Centrum voor Wiskunde en Informatica

REPORT*RAPPORT*

New Generation Shelf Flux Models

G. Stelling, D. Roose, B.P. Sommeijer, P.J. van der Houwen,
J. Kok, H.X. Lin, K. Tan

Modelling, Analysis and Simulation (MAS)

MAS-R9815 September, 1998

Report MAS-R9815
ISSN 1386-3703

CWI
P.O. Box 94079
1090 GB Amsterdam
The Netherlands

P.O. Box 94079, 1090 GB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

New Generation Shelf Flux Models

G. Stelling[1)], D. Roose[2)], B.P. Sommeijer[3)], P.J. van der Houwen[3)], J. Kok[3)],
H.X. Lin[4)], K. Tan[1)]

[1)] *Delft Hydraulics, P.O. Box 177, NL - 2600 MH Delft, The Netherlands*
[2)] *Katholieke Universiteit Leuven, De Croylaan 2, B - 3001 Heverlee, Belgium*
[3)] *CWI - Centre for Mathematics and Computer Science, Kruislaan 413, NL - 1098 SJ Amsterdam, The Netherlands*
[4)] *Delft University of Technology, Faculty of Mathematics and Informatics, Mekelweg 4, NL - 2600 GA Delft, The Netherlands*

ABSTRACT
The ecological condition of the Continental Shelf is of great concern for many countries. The understanding of the integrated effects that result in the present and future situation requires answers on many research topics, such as flow modelling, including turbulence and large eddy simulation, transport processes, chemistry, ecology, etc. This paper focuses on ‘high performance computing issues’ for flux modelling.

Flux modelling of contaminants, nutrients and ecosystem parameters in general in the Continental Shelf is essential to improve the understanding of this important ecosystem. From a computational point of view, flux modelling is a ‘grand challenge’; its computational demands are so huge that the present state-of-the-art does not yield numerical models of desired accuracy. Several assumptions and simplifications are needed to arrive at ‘manageable’ numerical models whose run times are acceptably low.

In an attempt to relieve the computational burden on flux modelling, a significant amount of research at the institutes participating in the NOWESP project has been and still is directed towards high performance computing techniques. In the quest for faster models several approaches are used:

- parallelization of (sequential) codes on parallel/vector computers of shared memory type;
- parallelization of (sequential) codes on parallel computers with distributed memory;
- development of numerical techniques that (are expected to) lead to better efficiency and robustness of flux models.

Developments of the latter kind usually take into account the use of a parallel machine but it also includes the implementation of sophisticated iterative solution techniques for solving systems of equations. This may already pay off on sequential machines.

In this paper we assess the progress made in recent years on high performance flux modelling. In the spirit of the learning-by-doing process adopted in the NOWESP project, this assessment is followed by some recommendations for future research to further overcome the lack of computational power that is needed for full scale flux models.

1991 Mathematics Subject Classification : 65M20, 65M55, 65L20
Keywords and Phrases: Transport models, Shallow water, HPCN, Domain Decomposition.
Note: Work carried out under project MAS1.2 - ‘Numerical Algorithms for Surface Water Quality Modelling’.
Note: The North-West European Shelf Programme (NOWESP) is an EU-funded project (within the MAST II project), in which 18 institutes from 8 European countries participated during the years 1993-1997. The primary objective of NOWESP was to quantify (in space and in time) the ecologically significant processes on the shelf, with the help of measured data and models.

1. Introduction

The complexity of flux modelling in the North Sea is highly influenced by the large variety of time and spatial scales. To mention but a few:

- Tidal variations;
- Day/night variation (temperature/thermocline);
- Seasonal variation (fresh water inflow, temperature, ...);
- Random variations (wind, emissions, temperature, ...);
- Spatial variations (vertical scales/horizontal scales, bottom topography, coastlines,...);
- Small scale random variations due to turbulent motion.

In principle, for accurate modelling, numerical models must be able to resolve all relevant scales. In practice, however, this is impossible. This leads to elimination of scales by some kind of averaging. Averaging leads to closure assumptions. Closure assumptions require skills and sometimes a-priori knowledge of the contaminant distribution by the modeller. This a-priori knowledge must increase as the averaging interval, in some dimension, is increasing. In general one could argue that a model formulation must fulfil the following requirements:

1. Given the modelling purpose, the model formulation must contain relevant physical effects and geometrical/topological details;
2. Given the modelling purpose, a stable numerical solution must be obtained within a reasonable amount of computational effort;
3. Given the modelling purpose, the numerical solution must be sufficiently accurate.

To fulfil requirement 1, the following processes are to be taken into account:

Flow

Of course, flow is input to flux modelling. The vertical structure of the almost horizontal Continental Shelf Flow plays an important role for transport of contaminants. Hence, flow models must simulate this structure with sufficient accuracy, and consequently the models must be three dimensional.

Turbulence

For the computation of the vertical structure and for subgrid effects in horizontal direction anisotropic turbulence models must be added.

Transport of salinity and heat

The intensity of turbulence is greatly influenced by density gradients. Density of water is primarily influenced by salinity and temperature, hence the model needs equations for the transport of salinity and heat.

Transport of contaminants including decay and reactions

The water quality of the North Sea follows from concentrations of constituents such as suspended sediments with heavy metals or biological substances such as bacteria. For a complete process description not only transport but also decay and reactions play a role.

For the processes as described, a large variation of model descriptions is available. If we apply state-of-the-art 3D flow models as are used for industrial flows, then we would end up with the following model formulations:

A.1 3D non-hydrostatic flow formulation;
A.2 At least a two equation turbulence model (e.g., k-ε), but preferably a ‘Reynolds stress model’;
A.3 Transport of heat and salinity;
A.4 Transport of contaminants based on eddy diffusivity resulting from (A.1-A.3).

Such a model (type A) seems to be ideally suitable for our aim of accurate flux modelling; however this is still impossible for computational reasons:

The model is of course three dimensional, and hence the grid is three dimensional. The spatial variation that should be resolved by the model is of the order of magnitude of the depth. In the Continental Shelf this is of O(10 m). The smaller scales are supposed to be turbulent eddies that are dissipated by the turbulence model. To represent phenomena with a minimum length scale of 10 m, a grid size of O(1 m) is imperative. If we assume isotropic turbulence, then this grid size must be isotropic as well. This means that the grid size must be of the same order of magnitude both in horizontal and vertical direction. For the Continental Shelf (1000∗1000 km ∗ 40 m) the number of grid points that is needed becomes roughly $4*10^{13}$. At each point at least 25 numbers are to be kept in memory. This requires a computer with a RAM memory of at least 4000 Tb (tera byte, 1 Tb=10^{12} byte). This is far beyond of what is presently available. Memories of 2 Gb (giga byte, 1 Gb=10^{9} byte) rank among the largest available. Also the computational speed is not sufficient for this kind of flux modelling. Some contaminants have a residence time of more than one year. If flux modelling is supposed to resolve the smallest scales, then the time step is supposed to be equivalent to the Courant number related to the flow speed. This implies a time step of 1 second. For one year of computation this requires $3.2*10^{7}$ time steps. Assuming that each time step needs at least 100 floating point operations per grid point, a total simulation will require $(3.2*10^{7})*(10^{2})*(4*10^{13})=1.3*10^{23}$ floating point operations for roughly one year of simulation. This requires 4000 years of computation on a ‘tera flop’ machine (10^{12} floating point operations per second). In summary, such a modelling approach is certainly impossible, also during the next decade.

To arrive at realistic models, more knowledge about the flow has to be built in into the model formulation. A general assumption relates to the difference in horizontal and vertical length scales, leading to the hydrostatic pressure assumption. This will allow larger horizontal grids of the order of 1 km. The grid becomes anisotropic, which automatically implies an anisotropic turbulence model. This is a consequence of the fact that the horizontal scales are much larger than the vertical scales. In other words, a so-called horizontal sub-grid model will be added to the equations. This gives the following type B model:

B.1 3D hydrostatic flow formulation;
B.2a At maximum, a two equation turbulence model for vertical eddy viscosity and diffusivity (e.g., k-ε);
B.2b Simple sub-grid formulation for horizontal dispersion of both vector and scalar quantities (simple constant or something like ‘Smagorinsky’, or other ‘Large Eddy Simulation’ formulations);
B.3 Transport of heat and salinity;
B.4 Transport of contaminants based on eddy diffusivity resulting from (B.1-B.3).

The number of grid points needed for such a model is roughly $4*10^{7}$. Again assuming 25 single precision numbers per grid point yields 4 Gb of required memory. The time step, because of the increased grid size, can be chosen of the order of magnitude of 100 seconds. This leads to a number of floating point operations for one year of simulation of $1.3*10^{15}$. On a tera flop machine this would lead to 1300 seconds of computation. However, most high performance computers at present perform in the giga flop range. This results in a computational time in the order of several days, provided the availability of software that performs in an optimal sense such that peak performance is realized. It seems though that within the foreseeable future this type of flux modelling will be possible. On today's single processor workstations this type of modelling would be at least 50 times as slow, leading again to computations that are not feasible. Within these B type models various assumptions can be made to reduce the computational effort or to reduce the process complexity such as:

- Flow averaging over a tidal cycle to generate residual flows;
- Simplification of vertical exchange models;
- Prescribing the density gradients instead of computing them (diagnostic *versus* prognostic mode).

Especially flow averaging eliminates smaller time scales such that larger time steps can be applied. This leads to models that can operate on today's workstations. The flow model must run in general for a neap-spring tide cycle and is therefore still quite demanding. Models of this type will be called ‘reduced B type models’. In practice, even reduced B type models are still yielding computational problems leading to the last class of models: C models. Here, the models are integrated in the vertical. This gives the following formulations:

C.1 Tidal averaged 2D hydrostatic flow formulation;
C.2 Horizontal (basically non-isotropic in horizontal direction) dispersion formulation, resulting from vertical integration;
C.3 At maximum horizontal density gradients;
C.4 Depth averaged transport of contaminants based on dispersion formulations.

Models of this C-type allow flexible operation on most of the hardware infrastructures available at European institutions. A drawback of these models is that the reduction of the vertical dimension neglects vertical variation of the density that is sometimes crucial for ecological parameters. Also the dependence of in principle non-isotropic dispersion coefficients in horizontal direction as a result of averaging over vertical velocity profiles is a serious aspect hampering the predictive ability of these type of models.
From the numerical/mathematical point of view model equations of type B consist of the following equations:

- Hydrostatic free surface flow equations (B.1);
- Transport equations including production and decay (B.1-4).

In general, numerical models of this type will be characterised by:

- Large difference between horizontal and vertical grid sizes. The vertical grid size is much smaller while at the same time the number of vertical grid points is much smaller too;
- Dominance of hydrostatic pressure in horizontal direction and of eddy viscosity in vertical direction for flow formulation;
- Dominance of advection in horizontal direction and of diffusion in vertical direction for transport formulation;
- Large stiffness ratio for all equations because of grid size variation and production/decay formulations.

For an efficient numerical simulation, it is crucial to take these typical characteristics into account. This motivates the development of special techniques for these problems. In the approaches of all the NOWESP partners this has played a role. For efficient implementation the following aspects have been considered:

- Parallelization of existing code;
- Domain decomposition, including fast iterative methods for implicit equations resulting from implicit time integration techniques;
- Special transport solvers for transport of contaminants;
- Efficient vertical discretization for vertical flow structure including k-ε models;
- Special approaches for vertical discretization to reduce the ‘hydrostatic consistency’ limitation of the so-called σ *-transformation.*

In the following sections each of these topics will be briefly described.

2. Parallelization of existing code

At the start of the NOWESP project, several partners involved in Task Group B (the so-called 'Modelling Group') already had a lot of experience with parallel (vector) computers with shared memory. During the last years however, there has been a growing interest in parallel computers with distributed memory, because they offer more perspective for high performances at a reasonable price. Therefore K.U. Leuven and T.U. Delft have worked on the parallelization of numerical models for execution on parallel computers with distributed memory. When writing code for parallel machines it is tempting to start from existing code and try to parallelize (and/or vectorize) it without modifying or replacing the underlying numerical algorithms. This approach can only be used when the underlying algorithm is sufficiently parallelizable. However, this approach often fails, especially when one wishes to use many processors. In this case, the parallel 'scalability' of the algorithms is very important. This is illustrated by the research done at K.U. Leuven and at T.U. Delft on the parallelization of numerical models for the Continental Shelf, which are based on the ADI (Alternating Direction Implicit) time integration method. The ADI method is a powerful numerical technique, which is however difficult to parallelize on distributed memory systems because it requires a rather large amount of communication. The research group at K.U. Leuven investigated several alternative strategies for the parallelization of the ADI method, within the framework of solving the Shallow Water Equations (Song, et al.^{1995}). These strategies have been implemented on two different parallel computers, viz., an Intel iPSC/860 and an IBM SP2. For a model for the Continental Shelf with a resolution of 4 km * 4 km (44302 wet grid points), the performance results are very satisfactory up to 32 processors. For example, on a 16 processor iPSC/860, a parallel speedup of 12.8 has been obtained (i.e., a parallel efficiency of 80%). On 32 processors, a speed-up of 22 has been obtained.

Also a different approach can be followed. By introducing slight modifications in the ADI method, the communication between processors can be decreased substantially, and hence the parallelization becomes easier and also more efficient. However, this modification also leads to a slower numerical convergence. This approach has been followed by the research group at T.U. Delft. In co-operation with RIKZ (Rijkswaterstaat) and other partners within the NOWESP project, they have investigated and implemented parallel versions of the TRIWAQ software. TRIWAQ is a three dimensional flow and transport simulation program developed at RIKZ of the Dutch Rijkswaterstaat, in which ADI is used as the time integration method. Parallel versions of TRIWAQ have been developed that run on the Parsytec PowerXplorer and a cluster of workstations (cf. Roest, et al.^{1996} and Vollebregt, et al.^{1996}). For a small model involving only 5000 grid points, speed-up factors of 17 and 30 have been obtained with 24 and 64 processors, respectively. For larger models an even better speed-up can be expected. The increased computational speed of the parallel implementations has enabled RIKZ to make more complex and more accurate simulations.

Summarizing, it can be said that the use of parallel computers imposes two major requirements on the algorithms that can be used. First, the inherent parallelism of an algorithm should be as high as possible (which includes the necessity of a good load balancing). Second, the data dependency among the operations in an algorithm should be as low as possible. This is necessary for a high data locality and minimal communication overhead. It must be stressed that in order to realize this, it is essential to modify some of the existing numerical algorithms.

3. Domain decomposition

Domain decomposition has been recognized as an important tool in the development of new generation flux models. Apart from the possibility to design efficient parallel algorithms, it also provides a numerical modeller with more modelling flexibility. This is based on the observation that many flux models are based on the use of structured grids. Domain decomposition then opens the gate to e.g. a more flexible geometry handling, and locally defined grid resolutions (only fine where needed). As such, it may help in reducing the total number of grid points, a strategy which easily pays off considering the number of species typically involved in bio-chemical transport (see also section 4.1). Experiments have shown that for a typical application with a complex geometry, a domain decomposition approach for the flow solver gave already rise to a memory reduction of more than 75%. CWI's approach will be described in which the time integration method is 'loosely coupled in the horizontal direction' in order to minimize the work in a domain decomposition method. A different approach has been followed at Delft Hydraulics where research has been done on the construction of interface conditions. A proper choice of the interface conditions may reduce the computational complexity of a domain decomposition method with a factor 3. This idea has been implemented in the sequential version of Delft Hydraulics' flow solver (De Goede, et al.[1995]).

3.1 Iterative solvers

In many current flow solvers ADI factorisation methods are employed in solving systems of equations. The reason for this can be found in the past when ADI methods were used as the time integration technique. With little effort this could be extended to iteration methods of ADI-type. However, ADI iteration methods can diverge for large systems in which large Courant numbers occur. A joint study of K.U. Leuven and Delft Hydraulics shows that the use of modern iterative solvers may help in improving the robustness and efficiency of flow solvers. For one particular application, a long term simulation of the hydrodynamics in the Clyde, the time step could be taken 4 times as large. As a result, the model was approximately 3 times faster.

4. HPC time integration for bio-chemical transport

4.1 Introduction

Another example of high performance computing for flux modelling is given by CWI's research on new, parallelizable (and vectorizable) time integration methods for bio-chemical transport. The mathematical model describing transport processes of salinity, pollutants, etc. in water, combined with their bio-chemical interactions, is defined by an initial-boundary value problem for the system of 3-D advection-diffusion-reaction equations

$$\begin{aligned}\frac{\partial c_i}{\partial t}+\frac{\partial}{\partial x}(uc_i)+\frac{\partial}{\partial y}(vc_i)+\frac{\partial}{\partial z}(wc_i)=\frac{\partial}{\partial x}\left(\varepsilon_x\frac{\partial c_i}{\partial x}\right)+\frac{\partial}{\partial y}\left(\varepsilon_y\frac{\partial c_i}{\partial y}\right)+\frac{\partial}{\partial z}\left(\varepsilon_z\frac{\partial c_i}{\partial z}\right)\\ +g_i(t,x,y,z,c_1,\ldots,c_m), i=1,\ldots,m\end{aligned} \tag{1}$$

where c_i denote the unknown concentrations of the contaminants. The local fluid velocities u, v, w (to be provided by a hydrodynamic model) and the diffusion coefficients ε_x, ε_y, ε_z are assumed to be given functions. The equations in (1) are mutually coupled by means of the functions g_i which model the (concentration-dependent) bio-chemical reactions and emissions from sources. The definition of the physical domain and of the initial and boundary conditions completes the model. Following the Method of Lines approach, equation (1), together with the initial condition and the boundary conditions, is converted into the semi-discrete initial value problem

$$\frac{dC(t)}{dt}=F(t,C(t)):=H(t,C(t))+G(t,C(t)),C(t_0)=C_0. \tag{2}$$

Here, C is a vector of dimension mN containing the m concentrations c_i at the total number of $N := N_x*N_y*N_z$ grid points ($N_{x \text{ or } y \text{ or } z}$ denotes the number of grid points in the various spatial directions, respectively). The term $H(t,C(t))$ originates from the discretization of the advection-diffusion terms (including the boundary conditions), and *G(t,C(t))* is the discrete analogue of the reaction terms and emissions. Finally, C_0 contains the initial values.

Since the functions *H* and *G* have quite a different origin, they give rise to a completely different coupling of the unknowns: in *H* the concentrations of the various species are uncoupled, but there is of course a coupling in space, due to the underlying spatial differential operators (in our experiments we have used a third-order upwind formula to discretize the advection terms, and second-order symmetric discretization of the diffusion terms). In *G*, on the other hand, we have in each grid point a local coupling of the concentrations. Another observation is that *H* is linear in *C*, whereas *G* is usually non-linear. These observations should be taken into account in selecting a suitable time integration method. In this context, 'suitable' means that the method should have the following properties:

1. *sufficient stability*; in the Continental Shelf we are primarily concerned with transport in shallow seas, resulting in small values for the mesh size in the vertical direction. As a consequence, stiffness is introduced into the discrete system. To avoid unrealistically small time steps, time integration methods should be sufficiently stable, which excludes the use of fully explicit methods.
2. *manageable level of computational effort;* in order to keep the coupling of unknowns as loosely as possible which helps in reducing the computational complexity of the model, we strive for a reduction of the amount of implicitness, while maintaining sufficient stability. For this item it is also relevant that the time integration method has good vectorization and parallelization properties.
3. *realistic accuracy;* in this PDE context, high precision results (e.g., produced by high order methods) are usually not necessary. On the other hand, since predictions over long time periods are an essential part in these kind of simulations, first order accuracy is, in our opinion, too low. Therefore, we restrict our attention to methods that are second order in time.
4. *storage economy*; although present-day computers are equipped with large memories, the nature of flow problems, especially in three dimensions, still necessitates a careful selection of an algorithm with respect to its storage requirements. A situation in which we are dealing with $N=10^6$ grid points and m = 10 to 20 species, is certainly not unusual.
5. *domain decomposition*; in many practical situations, different resolutions in space will be required in various regions of the domain. For example, near the coasts and in estuaries a fine grid is unavoidable to capture the physical phenomena. A natural way to efficiently cope with this demand is to apply a domain decomposition approach, in which the various subdomains are discretized with an appropriate resolution. Then the (sub)problems on the various subdomains can be solved in parallel. However, to obtain an efficient process for the overall solution, the coupling of these subproblems should not be too tight, since in that case many iterations would be necessary to match the interface conditions on the boundaries of these subdomains. Therefore, we are aiming at methods that are 'loosely coupled in the horizontal direction'.

With these requirements in mind, CWI has developed two HPC time integration techniques. One is based on a splitting method (see, Sommeijer and Kok[1996], Van der Houwen and Sommeijer[1997]). The other one is based on the implicit BDF2 method in combination with a special iteration scheme which has been designed in order to reduce the amount of linear algebra work involved (see, Van der Houwen, et al.[1997] and Eichler-Liebenow, et al.[1997]). Experiments confirm that both time integration methods can be vectorized and parallelized very well. In the next sections we will concentrate on the performance results of both methods.

4.2 A splitting method of hopscotch type

The splitting method of hopscotch type (including specially designed band solvers to exploit vectorization) has been implemented on the multi-processor Cray C90 vector machine. As a test problem we used a model with 2 species, including reaction terms describing their chemical interaction. We tested the code on three different spatial grids of increasing resolution:

Grid$_{\text{coarse}}$: $N_x = N_y = 41$, $N_z = 6$, amounting to ~ 6 10^3 internal grid points;
Grid$_{\text{middle}}$: $N_x = N_y = 81$, $N_z = 11$, amounting to ~ 5.6 10^4 internal grid points;
Grid$_{\text{fine}}$: $N_x = N_y = 161$, $N_z = 21$, amounting to ~ 4.8 10^5 internal grid points.

The performance results due to vectorization are summarized in Table 1.

Table 1. Global performance and speed-up factors obtained by the hopscotch method on various grids. The CPU times are per time step.

	Grid$_{\text{coarse}}$		Grid$_{\text{middle}}$		Grid$_{\text{fine}}$	
	CPU	Mflop	CPU	Mflop	CPU	Mflop
scalar	0.44	28.8	2.44	30.7	15.9	31.6
vector	0.062	206	0.28	276	1.46	351
speed-up	7.0		8.8		10.9	

Next, we investigated the speed-up that can be obtained by exploiting several processors of this machine. To that end we used a utility (available on the Cray) to estimate the speed-up factors, which are given in Table 2.

Table 2. Parallel performance of the hopscotch code.

processors	2	4	6	8	10	12	14	16
speed-up on Grid$_{\text{coarse}}$	1.92	3.63	5.05	5.97	7.00	8.25	8.46	8.64
speed-up on Grid$_{\text{middle}}$	1.95	3.75	5.41	6.90	7.80	9.65	9.92	11.17
speed-up on Grid$_{\text{fine}}$	1.96	3.75	5.41	6.94	8.14	9.71	10.43	12.06

In conclusion, we may say that this method is very suitable for implementation on a multi-processor vector machine, since the speed-up due to vectorization is close to 10 and, moreover, the algorithm possesses sufficient parallelism to obtain almost linear speed-up when using more processors. Compared with a sequential code run in scalar mode, a code run on p vector processors can be 10p times faster. The only disadvantage is of algorithmic nature: the method allows only modest CFL numbers (< 2.7) and has the peculiar property that the time step must go faster to zero than the mesh width in order to get convergence.This leads us to consider the approach described in the next section.

4.3 Iterative solution of a fully implicit method

In order to cope with the stiffness of the discrete system of ordinary differential equations (2), our starting point will be a fully implicit time discretization method (see also the first aim as formulated in Section 4.1). A suitable candidate is the second-order (cf. aim 3) Backward Differentiation Formula (BDF2). To solve the (non-linear) system of equations in each time step we employ a technique which is called ‘approximate factorization’ in the literature. This technique is efficient both in memory and in floating point operations. Moreover, it is vectorizable and, when extended with a domain decomposition approach, parallelizable. A convergence analysis shows (cf. Eichler-Liebenow, et al.[1997]) that large CFL numbers are now possible. As a result, in many practical situations, the time step can be chosen on the basis of accuracy considerations, rather than being restricted by stability conditions, while the amount of implicitness is kept quite modest (see aim 2). This method has been implemented and it turns out that 2-3 iterations are sufficient to (approximately) solve the underlying fully implicit BDF2 method. In comparing the BDF approach and the hopscotch method, we observe that:

- BDF yields a higher accuracy for the same step size;
- one step with BDF (in vector mode) on $\text{Grid}_{\text{coarse}}$ and $\text{Grid}_{\text{middle}}$ requires 0.024 sec. and 0.13 sec., respectively, which is more than twice as fast as hopscotch (compare the timings in Table 1);
- the CFL condition in the BDF approach is much better, allowing for larger time steps;
- the time step and the mesh width may arbitrarily tend to zero in order to let the BDF-solution converge to the PDE-solution.

From these considerations it is clear that the BDF approach is to be preferred. It needs, however, slightly more memory than the hopscotch approach.

5. Efficient vertical discretization for vertical flow structure including k-ε models

The 3D model for flow and transport simulation is supposed to be given by the following set of equations:

the continuity equation

$$\nabla \cdot \vec{u} = 0 \ , \tag{3.a}$$

the momentum equations in horizontal direction

$$\frac{Du_i}{Dt} + \frac{1}{\rho_0}\frac{\partial p}{\partial x_i} = \frac{\partial}{\partial x_j}\nu_H \frac{\partial u_i}{\partial x_j} + \frac{\partial}{\partial x_3}\nu_T \frac{\partial u_i}{\partial x_3}, i = 1,2, j = 1,2, \tag{3.b}$$

the hydrostatic pressure equation

$$\frac{\partial p}{\partial x_3} = -\rho g \ , \tag{3.c}$$

and the transport equation

$$\frac{\partial c_l}{\partial t} + \nabla \cdot \vec{F}_l = G_l,\ l = 1,\ldots,l_c \ , \tag{3.d}$$

where u_i = velocity in x_i direction, p = pressure, ρ = density, ρ_0 = reference density, c = concentration of scalar quantities (e.g., salt and temperature), F_l = concentration flux, due to advection and diffusion, ν_H = horizontal eddy viscosity, sometimes assumed to be a constant value sometimes related to some 'Large Eddy Simulation' approach such as the 'Smagorinky model', ν_T = vertical eddy viscosity computed by a k-ε model, $Du_i / Dt = \partial u_i / \partial t + u_j \partial u_i / \partial x_j$, where the summation convention is used, and G_l models the (concentration-dependent) bio-chemical reactions, production, dissipation or emissions from sources.

We will also use the following notation: $u=u_1$, $v=u_2$, $w=u_3$, $x=x_1$, $y=x_2$, $z=x_3$. The vertical co-ordinate z is bounded by

$$-\,d(x,y) \le z \le \zeta(x,y,t),$$

where d = depth below some plane of reference and ζ = water level above some plane of reference. The density ρ follows from an equation of state given by

$$\rho = \rho(s,T) \ , \tag{3.e}$$

where s = salinity and T = temperature. The eddy viscosity ν_T is, after dropping the index T, given by

$$\nu = c_\mu \frac{k^2}{\varepsilon} \ . \tag{3.f}$$

The transport equations for k and ε are given by

$$\frac{Dk}{Dt} = \frac{\partial}{\partial z}\left(\frac{\nu}{\sigma_k}\frac{\partial k}{\partial z}\right) + P_k + B_k - \varepsilon \ , \tag{3.g}$$

$$\frac{D\varepsilon}{Dt} = \frac{\partial}{\partial z}\left(\frac{\nu}{\sigma_\varepsilon}\frac{\partial \varepsilon}{\partial z}\right) + P_\varepsilon - \varepsilon_\varepsilon \ . \tag{3.h}$$

Here the production terms P_k and P_ε, the dissipation terms ε, ε_ε and the buoyancy effect B_k are defined as

$$P_k = \nu\left(\left(\frac{\partial u}{\partial z}\right)^2 + \left(\frac{\partial v}{\partial z}\right)^2\right), \quad P_\varepsilon = c_{1\varepsilon}\frac{\varepsilon}{k}P_k \ , \quad \varepsilon = c_D\frac{k^{3/2}}{L}, \quad \varepsilon_\varepsilon = c_{2\varepsilon}\frac{\varepsilon^2}{k}, \quad B_k = \frac{\nu}{\sigma_\rho}\frac{g}{\rho}\frac{\partial \rho}{\partial z} \ . \tag{4}$$

The constants are given as $c_\mu = 0.09$, $c_{1\varepsilon} = 1.44$, $c_{2\varepsilon} = 1.92$, $\sigma_k = 1.0$ and $\sigma_\varepsilon = 1.3$. These are the usual constants for a standard k-ε model. For nearly horizontal flows, as in the North Sea, horizontal velocity gradients are supposed to be much smaller than the vertical ones. This allows neglecting the horizontal production and horizontal gradient fluxes of k and ε.

Horizontal discretizations of (3) are very similar to what is used for the approximation of shallow water equations in 2 dimensions (see, e.g., Leendertse[1989], Stelling[1984]). Very important however for efficiency of the numerical model is the accuracy of the vertical discretizations. The primary aim of the vertical discretization is the limitation of the required number of layers. By application of so-called ‘compact differencing’ techniques, and by a special implementation of the k-ε model the number of layers can be kept as small as possible, often limited to a maximum of 10 to have sufficient accuracy even in case of stratified flows (see Stelling[1995]).

6. Sigma co-ordinates

In three dimensional shallow water models a sigma co-ordinate transformation is often applied (Phillips[1957]). For an elaborate discussion on the advantages and disadvantages of this approach, see Deleersnijder and Beckers[1992]. In our opinion the main advantage of this co-ordinate system is the fact that it is fitted both to the moving free surface and to the bottom topography. This is essential for the accurate approximation of the vertical flow distribution without a very large number of vertical grid points. Moreover, these ‘terrain following co-ordinates’ allow an efficient grid refinement near the free surface (in case of wind driven flow) and near the bed.

In this paper the σ-co-ordinate system is defined as

$$x^* = x, \ y^* = y, \ \sigma = \frac{z-\zeta}{H}, \ t^* = t \ , \tag{5}$$

where H = ζ+d (total water depth), $z = \zeta(x,y,t)$ or $\sigma = 0$, at the free water surface, and $z = -d(x,y)$ or $\sigma = -1$, at the bottom.

The derivatives in the original Cartesian co-ordinate system are expressed in σ co-ordinates by the chain rule

$$\begin{aligned}
&\frac{\partial}{\partial t} = \frac{\partial}{\partial t^*} + \frac{\partial\sigma}{\partial t}\frac{\partial}{\partial\sigma} \\
&\frac{\partial}{\partial x_i} = \frac{\partial}{\partial x_i^*} + \frac{\partial\sigma}{\partial x_i}\frac{\partial}{\partial\sigma}, \; i = 1{,}2 \\
&\frac{\partial}{\partial x_3} = \frac{1}{H}\frac{\partial}{\partial\sigma}
\end{aligned} \tag{6}$$

and substituted into (1).

The velocities u_i^*, i = 1,2,3 are defined by

$$u_i^* = u_i, i = 1{,}2 \quad u_3^* = \omega = H\frac{D\sigma}{Dt} \tag{7}$$

The velocities u_1^* and u_2^* remain the strictly horizontal components of the velocity vector.

For numerical approximation a grid has to be defined. Such a grid may deteriorate quite strongly in the presence of steep bottom slopes and shallow areas for example near tidal flats or the Continental Shelf edge (see Figure 1). Grids of this type cause problems when computing horizontal gradients as is recognized by several authors (e.g. Gary[1973], Janjic[1977], Mesinger and Janjic[1985], Leendertse[1990], Haney[1991], or Deleersnijder and Beckers[1992]).

For example, if we consider the transformation of the horizontal pressure gradient which reads

$$\frac{\partial p}{\partial x} = \frac{\partial p^*}{\partial x^*} + \frac{\partial\sigma}{\partial x}\frac{\partial p^*}{\partial\sigma} = \frac{\partial p^*}{\partial x^*} - \frac{1}{H}\left[\frac{\partial\zeta}{\partial x} + \sigma\frac{\partial H}{\partial x}\right]\frac{\partial p^*}{\partial\sigma}, \tag{8}$$

then near steep bottom slopes small pressure gradients might be the result of the sum of relatively large terms with opposite sign. Small truncation errors in the approximation of both terms result in a relatively large error in the pressure gradient. This might produce artificial flow. Observations of this kind have led to the notion of ‘hydrostatic consistency’ (cf., e.g. Janjic[1977]). In a notation used by Haney[1991], this consistency relation is given by

$$\left|\frac{\sigma}{H}\frac{\partial H}{\partial x}\right|\Delta x < \Delta\sigma \;, \tag{9}$$

where Δx and $\Delta\sigma$ are grid sizes.

If this relation is not satisfied then a numerical scheme might be non-convergent. However tidal flats are characterised by $H \to 0$, which means that convergence might become impossible.

There are similar difficulties for the approximation of the horizontal gradient flux for transport of matter, such as salinity and heat.

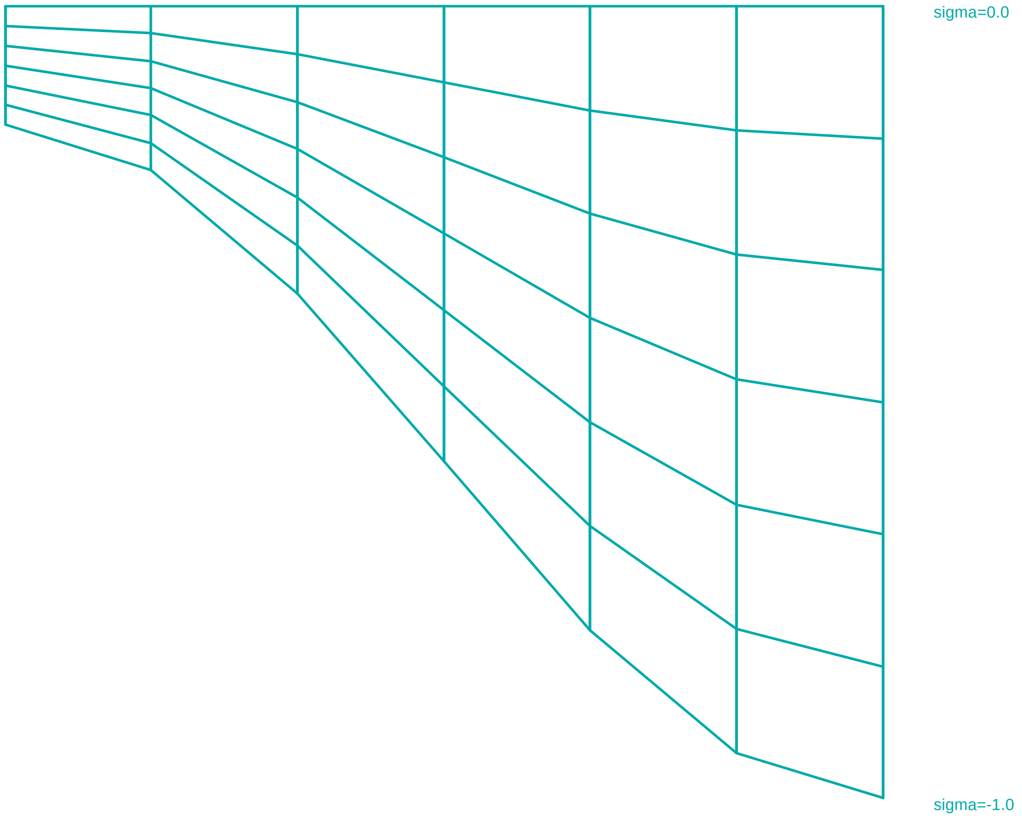

Figure 1, σ-grid

Diffusive fluxes are given by

$$F_i = D_H \frac{\partial c}{\partial x_i}, i = 1,2, F_3 = D_V \frac{\partial c}{\partial x_3} \tag{10}$$

where D_H and D_V denote the eddy diffusion coefficients in horizontal and vertical direction, respectively. The vertical eddy diffusion coefficient is determined by a turbulence closure scheme, the horizontal eddy diffusion coefficient depends on the horizontal grid resolution, and is usually an order of magnitude larger.
Transformation of the horizontal diffusion terms to x^*, y^*, σ, t^* is a tedious task, since applying the chain rule leads to various cross derivatives. For example, the transformation of a simple second order derivative leads to

$$\frac{\partial^2 c}{\partial x^2} = \frac{\partial^2 c^*}{\partial x^{*2}} + \left(\frac{\partial\sigma}{\partial x}\right)^2 \frac{\partial^2 c^*}{\partial\sigma^2} + 2\frac{\partial\sigma}{\partial x}\frac{\partial^2 c^*}{\partial x^*\partial\sigma} + \frac{\partial^2\sigma}{\partial x^2}\frac{\partial c^*}{\partial\sigma} \; . \tag{11}$$

For such a combination of terms it is difficult to find a numerical approximation that is stable and positive. Near steep bottom slopes or near tidal flats where the total depth becomes very small, truncations errors in the approximation of the horizontal diffusive fluxes in σ-co-ordinates are likely to become very large, similar to the horizontal pressure gradient. Some authors (e.g., Mellor and Blumberg[1985]), omit several terms of the transformation, which yields the following diffusive fluxes

$$F_1^* = D_H \frac{\partial c^*}{\partial x^*}$$

$$F_2^* = D_H \frac{\partial c^*}{\partial y^*}$$

$$F_3^* = \frac{D_V}{H} \frac{\partial c^*}{\partial \sigma} \tag{12}$$

The physical conditions (upwelling) which cause this new formulation to give a better description of the transport process, are certainly not fulfilled in many estuaries. If we omit vertical diffusion, then this new formulation will still cause some numerical vertical diffusion, especially near steep bottom slopes such as near tidal flats. Due to this phenomenon it will be difficult to simulate the density stratification due to temperature or salinity near steep bottom slopes. Hence, the complete transformation must be included. But in that case numerical problems are encountered concerning accuracy, stability and monotonicity. In Stelling and Van Kester[1994], a method is introduced which gives a consistent, stable and monotonic approximation of the horizontal diffusion terms and baroclinic pressure, even when the hydrostatic consistency condition is violated. The method is based upon a Finite Volume Method.

7. Conclusions and recommendations

The experiences with High Performance Computing techniques lead to the following conclusions and recommendations:

- The availability of parallel computers with distributed memory offers interesting possibilities for large scale simulation. Although the development of codes for these machines is much more difficult and time-consuming than for sequential machines and for vector processors, the usage of parallel computers alleviates the computational burden quite a bit, at the same time increasing memory resources. As a result, it opens the gate to e.g. larger models, finer resolutions and modelling of more physical phenomena. Models defined as type B in this report are realistic to use also for long term simulations.
- The numerical algorithms employed in existing models (e.g., ADI time integration) are often hard to parallelize. Therefore, new generation flux models should make use of numerical algorithms which are not only chosen for mathematical properties like robustness, stability, efficiency, and accuracy but also for high performance computing properties like parallelizability and vectorizability. Examples are the domain decomposition methods and the time integration methods for bio-chemical transport, as developed at CWI. With such newly developed numerical algorithms significantly faster simulation is indeed possible as experiments at CWI, KUL, TUD, and DH show.
- It is important to realize that besides increasing the parallel properties of numerical algorithms, effort must also be put in improving the mathematical and numerical properties of algorithms. As an example, the execution time of most numerical methods increases faster than linear with the number of grid points. Models of fine(r) resolution thus may give rise to a dramatic increase in the execution time. Hence, there is a need to develop numerical schemes that come close to a computational complexity that is linear in the number of grid points.
- Some other topics that require further research are:

  - the usage of locally refined grids. In particular the question should be answered how to deal with (internal) boundaries, where grids of different resolution meet;
  - drying-and-flooding procedures;
  - the development of algorithms based on a finite element discretization, which allows to use irregular grids.
- The current physical models need to be improved in order to allow more accurate simulations. Bottlenecks of the current models are for instance the usage of imprecise boundary conditions (more accurate information about what happens at the edge of the Continental Shelf is needed) and the imprecise modelling of the forcing terms (usage of averaged wind fields, because of the lack of information about local wind fields). Also proper modelling of horizontal exchange of momentum for L.E.S. is an important research field.
- The most complex numerical problems at present deal with the correct representation of vertical structure resulting from density stratification. Much research on this topic is needed to obtain efficient flux models of the Continental Shelf.

References

E. Deleersnijder and J.M. Beckers (1992),
On the use of the σ co-ordinate system in regions of large bathymetric variations, Journal of Marine Systems, 3, pp. 381-390.

C. Eichler-Liebenow, P.J. van der Houwen and B.P. Sommeijer (1997),
Analysis of approximate factorization in iteration methods, Report MAS-R9718, CWI, Amsterdam, to appear in Appl. Numer. Math., 1998.

M. Gary (1973),
Estimate of Truncation Error in Transformed Primitive Equation, Atmospheric Models, 30, pp. 223-233.

E.D. de Goede, J. Groeneweg, K.H. Tan, M.J.A. Borsboom and G.S. Stelling (1995),
A domain decomposition method for the three-dimensional shallow water equations, Simulation Practice and Theory, 3, pp. 307-325.

R.L. Haney (1991),
On the pressure gradient force over steep topography in sigma coordinate ocean models, Journal of Physical Oceanography, 21, pp. 610-619.

P.J. van der Houwen and B.P. Sommeijer (1997),
Splitting methods for three-dimensional transport models with interaction terms, J. of Scientific Computing, 12, pp. 215-231.

P.J. van der Houwen, B.P. Sommeijer and J. Kok (1997),
The iterative solution of fully implicit discretizations of three-dimensional transport models, Appl. Numer. Math., 25, pp. 243-256.

Z.I. Janjic (1977),
Pressure gradient force and advection scheme used for forecasting with steep and small scale topography, Beiträge zur Physik der Atmosphäre, 50, pp. 186-199.

J.J. Leendertse (1989),
A new approach to three-dimensional free-surface flow modelling, R-3712-NETH/RC, The Rand Corporation (Santa Monica).

J.J. Leendertse (1990),
Turbulence modelling of surface water flow and transport: Part IIIa, J. Hydraulic Eng., 116, pp. 600-607.

G.L. Mellor and A.F. Blumberg (1985),
Modelling vertical and horizontal diffusivities with the sigma coordinate system, Monthly Weather Rev., 113, pp. 1379-1383.

F. Mesinger and Z.I. Janjic (1985),
Problems and numerical methods in the incorporation of mountains in atmospheric models, In: Lectures in applied mathematics, 22, pp. 81-120.

N.A. Phillips (1957),
A co-ordinate system having some special advantages for numerical forecasting, J. Meteor., 14, pp. 184-185.

M.R.T. Roest, E.A.H. Vollebregt, H.H. ten Cate and H.X. Lin (1996),
Design of a parallel flow simulation package and its performance on a cluster of workstations, In: E.J.H. Kerckhoffs et al., (Ed.), Proc. 2nd Annual Conference of the ASCI research school, Lommel, Belgium, pp. 33-38.

B.P. Sommeijer and J. Kok (1996),
Splitting methods for three-dimensional bio-chemical transport, Appl. Numer. Math., 21, pp. 303-320.

Z.W. Song, D. Roose and J. Berlamont (1995),
Investigations of different time integration schemes for the 2D shallow water equations on distributed memory parallel computers, J. Simulation Practice and Theory, 3, pp. 287-305.

G.S. Stelling (1984),
On the construction of computational methods for the shallow water equations, Rijkswaterstaat communication nr. 35.

G.S. Stelling and J.A.Th.M. van Kester (1994)
On the approximation of horizontal gradients in sigma transformed bathymetries with steep bottom slopes, Numer. Meth. Fluids, 18, pp. 915-935.

G.S. Stelling (1995),
Compact differencing for stratified free surface flows, In: Int. Conf. on Advances in Hydro-Science and Engineering, Beijing, China, Vol. I, part A, pp. 378-386.

E.A.H. Vollebregt et al. (1996),
Possibilities and implications of HPC to 3D flow and transport simulations on the North West Continental Shelf Seas, In: Proc. Intern. EUROSIM Conf. HPCN challenges in telecomp and telecom: parallel simulation of complex systems and large-scale applications, Delft, the Netherlands, Elsevier.