step will never be exceeded.

`itsplt`	Type of variable time step computation. If this value is 0, the time step will kept constant at its initial value. A value of 1 devides the initial time step into (possibly) equal parts, but makes sure that at the end of the micro time steps one complete macro time step has been executed. The last mode `itsplt` = 2 does not care about the macro time step, but always uses the biggest time step possible. In this case it is not assured that after some micro time steps a macro time step will be recovered. Please note that the initial macro time step will never be exceeded. (Default 0)
`coumax`	Normally the time step is computed in order to not exceed the Courant number of 1. However, in some cases the non-linear terms are stable even for a value higher than 1 or there is a need to achieve a lower Courant number. Setting `coumax` to the desired Courant number achieves exactly this effect. (Default 1)
`idtsyn`	In case of `itsplt` = 2 this parameter makes sure that after a time of `idtsyn` the time step will be syncronized to this time. Therefore, setting `idtsyn` = 3600 means that there will be a time stamp every hour, even if the model has to take one very small time step in order to reach that time. This parameter is useful only for `itsplt` = 2 and its default value of 0 does not make any syncronization.

These parameters define the weighting of time steps in the semi-implicit algorithm. With these parameters the damping of gravity waves can be controlled. Only modify them if you know what you are doing.

`azpar`	Weighting of the new time level of the transport terms in the continuity equation. (Default 0.5)
`ampar`	Weighting of the new time level of the pressure term in the momentum equations. (Default 0.5)

Coriolis parameters The next parameters define the parameters to be used in the Coriolis terms.

`icor`	If this parameter is 0, the Coriolis terms are not included in the computation. A value of 1 uses a beta-plane approximation with a variable Coriolis parameter f, whereas a value of 2 uses an f-plane approximation where the Coriolis parameter f is kept constant over the whole domain. (Default 0)
`dlat`	Average latitude of the basin. This is used to compute the Coriolis parameter f. If not given the latitude in the basin file is used. If given the value of `dlat` in the input parameter file effectively substitues the value given in the basin file.

Depth parameters The next parameters deal with various depth values of the basin.

`href`	Reference depth. If the depth values of the basin and the water levels are referred to mean sea level, `href` should be 0 (default value). Else this value is subtracted from the given depth values. For example, if `href = 0.20` then a depth value in the basin of 1 meter will be reduced to 80 centimeters.

`hzmin`	Minimum total water depth that will remain in a node if the element becomes dry. (Default 0.01 m)
`hzoff`	Total water depth at which an element will be taken out of the computation because it becomes dry. (Default 0.05 m)
`hzon`	Total water depth at which a dry element will be re-inserted into the computation. (Default 0.10 m)
`hmin`	Minimum water depth (most shallow) for the whole basin. All depth values of the basin will be adjusted so that no water depth is shallower than `hmin`. (Default is no adjustment)
`hmax`	Maximum water depth (deepest) for the whole basin. All depth values of the basin will be adjusted so that no water depth is deeper than `hmax`. (Default is no adjustment)

Bottom friction The friction term in the momentum equations can be written as Ru and Rv where R is the variable friction coefficient and u,v are the velocities in x,y direction respectively. The form of R can be specified in various ways. The value of `ireib` is choosing between the formulations. In the parameter input file a value λ is specified that is used in the formulas below.

`ireib` Type of friction used (default 0):

0 No friction used

1 $R = \lambda$ is constant

2 λ is the Strickler coefficient. In this formulation R is written as $R = \frac{g}{C^2}\frac{|u|}{H}$ with $C = k_s H^{1/6}$ and $\lambda = k_s$ is the Strickler coefficient. In the above formula g is the gravitational acceleration, $|u|$ the modulus of the current velocity and H the total water depth.

3 λ is the Chezy coefficient. In this formulation R is written as $R = \frac{g}{C^2}\frac{|u|}{H}$ and $\lambda = C$ is the Chezy coefficient.

4 $R = \lambda/H$ with H the total water depth

5 $R = \lambda\frac{|u|}{H}$

`czdef`	The default value for the friction parameter λ. Depending on the value of `ireib` the coefficient λ is describing linear friction or a Chezy or Strickler form of friction (default 0).
`iczv`	Normally R is evaluated at every time step (`iczv = 1`). If for some reason this behavior is not desirable, `iczv = 0` evaluates the value of R only before the first time step, keeping it constant for the rest of the simulation. (default 1)

The value of λ may be specified for the whole basin through the value of `czdef`. For more control over the friction parameter it can be also specified in section `area` where the friction parameter depending on the type of the element may be varied. Please see the paragraph on section `area` for more information.

Physical parameters The next parameters describe physical values that can be adjusted if needed.

`rowass`	Average density of sea water. (Default 1025 $\mathrm{kg\,m^{-3}}$)
`roluft`	Average density of air. (Default 1.225 $\mathrm{kg\,m^{-3}}$)
`grav`	Average gravitational acceleration. (Default 9.81 $\mathrm{m\,s^{-2}}$)

Wind parameters The next two parameters deal with the wind stress to be prescribed at the surface of the basin.
The wind data can either be specified in an external file (ASCII or binary) or directly in the parameter file in section `wind`. The ASCII file or the wind section contain three columns, the first giving the time in seconds, and the others the components of the wind speed. Please see below how the last two columns are interpreted depending on the value of `iwtype`. For the format of the binary file please see the relative section. If both a wind file and section `wind` are given, data from the file is used.
The wind stress is normally computed with the following formula

$$\tau^x = \rho_a c_D |u| u^x \quad \tau^y = \rho_a c_D |u| u^y \tag{4.1}$$

where ρ_a, ρ_0 is the density of air and water respectively, u the modulus of wind speed and u^x, u^y the components of wind speed in x, y direction. In this formulation c_D is a dimensionless drag coefficient that varies between $1.5 \cdot 10^{-3}$ and $3.2 \cdot 10^{-3}$. The wind speed is normally the wind speed measured at a height of 10 m.

`iwtype` The type of wind data given (default 1):

0 No wind data is processed

1 The components of the wind is given in [m/s]

2 The stress (τ^x, τ^y) is directly specified

3 The wind is given in speed [m/s] and direction [degrees]. A direction of 0^o specifies a wind from the north, 90^o a wind from the east etc.

4 As in 3 but the speed is given in knots

`dragco` Drag coefficient used in the above formula. The default value is 0 so it must be specified. Please note also that in case of `iwtype` = 2 this value is of no interest, since the stress is specified directly.

Concentrations The next parameters deal with the transport and diffusion of a conservative substance. The substance is dissolved in the water and acts like a tracer.

`iconz`	Flag if the computation on the tracer is done. A value different from 0 computes the transport and diffusion of the substance. (Default 0)
`conref`	Reference (ambient) concentration of the tracer in any unit. (Default 0)
`itvd`	Type of advection scheme used for the transport and diffusion equation. Normally an upwind scheme is used (0), but setting the parameter `itvd` to 1 choses a TVD scheme. This feature is still experimental, so use with care. (Default 0)

`dhpar`	Horizontal diffusion parameter (general). (Default 0)
`chpar`	Horizontal diffusion parameter for the tracer. (Default 0)
`diftur`	Vertical turbulent diffusion parameter for the tracer. (Default 0)
`difmol`	Vertical molecular diffusion parameter for the tracer. (Default 1.0e-06)

Temperature and salinity The next parameters deal with the transport and diffusion of temperature and salinity.

`itemp`	Flag if the computation on the temperature is done. A value different from 0 computes the transport and diffusion of the temperature. (Default 0)
`isalt`	Flag if the computation on the salinity is done. A value different from 0 computes the transport and diffusion of the salinity. (Default 0)
`temref`	Reference (ambient) temperature of the water in centigrade. (Default 0)
`salref`	Reference (ambient) salinity of the water in psu (practical salinity units, per mille). (Default 0)
`thpar`	Horizontal diffusion parameter for temperature. (Default 0)
`shpar`	Horizontal diffusion parameter for salinity. (Default 0)

Output for concentration, temperature and salinity The next parameters define the output frequency of the computed concentration (temperature, salinity) to file. They also define the internal time step to be used with the time integration.

`idtcon` `itmcon`	Time step and start time for writing to file CON (concentration), TEM (temperature) and SAL (salinity).
`istot`	Frequency of internal time step of the solution of the transport and diffusion equation. Normally at every (external) time step of the hydrodynamic equations one transport-diffusion (internal) time step is executed. If the external time step is too long, the solution of the transport-diffusion equations with the same time step may lead to instabilities. These instabilities can be avoided if more internal time steps are executed in one external step. `istot` gives the number of internal time steps to be executed in one external step. (Default 1)

Section `$name`

In this sections names of directories or input files can be given. All directories default to the current directory, whereas all file names are empty, i.e., no input files are given.

Directory specification This parameters define directories for various input and output files.

`basdir`	Directory where basin file BAS resides. (Default .)
`datdir`	Directory where output files are written. (Default .)

`tmpdir`	Directory for temporary files. (Default .)
`defdir`	Default directory for other files. (Default .)

File names The following strings enable the specification of files that account for initial conditions or forcing.

`bound`	File with initial water level distribution. This file must be constructed by the utility routine `zinit`.
`wind`	File with wind data. The file may be either formatted or unformatted. For the format of the unformatted file please see the section where the WIN file is discussed. The format of formatted ASCII file is in standard time-series format, with the first column containing the time in seconds and the next two columns containing the wind data. The meaning of the two values depend on the value of the parameter `iwtype` in the `para` section.
`rain`	File with rain data. This file is a standard time series with the time in seconds and the rain values in mm/day.
`qflux`	File with heat flux data. This file must be in a special format to account for the various parameters that are needed by the heat flux module to run. Please refer to the information on the file `qflux`.

`restrt` Name of the file if a restart is to be performed. The file has to be produced by a previous run with the parameter `idtrst` different from 0. The data records used in the file for the restart must be given by time `itrst`.
Lagrangian trajectories !LAGR

Section `$bound`

These parameters determine the open boundary nodes and the type of the boundary: level or flux boundary. At the first the water levels are imposed, on the second the fluxes are prescribed.
There may be multiple sections `bound` in one parameter input file, describing all open boundary conditions necessary. Every section must therefore be supplied with a boundary number. The numbering of the open boundaries must be increasing. The number of the boundary must be specified directly after the keyword `bound`, such as `bound1` or `bound 1`.

`kbound`	Array containing the node numbers that are part of the open boundary. The node numbers must form one contiguous line with the domain (elements) to the left. This corresponds to an anti-clockwise sense. At least two nodes must be given.

`ibtyp`	Type of open boundary. **0** No boundary values specified **1** Level boundary. At this open boundary the water level is imposed and the prescribed values are interpreted as water levels in meters. **2** Flux boundary. Here the discharge in $m^3\ s^{-1}$has to be prescribed. **3** Internal flux boundary. As with `ibtyp = 2` a discharge has to be imposed, but the node where discharge is imposed can be an internal node and need not be on the outer boundary of the domain. For every node in `kbound` the volume rate specified will be added to the existing water volume. This behavior is different from the `ibtyp = 2` where the whole boundary received the discharge specified. **4** Momentum input. The node or nodes may be internal. This feature can be used to describe local acceleration of the water column. The unit is force / density [$m^4\ s^{-2}$]. In other words it is the rate of volume [$m^3\ s^{-1}$] times the velocity [m/s] to which the water is accelerated.
`iqual`	If the boundary conditions for this open boundary are equal to the ones of boundary `i`, then setting `iqual = i` copies all the values of boundary `i` to the actual boundary. Note that the value of `iqual` must be smaller than the number of the actual boundary, i.e., boundary `i` must have been defined before.

The next parameters give a possibility to specify the file name of the various input files that are to be read by the model. Values for the boundary condition can be given at any time step. The model interpolates in between given time steps if needed. The grade of interpolation can be given by `intpol`.
All files are in ASCII and share a common format. The file must contain two columns, the first giving the time of simulation in seconds that refers to the value given in the second column. The value in the second column must be in the unit of the variable that is given. The time values must be in increasing order. There must be values for the whole simulation, i.e., the time value of the first line must be smaller or equal than the start of the simulation, and the time value of the last line must be greater or equal than the end of the simulation.

`boundn`	File name that contains values for the boundary condition. The value of the variable given in the second column must be in the unit determined by `ibtyp`, i.e., in meters for a level boundary, in $m^3\ s^{-1}$for a flux boundary and in $m^4\ s^{-2}$for a momentum input.
`zfact`	Factor with which the values from `boundn` are multiplied to form the final value of the boundary condition. E.g., this value can be used to set up a quick sensitivity run by multiplying all discharges by a factor without generating a new file. (Default 1)

`conzn`
`tempn`
`saltn`
	File name that contains values for the respective boundary condition, i.e., for concentration, temperature and salinity. The format is the same as for file `boundn`. The unit of the values given in the second column must the ones of the variable, i.e., arbitrary unit for concentration, centigrade for temperature and psu (per mille) for salinity.

`intpol`	Order of interpolation for the boundary values read through files. Use for 1 for stepwise (no) interpolation, 2 for linear interpolation. The default is cubic interpolation (4)

The next parameters can be used to impose a sinusoidal water level (tide) or flux at the open boundary. These values are used if no boundary file `boundn` has been given. The values must be in the unit of the intended variable determined by `ibtyp`.

`ampli`	Amplitude of the sinus function imposed. (Default 0)
`period`	Period of the sinus function. (Default 43200, 12 hours)
`phase`	Phase shift of the sinus function imposed. A positive value of one quarter of the period reproduces a cosine function. (Default 0)
`zref`	Reference level of the sinus function imposed. If only `zref` is specified (`ampli = 0`) a constant value of `zref` is imposed on the open boundary.

With the next parameters a constant value can be imposed for the variables of concentration, temperature and salinity. In this case no file with boundary values has to be supplied. The default for all values is 0, i.e., if no file with boundary values is supplied and no constant is set the value of 0 is imposed on the open boundary.

`conz` `temp` `salt`	Constant boundary values for concentration, temperature and salinity respectively. If these values are set no boundary file has to be supplied. (Default 0)

The next two values are used for constant momentum input. This feature can be used to describe local acceleration of the water column. The values give the input of momentum in x and y direction. The unit is force / density (m^4 s^{-2}). In other words it is the rate of volume (m^3 s^{-1}) times the velocity (m/s) to which the water is accelerated.
These values are used if boundary condition `ibtyp = 4` has been chosen and no boundary input file has been given. If the momentum input is varying then it may be specified with the file `boundn`. In this case the file `boundn` must contain three columns, the first for the time, and the other two for the momentum input in x, y direction.

`umom` `vmom`	Constant values for momentum input. (Default 0)

Section `$wind`

In this section the wind data can be given directly without the creation of an external file. Note, however, that a wind file specified in the `name` section takes precedence over this section. E.g., if both a section `wind` and a wind file in `name` is given, the wind data from the file is used.
The format of the wind data in this section is the same as the format in the ASCII wind file, i.e., three columns, with the first specifying the time in seconds and the other two columns giving the wind data. The interpretation of the wind data depends on the value of `iwtype`. For more information please see the description of `iwtype` in section `para`.

Section `$extra`

In this section the node numbers of so called "extra" points are given. These are points where water level and velocities are written to create a time series that can be elaborated later. The output for these "extra" points consumes little memory and can be therefore written with a much higher frequency (typically the same as the integration time step) than the complete hydrodynamical output. The output is written to file EXT.

The node numbers are specified in a free format on one ore more lines. An example can be seen in figure 4.1. No keywords are expected in this section.

Section `$flux`

In this section transects are specified through which the discharge of water is computed by the program and written to file FLX. The transects are defined by their nodes through which they run. All nodes in one transect must be adjacent, i.e., they must form a continuous line in the FEM network.
The nodes of the transects are specified in free format. Between two transects one or more 0's must be inserted. An example is given in figure 4.3.

```
$flux
1001 1002 1004 0
35 37 46 0 0 56 57 58 0
407
301
435 0 89 87
$end
```

Figure 4.3: Example of section `$flux`

The example shows the definition of 5 transects. As can be seen, the nodes of the transects can be given on one line alone (first transect), two transects on one line (transect 2 and 3), spread over more lines (transect 4) and a last transect.

Chapter 5

Post-Processing

There are several routines that do a post-processing of the results of the main routine. The most important are described in this chapter. Note that in the model framework no program is supplied to do time series plots. However, there are utility routines that will extract data from the output files. These data will be written in a way that it can be imported into a spreadsheet or any other plotting program that does the nice plotting.

5.1 Plotting of maps with `plotmap`

5.1.1 The parameter input file for `plotmap`

The format of the parameter input file is the same as the one for the main routine. Please see this section for more information on the format of the parameter input file.
Some sections of the parameter input file are identical to the sections used in the main routine. For easier reference we will repeat the possible parameters of these section here.

Section `$title`

This section must be always the first section in the parameter input file. It contains only three lines. An example has been given already in figure 4.2.
The only difference respect to the `$para` section of the main routine is the first line. Here any description of the output can be used. It is just a way to label the parameter file. The other two line with the name of simulation and the basin are used to open the files needed for plotting.

Section `$para`

These parameters set generic values for the plot.
Note that the only compulsory parameter in this section is `iwhat`, a parameter that determines what to plot. All other parameters are optional.

`iwhat`	Flag that determines what to plot (default 0): **0** Nothing plotted **1** Plot basin (grid and isolines of depth) **2** Plot velocities **3** Plot transports **4** Plot water levels **5** Plot concentration **6** Plot temperature **7** Plot salinity **8** Plot rms-velocity
`x0` `y0`	Lower left corner of the plotting area. (Default is whole area)
`x1` `y1`	Upper right corner of the plotting area. (Default is whole area)
`x0leg` `y0leg`	Lower left corner of the area where the legend is plotted.
`x1leg` `y1leg`	Upper right corner of the area. where the legend is plotted.
`dxygrd`	Grid size if the results are interpolated on a regular grid. A value of 0 does not use a regular grid but the original finite element grid for plotting. (Default 0)
`velfac` `trafac`	Not used anymore.
`typls`	Typical length scale to be used when scaling velocity or transport arrows. If `dxygrd` is given this length is used and `typls` is not used. If not given it is computed from the basin parameters. (Default 0)
`typlsf`	Additional factor to be used with typls to determine the length of the maximum or reference vector. This is the easiest way to scale the velocitiy arrows with an overall factor. (Default 1)
`velref` `traref`	Reference value to be used when scaling arrows. If given, a vector with this value will have a length of `typls*typlsf` on the map, or, in case `dxygrd` is given, `dxygrd*typlsf`. If not set the maximum value of the velocity/transport will be used as `velref`,`traref`. (Default 0)
`velmin` `tramin`	Minimum value for which an arrow will be plotted. With this value you can eliminate small arrows in low dynamic areas. (Default 0)
`bndlin`	Name of file that gives the boundary line that is not part of the finite element domain. The file must be in BND format. You can use the program grd2bnd.pl to create the file from a GRD file. (Default is no file)

`ioverl`	Create overlay of velocity vectors on scalar value. With the value of 0 no overlay is created, 1 creates an overlay with the velocity speed. The value of 2 overlays vertical velocities 3 water levels and 4 overlays bathymetry.(Default 0)
`inorm`	Normally the horizontal velocities are plotted in scale. The value of `inorm` can change this behavior. A value of 1 normalizes velocity vectors (all vectors are the same length), whereas 2 scales from a given minimum velocity `velmin`. Finally, the value of 3 uses a logarithmic scale. (Default 0)

Section `$color`

The next parameters deal with the definition of the colors to be used in the plot. A color bar is plotted too.

`icolor`	Flag that determines the type of color table to be used. 0 stands for gray scale, 1 for HSB color table. (Default 0)
`isoval`	Array that defines the values for the isolines and colors that are to be plotted. Values given must be in the unit of the variable that will be plotted, i.e., meters for water levels etc.
`color`	Array that gives the color indices for the plotting color to be used. Ranges are from 0 to 1. The type of the color depends on the variable `icolor`. For the gray scale table 0 represents black and 1 white. Values in between correspond to tones of gray. For the HSB color table going from 0 to 1 gives the color of the rainbow. There must be one more value in `color` than in `isoval`. The first color in `color` refers to values less than `isoval(1)`, the second color in `color` to values between `isoval(1)` and `isoval(2)`. The last color in `color` refers to values greater than the last value in `isoval`.
`x0col` `y0col`	Lower left corner of the area where the color bar is plotted.
`x1col` `y1col`	Upper right corner of the area where the color bar is plotted.
`dval`	Difference of values between isolines. If this value is greater then 0 the values for isolines and the respective colors are created automatically without need to specify the single values in arrays `isoval` and `color`. (Default 0)
`faccol`	Factor for the values that are written to the color bar legend. This enables you, e.g., to give water level results in mm (`faccol = 1000`). (Default 1)
`ndccol`	Decimals after the decimal point for the values written to the color bar legend. Use the value `-1` to not write the decimal point. (Default -1)
`legcol`	Text for the description of the color bar. This text is written above the color bar.

Section `$arrow`

The next parameters deal with the reference arrow that is plotted in a legend. The arrow regards the plots where the velocity or the transport is plotted.

`x0arr` `y0arr`	Lower left corner of the area where the reference arrow is plotted.
`x1arr` `y1arr`	Upper right corner of the area where the reference arrow is plotted.
`facvel` `factra`	Factor for the value that are written to the arrow legend for the velocity and transport. This enables you, e.g., to give velocities in mm/s (`facvel = 1000`). (Default 1)
`ndcvel` `ndctra`	Decimals after the decimal point for the values written to the arrow legend (velocity and transport). Use the value `-1` to not write the decimal point. (Default 2)
`legvel` `legtra`	Text for the description of the arrow legend (velocity and transport). This text is written above the arrow legend.
`arrvel` `arrtra`	Length of arrow in legend (in velocity or transport units). If not given the arrow length will be computed automatically. (Default 0)

Section `$legend`

In this section annotations in the plots can be given. The section consists of a series of lines that must contain the following information:
The first two values (x,y) give the position of the annotation. The third value gives the font size in points for the text to be written and the last entry on the line is the text for the annotation. A small example of an annotation would be:

```
$legend
-3000. -2000. 12   'Adriatic Sea'
3200.  2500. 12   'Lido Inlet'
$end
```

where the two annotations are written with a font size of 12 points. The text `Adriatic Sea` is written at (-3000,-2000), so that the lower left corner of the `A` of Adriatic is at the specified position.

Section `$name`

Directory specification This parameters define directories for various input and output files.

`basdir`	Directory where basin file BAS resides. (Default .)
`datdir`	Directory where output files are written. (Default .)
`tmpdir`	Directory for temporary files. (Default .)
`defdir`	Default directory for other files. (Default .)

Chapter 6

The Water Quality Module

by Donata Melaku Canu, Georg Umgiesser, Cosimo Solidoro

The coupling between EUTRO and FEM constitute a structure which is meant to be a generic water quality for full eutrophication dynamics. The Water Quality model is described fully in Umgiesser et al. (2003).

6.1 General Description

The water quality model has been derived from the EUTRO module of WASP (released by the U.S. Environmental Protection Agency (EPA) (Ambrose et al., 1993) and modified. It simulates the evolution of nine state variables in the water column and sediment bed, including dissolved oxygen (DO), carbonaceous biochemical oxygen demand (CBOD), phytoplankton carbon and chlorophyll a (PHY), ammonia (NH3), nitrate (NOX), organic nitrogen (ON), organic phosphorus (OP), orthophosphate (OPO4) and zooplankton (ZOO). The interacting nine state variables can be considered as four interacting systems: the carbon cycle, the phosphorous cycle, the nitrogen cycle and the dissolved oxygen balance (Fig. ??). Different levels of complexity can be selected by switching the eight variables on and off, in order to address the specific topics.
The evolution of phytoplankton concentration (Reaction 1, Table 6.1) is described by the anabolic and the catabolic terms, plus a grazing term related to zooplankton concentration (Reaction 10, 11 and 12, Table 6.2), which however is treated as a constant in the original version. The anabolic term (Reaction 10, Table 6.2) is related to light intensity, temperature and concentration of nutrients in water, while the catabolic term (Reaction 11, Table 6.2) depends on temperature.
Phytoplankton growth is described by combining a maximum growth rate under optimal conditions, and a number of dimensionless factors, each ranging from 0 to 1, and each one referring to a specific environmental factor (nutrient, light availability), which reduces the phytoplanktonic growth insofar as environmental conditions are at sub-optimal levels. Phytoplankton stochiometry is fixed at the user-specified ratio, so that no luxury uptake mechanisms are considered, and the uptake of nutrients is directly linked to the phytoplankton growth, and described by the same one-step kinetic law. More specifically, the influence of inorganic phosphorous and nitrogen availability on phytoplankton growth/nutrients uptake is simulated by means of Michealis-Menten-Monod kinetics (Reactions 42 and 43, Table 6.2). Phytoplankton uptakes nitrogen both in the forms of ammonia and nitrate, but ammonia is assimilated preferentially, as indicated in the ammonia preference relation (Reaction 38, Table 6.2). The influence of temperature is given by an exponential relation (Reaction 13, Table 6.2), while the functional forms for the limitation due to sub-optimal light con-

dition can be chosen between three alternative options, namely the formulation proposed by Di Toro et al. (1971) and the one proposed by Smith (1980) (Di Toro and Smith subroutines, Reaction 44, Table 6.2) and the Steele formulation (Steele, 1962) that can use hourly light input values. The choice between different available functional forms (Ditoro, Smith, and Steele) is made by setting the index `LGHTSW` equal to 1, 2 or 3. The new version is therefore able to simulate diurnal variations depending on light intensity, such as night anoxia due to phytoplankton respiration during nighttime.
Finally, the two frequently used models for combining maximum growth and limiting factors, the multiplicative and the minimum (or Liebig's) model, are both implemented, and the user can choose which one to adopt (Reaction 41, Table 6.2).
Nitrogen and phosphorous are then returned to the organic compartment (ON, OP) via phytoplankton and zooplankton respiration and death. After mineralization, the organic form is again converted into the dissolved inorganic form available for phytoplankton growth.
The DO mass balance is influenced by almost all of the processes going on in the system. The reaeration process acts to restore the thermodynamic equilibrium level, the saturation value, while respirations activities and mineralization of particulated and dissolved organic matter consume DO and, of course, photosynthetic activity produces it. Other terms included in the DO mass balance are the ones referring to redox reactions such as nitrification and denitrification. The reaeration rate is computed from the model in agreement with either the flow-induced rate or the wind-induced rate, whichever is larger. The wind-induced reaeration rate is determined as a function of wind speed, water and air temperature, in agreement with O'Connor (1983), while the flow-induced reaeration is based on the Covar method (Covar, 1976), i.e., it is calculated as a function of current velocity, depth and temperature.
The dynamic of a generic herbivorous zooplankton compartment (ZOO), meant to be representative of the pool of all the herbivorous zooplankton species, is followed and accordingly the subroutines relative to phytoplankton, organic matter, nutrients, and dissolved oxygen, which were influenced by such a modification.
The grazing has been described by means of a type II functional relationship, as it is usually done for aquatic ecosystems. However, the possibility to select a type III relationship, as well as to maintain the original parameterisation of constant zooplankton, has been included.
The zooplankton assimilates the ingested phytoplankton with an efficiency EFF, and the fraction not assimilated, ecologically representative of faecal pellets and sloppy feeding, is transferred to the organic matter compartments (dotted lines Fig. ??). Finally, zooplankton mortality is described by a first order kinetics. The code has been written by adopting the standard WASP nomenclature system, and the choice between the different available functional forms is performed by setting the index `IGRAZ`. A choice of 0 (the default value) corresponds to the original EUTRO version, giving the user the ability to chose easily between the extended version or revert to the original one.

6.2 The coupling

Mathematical models usually describe the coupling between ecological and physical process by suitable implementation of an advection/diffusion equation for a generic tracer, reads

$$\frac{\partial \Theta_i}{\partial t} + U \cdot \nabla \Theta_i - w_i^s \frac{\partial \Theta_i}{\partial z} = K_h \nabla_H^2 \Theta_i + \frac{\partial}{\partial z}\left[K_v \frac{\partial \Theta_i}{\partial z}\right] + F\left(\Theta, T, I, ..\right) \tag{6.1}$$

where U is the (average components of the) velocity, the Θ_i are the tracers which compose the entire vector of the biological state variable Θ and F is a source term. T and I indicate, respectively, water temperature and Irradiance level, while w_i^s represent the downward

$\frac{\partial S}{\partial t} = Q(S)$		General Reactor Equation
$Q(PHY) = GPP - DPP - GRZ$	1	Phytoplankton PHY [mg C/L]
$Q(ZOO) = GZ - DZ$	2	Zooplankton ZOO [mg C/L]
$Q(NH3) = N_{alg1} + ON1 - N_{alg2} - N1$	3	Ammonia NH3 [mg N/L]
$Q(NOX) = N1 - NO_{alg} - NIT1$	4	Nitrate NOX [mg N/L]
$Q(ON) = ON_{alg} - ON1$	5	Organic Nitrogen ON [mg N/L]
$Q(OPO4) = OP_{alg1} + OP1 - OP_{alg2}$	6	Inorganic Phosphorous OPO4 [mg P/L]
$Q(OP) = OP_{alg3} - OP1$	7	Organic Phosphorous OP [mg P/L]
$Q(CBOD) = C1 - OX - NIT2$	8	Carbonaceous Biological Oxygen Demand CBOD [mg O_2/L]
$Q(DO) = DO1 + DO2 + DO3 - DO4 - N2 - OX - SOD$	9	Dissolved Oxygen DO [mg O_2/L]

Table 6.1: Mass balances

flux rates (sinking velocity) for the tracer Θ_i, and K_h and K_v are the eddy coefficients for horizontal and vertical turbulent diffusion.
The term F includes the contributions of the biological/biogeochemical activities, and the whole biological state vector Θ is explicitly considered in the last term of equation 6.1, without a spatial operator. As far as the biologically induced variations are regarded, the fate of each tracer in every location x, y, z is tightly coupled to other tracers in the same location, but is not directly influenced by processes going on elsewhere.
Therefore, in this approximation the global temporal variation of any tracer (state variable, conservative or not) can be split into the sum of two independent contributions:

$$\frac{\partial \Theta_i}{\partial t} = \left.\frac{\partial \Theta_i}{\partial t}\right|_{phys} + \left.\frac{\partial \Theta_i}{\partial t}\right|_{biol} \tag{6.2}$$

and it might be convenient, in writing a computer code, to devote independent modules to computation of each of them. Indeed, most of the modern water quality programs do have, at least conceptually, a modular structure. In this way the same code can be used for simulating different situations: by switching off the module referring to the reactor term the transport of a purely passive tracer is reproduced, while a 0D, close and uniformly stirred biological system is simulated if the module referring to the physical term is not included. Finally, the inclusion of both modules gives the evolution of tracers subjected to both physical and biogeochemical transformation, in a representation that, depending upon the parameterisation of the physical module, can be 1, 2 or 3 dimensional.
The whole water quality module is contained in a file `weutro.f` and the call to EUTRO is made through a subroutine call that is done from the main program through an appropriate interface. There is a clean division between the hydrodynamic motor, parameters used by the model and the resolution of the differential equations and the ecological model as evidenced by the overall structure of the modules.
It is the responsibility of the main module to implement the time loop administration, the advective and diffusive transport of the state variables, both in the horizontal and vertical direction and the application of the boundary conditions.
The typical use of the new EUTRO module is as follows: the main program first sets all parameters needed in EUTRO through the call to `EUTRO_INI`. These parameters are the kinetic constants of the reactions that are described in EUTRO and are considered constant

$GPP = GP1 * PHY$	10	phytoplankton growth
$DPP = DP1 * PHY$	11	phytoplankton death
$GRZ = KGRZ * \frac{PHY}{PHY+KPZ} * ZOO$	12	grazing rate coefficient
$GP1 = L_{nut} * L_{light} * K1C * K1T^{(T-T_0)}$	13	phytoplankton growth rate with nutrient and light limitation
$DP1 = RES + K1D$	14	phytoplankton respiration and death rate
$GZ = EFF * GRZ$	15	zooplankton growth rate
$DZ = KDZ * ZOO$	16	zooplankton death rate
$Z_{ineff} = (1 - EFF) * GRZ$	17	grazing inefficiency on phytoplankton
$Z_{sink} = Z_{ineff} + DZ$	18	sink of zooplankton
$N_{alg1} = NC * DPP * (1 - FON)$	19	source of ammonia from algal death
$N_{alg2} = PN * NC * GPP$	20	sink of ammonia for algal growth
$NO_{alg} = (1. - PN) * NC * GPP$	21	sink of nitrate for algal growth
$ON_{alg} = NC * (DPP * FON + Z_{sink})$	22	source of organic nitrogen from phytoplankton and zooplankton death
$N1 = KC_{nit} * KT_{nit}^{(T-T_0)} * NH3 * \frac{DO}{K_{nit}+DO}$	23	nitrification
$NIT1 = KC_{denit} KT_{denit}^{(T-T_0)} * NOX * \frac{K_{denit}}{K_{denit}+DO}$	24	denitrification
$ON1 = KNC_{\min} * KNT_{\min}^{(T-T_0)} * ON$	25	mineralization of ON
$OP1 = KPC_{\min} * KPT_{\min}^{(T-T_0)} * OP$	26	mineralization of OP
$OP_{alg1} = PC * DPP * (1. - FOP)$	27	source of inorganic phosphorous from algal death
$OP_{alg2} = PC * GPP$	28	sink of inorganic phosphorous for algal growth
$OP_{alg3} = PC * (DPP * FOP + Z_{sink})$	29	source of organic phosphorous from phytoplankton and zooplankton death
$OX = KDC * KDT^{(T-T_0)} * CBOD * \frac{DO}{KBOD+DO}$	30	oxidation of CBOD

Table 6.2: Functional Expression Description

$C1 = OC * (K1D * PHY + Z_{sink})$	31	source of CBOD from phytoplankton and zooplankton death
$NIT2 = \left(\frac{5}{4} * \frac{32}{14} * NIT1\right)$	32	sink of CBOD due to denitrification
$DO1 = KA * (O_{sat} - DO)$	33	reareation term
$DO2 = PN * GP1 * PHY * OC$	34	dissolved oxygen produced by phytoplankton using NH3
$DO3 = (1 - PN) * GP1 * PHY * 32 * \left(\frac{1}{12} + 1.5 * \frac{NC}{14}\right)$	35	growth of phytoplankton using NOX
$DO4 = OC * RES * PHY$	36	respiration term
$N2 = \left(\frac{64}{14} * N1\right)$	37	oxygen consumption due to nitrification
$PN = \frac{NH3*NOX}{(KN+NH3)*(KN+NOX)} + \frac{NH3*KN}{(NH3+NOX)*(KN+NOX)}$	38	ammonia preference
$RES = K1RC * K1RT^{(T-T_0)}$	39	algal respiration
$SOD = \frac{SOD1}{H} * SODT^{(T-T_0)}$	40	sediment oxygen demand
$L_{nut} = min(X1, X2), mult(X1, X2)$	41	minimum or multiplicative nutrient limitation for phytoplankton growth
$X1 = \frac{NH3+NOX}{KN+NH3+NOX}$	42	nitrogen limitation for phytoplankton growth
$X2 = \frac{OPO4}{\frac{KP}{FOPO4}+OPO4}$	43	phosphorous limitation for phytoplankton growth
$L_{light} = \frac{I_0}{I_s} * e^{-(KE*H)} * e^{\left(1-\frac{I_0}{I_s}*e^{(-KE*H)}\right)}$	44	light limitation for phytoplankton growth
$KA = F(Wind, Vel, T, T_{air}, H)$	45	re-areation coefficient

Table 6.2: (continued) Functional Expression Description

$K1D = 0.12$ day^{-1}	phytoplankton death rate constant
$KGRZ = 1.2$ day^{-1}	grazing rate constant
$KPZ = 0.5$ mg C/L	half saturation constant for phytoplankton in grazing
$KDZ = 0.168$ day^{-1}	zooplankton death rate
$K1C = 2.88$ day^{-1}	phytoplankton growth rate constant
$K1T = 1.068$	phytoplankton growth rate temperature constant
$KN = 0.05$ mg N/L	nitrogen half saturation constant for phytoplankton growth
$KP = 0.01$ mg P/L	phosphorous half saturation constant for phytoplankton growth
$KC_{nit} = 0.05$ day^{-1}	nitrification rate constant
$KT_{nit} = 1.08$	nitrification rate temperature constant
$K_{nit} = 2.0$ mg O_2/L	half saturation constant for nitrification
$KC_{denit} = 0.09$ day^{-1}	denitrification rate constant
$KT_{denit} = 1.045$	denitrification rate temperature constant
$K_{denit} = 0.1$ mg O_2/L	half saturation constant for denitrification
$KNC_{min} = 0.075$ day^{-1}	mineralization of dissolved ON rate constant
$KNT_{min} = 1.08$	mineralization of dissolved ON rate temperature constant
$KDC = 0.18$ day^{-1}	oxidation of CBOD rate constant
$KDT = 1.047$	oxidation of CBOD rate temperature constant
$NC = 0.115$ mg N/mg C	N/C ratio
$PC = 0.025$ mg P/mg	C P/C ratio
$OC = 32/12$ mg O_2/mg C	O/C ratio
$EFF = 0.5$	grazing efficiency
$FON = 0.5$	fraction of ON from algal death
$FOP = 0.5$	fraction of OP from algal death
$FOPO4 = 0.9$	fraction of dissolved inorganic phosphorous
$KPC_{min} = 0.0004$ day^{-1}	mineralization of dissolved OP rate constant
$KPT_{min} = 1.08$	mineralization of dissolved OP rate temperature constant
$KBOD = 0.5$ mg O_2/L	CBOD half saturation constant for oxidation
$K1RC = 0.096$ day^{-1}	algal respiration rate constant
$K1RT = 1.068$	algal respiration rate temperature constant
$I_s = 1200000$ lux/day	optimal value of light intensity for phytoplankton growth
$KE = 1.0$ m^{-1}	light extinction coefficient
$SOD1 = 2.0$ mg O_2/L day^{-1} m	sediment oxygen demand rate constant
$SODT = 1.08$	sediment oxygen demand temperature constant
$T_0 = 20$ ^{o}C	optimal temperature value

Table 6.3: Parameters

T	[^{o}C]	water temperature
T_{air}	[^{o}C]	air temperature
O_{sat}	[mg/L]	DO concentration value at saturation
I_0	[lux/day]	incident light intensity at the surface
H	[m]	depth
Vol	[m^3]	volume
Vel	[m/sec]	current speed
$Wind$	[m/sec]	wind speed

Table 6.4: Variables

for a site. They have to be set therefore only once at the beginning of the simulation. Once set, these parameters are available to the EUTRO module as global parameters.
For every box in the discretized domain (horizontal and vertical), and for every time step, the main program calls the subroutine `EUTRO0D`. Inside `EUTRO0D` the differential equations that describe the bio-chemical reactions are solved with a simple Euler scheme.
The values passed into `EUTRO0D` can be roughly divided into 4 groups. The first group is made out of the aforementioned constants that represent the kinetic constants and other parameters that do not vary in time and space. The second group represents the state variables that are actually modified by `EUTRO0D` through the bio-chemical reactions. These variables are transported and diffused by the main routine and are just passed into `EUTRO0D` for the description of the processes. After the call no memory remains in `EUTRO0D` of these state variables. They must therefore be stored away by the main routine to be used in the next time step again. The third and fourth groups of values have to do with the forcing terms. They have been divided in order to account for the different nature of the forcing terms. The third group consists of the hydrodynamic forcing terms that are directly computed by the hydrodynamic model and parameters related to the box discretization. They consist of water temperature, salinity, current velocity, and depth and type of the box. Here the type identifies the position of the box (surface, water column, sediment), which is needed for some of the forcings to be applied. These variables are passed directly into EUTRO through a parameter list. The last group contains other forcing terms that are not directly related to the hydrodynamic model. These consist of the meteorological forcings (wind speed, air temperature, ice cover), light climate (surface light intensity, day length) and sediment fluxes. These parameters are set through a number of commodity functions that are called by the main routine. The reason why the last two parameter groups are handled differently from each other has also to do with the fact that the third group is highly variable in time and space. Variables like current velocity change with every time step and are normally different from element to element. The fourth group is very often only slowly variable in time (light, wind) and can very often be set constant in space. Therefore these values can be set at larger intervals, and do not have to be changed when looping over all the elements in the domain.
The overall flow of information during one time step is the following: First the hydrodynamic model resolves the momentum and continuity equation to update the current velocities and water levels. After that the physical (temperature and salinity) and bio-chemical scalars are advected and diffused. Once this advection step has been handled the new loadings and forcing terms are set-up and then `EUTRO0D` is called for the bio-chemical reactions. Note that the operator splitting technique, which decouples the advective and diffusive transport from the source term, allows for different time steps of the two processes for a more efficient use of the computer resources.
Default values of the water quality parameters are already set in the code. Owner specific parameters for the water quality model should be written in the subroutine `param_user`.

6.3 Light limitation

6.3.1 Light attenuation formula by Steele and Di Toro

The well known light limitation function proposed by Steele is given as:

$$P = \frac{I}{I_o} e^{1-\frac{I}{I_o}} \tag{6.3}$$

where I is the light intensity and I_o the optimal light intensity. P is the limiting function and takes values between 0 and 1.
In this form, P is a function of depth z and of time t ($P = P(z,t)$) since the light intensity depends on depth and time ($I = I(z,t)$). The depth dependence of I can be written as

$$I(z) = I_i e^{-kz} \tag{6.4}$$

where I_i is the incident light intensity on the surface (still dependent on time) and k is an extinction coefficient. Inserting (6.4) into (6.3) gives the equation

$$P(z,t) = \frac{I_i}{I_o} e^{-kz} e^{1-\frac{I_i}{I_o}e^{-kz}} = e\frac{I_i}{I_o} e^{-kz} e^{\frac{I_i}{I_o}e^{-kz}}. \tag{6.5}$$

We now compute the average of P over the water column. This gives

$$\overline{P(t)} = \frac{1}{H}\int_0^H P(z,t)dz = \frac{e\frac{I_i}{I_o}}{H}\int_0^H e^{-kz} e^{-\frac{I_i}{I_o}e^{-kz}} dz.$$

With the substitution $x = e^{-kz}$ and therefore $dx/dz = -ke^{-kz} = -kx$ the integral can be transformed into

$$\overline{P(t)} = \frac{e\frac{I_i}{I_o}}{H}\int_0^H x e^{-\frac{I_i}{I_o}x} \frac{-1}{kx} dx = -\frac{e\frac{I_i}{I_o}}{kH}\int_0^H e^{-\frac{I_i}{I_o}x} dx.$$

Solving this integral gives finally

$$\overline{P(t)} = \frac{e}{kH}\left[e^{-\frac{I_i}{I_o}x}\right]_0^H = \frac{e}{kH}\left[e^{-\frac{I_i}{I_o}e^{-kz}}\right]_0^H = \frac{e}{kH}\left[e^{-\frac{I_i}{I_o}e^{-kH}} - e^{-\frac{I_i}{I_o}}\right]. \tag{6.6}$$

This is the depth integrated form of the Steele limiting function for instantaneous light.
If we want to work with the average light over one day, then equation (6.6) can be easily averaged over one day. If T is the averaging period (one day), f the fraction of the day with daylight and I_i is approximated with a step function, 0 at night and I_a during daytime, then we can write

$$\overline{P} = \frac{1}{T}\int \overline{P(t)}dt = \frac{1}{T} fT \frac{e}{kH}\left[e^{-\frac{I_a}{I_o}e^{-kH}} - e^{-\frac{I_a}{I_o}}\right] = \frac{e}{kH} f \left[e^{-\frac{I_a}{I_o}e^{-kH}} - e^{-\frac{I_a}{I_o}}\right]. \tag{6.7}$$

Equation (6.7) represents the limiting function given by Di Toro and used in the EUTRO program of WASP. Therefore, equation (6.7) is just the Steele limiting function, but using I_a, the average incident light intensity over daylight hours instead the instantaneous incident light intensity I_i used in the original Steele formula (6.6).

6.3.2 Light attenuation formula by Smith

EUTRO uses also a light limitation formula by Smith. In the manual this is given as

$$\overline{P(t)} = \frac{e}{kH}\left[e^{-\frac{I_i}{I_s}e^{-kH}} - e^{-\frac{I_i}{I_s}}\right]. \tag{6.8}$$

where now I_S is the optimal light intensity which is not a constant but is continuously adjourned by the program. Again, I_i is the instantaneous light intensity at the surface.
I_i is given as

$$I_i = \frac{\pi I_t}{2f} \sin(\frac{\pi t}{f}) \tag{6.9}$$

between 0 and f (daylight) and $I_i = 0$ otherwise. Averaging (6.9) over one whole day (0–1) gives

$$\overline{I_i}^{day} = \int_0^f \frac{\pi I_t}{2f} \sin(\frac{\pi t}{f}) dt = I_t$$

and averaging only over daylight hours gives

$$\overline{I_i}^{daylight} = \frac{1}{f} \int_0^f \frac{\pi I_t}{2f} \sin(\frac{\pi t}{f}) dt = \frac{I_t}{f} = I_a.$$

This shows that I_t in equation (6.9) is the average light intensity at the surface over one whole day (`ITOT` in Eutro) and that $I_a = I_t/f$ is the average light intensity at the surface over daylight hours. This must be taken into account when the Di Toro formulation is used. Note that in EUTRO the actual formula used is $I_a = 0.9 I_t/f$ where the parameter 0.9 probably accounts for some losses during the integration. The corresponding variables in EUTRO are `FDAY` for f and `IAV` for I_a.

6.4 Initialization

This section describes the interpolation of data for the initialisation of the model.
To create a file with initial conditions the program `laplap` can be used. The program is called as `laplap < namefile.dat`. This makes a laplacian interpolation of specified data contained in the `namefile.dat`. This data file should have the first line empty and shold contain two colums containing, respectively, node number and data values for the node.
It generates two files, `laplap.nos` and `laplap.dat`. The first one can be used to check if the procedure has been conducted well, creating a map with the plotting procedure (see Postprocessing section). The `.dat` file name should be given in the section `$name` of the `.str` file to initialize the model.
You can create initialisation files for temperature, salinity, wind field and biological variables. If you want to initialise the biological model with biological data you should create a single data file merging the 9 data files (one for each variable) using the `inputmerge.f` routine.

6.5 Post processing

This section shows how to generate derivate variables.
The post processing routines elaborate the water quality outputs to generate derivate variables. They allow to generate variables such as averages, (both, in time and in space), sum differences, and water quality variables such us Vismara, TRIX and BOD5.
The routines and their usage are the following:

`nosmaniav.f` It generates a file containing, for each node of the spatial domani, average, minimum and maximum values of the specified variable of the whole simulation.

`nosmaniqual.f` It generates a water quality file from the elaboration of the state variable. It computes, for each node, and at each time step a water quality index that can be chosen between two suggested indexes: Vismara QualityV and TRIX a well known quality index applied to the water quality definition at coastal seas and estuaries.

Class(Var)	1	2	3	4	5
O2sat	90-110	70-90 or 110-120	50-70 or 120-130	30-50	<30 or >130
BOD5	<3	3-6	6-9	9-15	>15
NH3	<0.4	0.4-1	1-2	2-5	<5

Table 6.5: Classification of Water Quality Indices

These indices can be computed using the definitions in Table 6.5 and the following equations:

```
QualityV = class(O2sat) + class(BOD5) + class(NH3)
```

auxt1 = (phyto/30.)*1000
auxt2 = (nh3+nox)*1000
auxt3 = (opo4+op)*1000
TRIX = log10(auxt1*o2satp*auxt2*auxt3+1.5)/1.2

nosmanintot.f generates a file of total inorganic Nitrogen as sum of NH3 and NOx

nosdif.f computer for the chosen state variable, the difference between the values at two times step.

nosdiff.f computer the difference between the variable outputs of two simulations

nosmanibod5.f computes the BOD5 values from the CBOD outputs as:

```
bod5 = cbod*(1. - exp( -5. * par1 ))
```

+ (64./14.) * nh3 * (1. - exp(-5. * par2))

To run one of the postprocessing routine write the name of the routine and enter.

6.6 The Sediment Module

The sediment buffer action on the biogeochemical cycles could be very important, especially in the shallow water basins and during the storm surge events.
The routine `wsedim` (introduced in April 2004) aims to address the resuspention/sinking dynamics of nitrogen and phosphorous. This routine can be switched on and off as needed by the user, setting the `bsedim` parameter `true` or `false` in the `bio3d` routine. It is called after the the eutrophication subroutine.
It allows to follow the dynamics of two additional variables, OPsed and ONsed that simulate the evolution of Nitrogen and Phosphorous detritus in sediment that are not subjected to advection-diffusion processes.
These two variables interact with the Nitrogen and Phosphorous cycle as decribed by the equations in Table 6.6. When the `wsedim` subroutine is switched on, OP, ON, NH3 and OPO4 are updated at each time step in agreement with those equations.
The resuspension is a linear function of the water velocity calculated by the hydrodynamic model at each box, as written in Table 6.7. The amount of the sinking nutrients depends on specific prossess parameters, as given in Table 6.7, and on the depth of the underlying column.

$\frac{\partial S}{\partial t} = Q(S)_{sed}$		General Reactor Equation
$Q(NH3)_{sed} = NH3_{res}$	3	Ammonia NH3 [mg N/L]
$Q(ON)_{sed} = (ON_{res} - ON_{sink})$	5	Organic Nitrogen ON [mg N/L]
$Q(OPO4)_{sed} = OPO4_{res}$	6	Inorganic Phosphorous OPO4 [mg P/L]
$Q(OP)_{sed} = (OP_{res} - OP_{sink})$	7	Organic Phosphorous OP [mg P/L]
$Q(ON_{sed}) = ON_{sink} - ON_{res} - NH3_{res}$	1sed	Sediment Organic Nitrogen ON$_{sed}$ [mg N/L]
$Q(OP_{sed}) = OP_{sink} - OP_{res} - OPO4_{res}$	2sed	Sediment Organic Phosphorous OP$_{sed}$ [mg N/L]

Table 6.6: Sediment Mass balances

$NH3_{res} = Vol_{sed} * KNC_{sed} * KNT^{(T-T_0)} * ON_{sed}$	1	Mineralization of organic nitrogen in sediment
$ON_{res} = Vol_{sed} * KN_{res} * F_{vel} * ON_{sed}$	2	Resuspention of organic nitrogen in sediment
$ON_{sink} = Vol * \frac{(1-exp(-dt/\tau_N))}{dt} * ON * FPON$	3	Sink of organic nitrogen from the water column
$OPO4_{res} = Vol_{sed} * KPC_{sed} * KPT^{(T-T_0)} * OP_{sed}$	4	Mineralization of organic phosphorous in sediment
$OP_{res} = Vol_{sed} * KP_{res} * F_{vel} * OP_{sed}$	5	Resuspention of organic phosphorous in sediment
$OP_{sink} = Vol * \frac{(1-exp(-dt/\tau_P))}{dt} * OP * FPOP$	6	Sink of organic phosphorous from the water column
$\tau_N = H/w_s$	7	Time scale for sinking processes of organic N
$\tau_P = H/w_s$	8	Time scale for sinking processes of organic P

Table 6.7: Sediment functional expressions

$KNC_{sed} = 0.075$	Mineralization of sediment ON rate constant
$KNT = 1.08$	Mineralization of sediment ON rate temperature constant
$KPC_{sed} = 0.22$	Mineralization of sediment OP rate constant
$KPT = 1.08$	Mineralization of sediment OP rate temperature constant
$KN_{res} = 0.1$	Fraction of sediment depth resuspended/day
$KP_{res} = 0.1$	Fraction of sediment depth resuspended/day
$FPON = 0.5$	Fraction of particulate organic N
$FPOP = 0.5$	Fraction of particulate organic P
$F_{vel} = 1$	Velocity coefficient
$w_s = 10$ m/day	Sinking velocity
$T_0 = 20\ ^o$C	Optimal temperature value

Table 6.8: Sediment parameters

Vol	[m^3]	volume
Vol_{sed}	[m^3]	sediment volume
H	[m]	total depth of water column
dt	[sec]	time step

Table 6.9: Sediment variables

6.6.1 Parameters for the Water Quality Module

Compulsory bio parameters These parameters are compulsory parameters that define if the water quality module is run and what kind of output is written.

`ibio`	Flag if the computation on the temperature is done. The model writes at each time step the state variable values in the the .bio output file
`itsmed`	Flag if the average, minimum, maximum file of variables bio, salinity, temperature is done. if `itsmed=1` the model writes `.sav`, `.tav` output files of the corresponding variables.

Boundary conditions Boundary conditions have to be given in a file in every section `$bound`.

`bio2dn`	File name that contains boundary conditions for concentration of the water quality state variables. The format is the same as for the file `boundn`. The unit of the values given in the second and following column (9 data columns for EUTRO) must the ones of the variable.

Initial conditions Initialization of variables are done by file. The files can be created by the progam `laplap`. They have to be given in section `$name`.

`bio`	File with concentration values of water quality variable to be used for the initialization.
`salt` `temp`	Files with salinity concentration values [psu] and Temperature values [deg C] for the initialization.
`conz`	Files with tracer concentration values [for the initialization.

Bibliography

[1] Jan O. Backhaus. A semi-implicit scheme for the shallow water equations for aplication to shelf sea modelling. *Continental Shelf Research*, 2(4):243–254, 1983.

[2] Kurt C. Duwe and Regina R. Hewer. Ein semi-implizites gezeitenmodell für wattgebiete. *Deutsche Hydrographische Zeitschrift*, 35(6):223–238, 1982.

[3] G. Grotkop. Finite element analysis of long-period water waves. *Computer Methods in Applied Mechanics and Engineering*, 2(2):147–157, 1973.

[4] B. Herrling. Computation of shallow water waves with hybrid finite elements. *Advances in Water Resources*, 1:313–320, 1978.

[5] Bruno Herrling. Ein finite-element-modell zur berechnung von Tideströmungen in ästuarien mit Wattflächen. *Die Küste*, 31:102–113, 1977.

[6] K.-P. Holz and G. Nitsche. Tidal wave analysis for estuaries with intertidal flats. *Advances in Water Resources*, 5:142–148, 1982.

[7] Michael Kwizak and André J. Robert. A semi-implicit scheme for grid point atmospheric models of the primitive equations. *Monthly Weather Review*, 99(1):32–36, 1971.

[8] A. Schoenstadt. A transfer function analysis of numerical schemes used to simulate geostrophic adjustment. *Monthly Weather Review*, 108:1248, 1980.

[9] C. Taylor and J. Davis. Tidal and long wave propagation—a finite element approach. *Computers & Fluids*, 3:125–148, 1975.

[10] Georg Umgiesser. A model for the Venice Lagoon. Master's thesis, University of Hamburg, 1986.

[11] Georg Umgiesser and Andrea Bergamasco. A staggered grid finite element model of the Venice Lagoon. In J. Periaux K. Morgan, E. Ofiate and O.C. Zienkiewicz, editors, *Finite Elements in Fluids*. Pineridge Press, 1993.

[12] R. T. Williams. On the formulation of finite-element prediction models. *Monthly Weather Review*, 109:463, 1981.

[13] R. T. Williams and O. C. Zienkiewicz. Improved finite element forms for the shallow-water wave equations. *International Journal for Numerical Methods in Fluids*, 1:81, 1981.