INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2009; **59**:1157–1179
Published online 5 August 2008 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1859

Parallel implementation of a non-hydrostatic model for free surface flows with semi-Lagrangian advection treatment

Jacek A. Jankowski*, †

Department of Hydraulic Engineering in Inland Areas, Federal Waterways Engineering and Research Institute (BAW), Kussmaulstr. 17, Karlsruhe 76187, Germany

SUMMARY

The parallel implementation of an unstructured-grid, three-dimensional, semi-implicit finite difference and finite volume model for the free surface Navier–Stokes equations (UnTRIM) is presented and discussed. The new developments are aimed to make the code available for high-performance computing in order to address larger, complex problems in environmental free surface flows. The parallelization is based on the mesh partitioning method and message passing and has been achieved without negatively affecting any of the advantageous properties of the serial code, such as its robustness, accuracy and efficiency. The key issue is a new, autonomous parallel streamline backtracking algorithm, which allows using semi-Lagrangian methods in decomposed meshes without compromising the scalability of the code. The implementation has been carefully verified not only with simple, abstract test cases illustrating the application domain of the code but also with advanced, high-resolution models presently applied for research and engineering projects. The scheme performance and accuracy aspects are researched and discussed. Copyright © 2008 John Wiley & Sons, Ltd.

Received 28 January 2008; Revised 7 April 2008; Accepted 4 May 2008

KEY WORDS: parallel; free surface; non-hydrostatic; semi-Lagrangian; unstructured grid; UnTRIM

1. INTRODUCTION

High-performance computing for modelling of environmental flows is often regarded as a highly specialized domain associated with large supercomputer facilities and requires specific technical know-how. The appearance of widely available multi-core processors changes this situation, making parallel computing readily affordable and awaking a broader interest in the high-performance computing issues. In order to profit from computational resources delivered by modern parallel computers of any kind or size, an additional, specific effort is required from the code developers. Unfortunately, presently only in exceptional cases hydrodynamic codes are designed as parallel

*Correspondence to: Jacek A. Jankowski, Bundesanstalt für Wasserbau, Postfach 21 02 53, Karlsruhe 76152, Germany.
†E-mail: jacek.jankowski@baw.de

programs from scratch (e.g. [1, 2]) and normally a serial code must be adapted for the parallel execution. This may often result in solving non-trivial problems due to the properties of the numerical formulation, which are adverse for the parallel execution. The semi-Lagrangian advection treatment has been an example of a numerical algorithm whose parallel implementations have been connected with some limitations compared with their serial, original versions [3, 4].

The demand that the parallel codes are able to scale with the growing problem size is accompanied by the requirement that the numerical schemes they are based on have advantageous, well-researched mathematical properties allowing the correct assessment of the application domain of the numerical model. In this situation, the attention of researchers and engineers working with numerical models for studying hydraulic environmental problems is attracted by simpler but general-purpose codes, which not only cover the aimed application domain but also guarantee efficiency, robustness and accuracy of the numerical scheme.

Since 1990 the TRIM family of finite difference models for structured meshes has been systematically developed by Casulli and his co-workers [5, 6], with a special attention not only to the numerical properties of the scheme [7], but also to its practical applicability [8]. All these features are inherited by UnTRIM, the extension of the final TRIM algorithm [9] for unstructured meshes in order to deal with complex boundaries more flexibly and allow for local mesh refinements [10–12].

In the shortest definition, UnTRIM is a practically oriented scheme for solving the three-dimensional (3D) Navier–Stokes equations with a semi-implicit, fractional time-step integration, a finite difference/volume spatial discretization and a semi-Lagrangian treatment of advection using an unstructured, staggered, orthogonal, z-level grid. The aimed application domains are 3D, non-hydrostatic, environmental free surface flows including species transport. The mesh consists of horizontal layers of prismatic cells with horizontal triangular or quadrilateral bases. In the particular case, when only one layer is defined, the algorithm is consistent with the two-dimensional (2D), vertically integrated shallow water equations. Drying and flooding of computational mesh cells is also included in a natural way.

The governing momentum equations together with the equation for obtaining the free surface position and the incompressibility condition are treated by this relatively simple and efficient algorithm in such a way that the numerical solution is stable with respect to the gravity waves speed, bottom and free surface friction as well as vertical viscosity. The presence of the horizontal viscosity implies a mild stability criterion treated iteratively when necessary. For modelling of internal waves, their speed is also a limiting factor for the time step. In the case of barotropic flows, when the horizontal viscosity is neglected, the scheme is unconditionally stable. Local and global volume conservation is assured by a finite volume formulation for the incompressibility condition and the free surface equation. The hydrodynamic algorithm is accompanied with an algorithm for the advection–diffusion equation describing the species transport. It is globally and locally mass conservative, satisfies a discrete minimum–maximum principle and applies flux limiters for a higher resolution [13].

For regular meshes the discretization error is second order in space, as well as in time when the semi-implicit scheme is equally balanced between time levels. For irregular meshes or fully implicit computations, the discretization error grows, but can be diminished when mesh polygon dimensions vary gradually in the domain. The highest accuracy for species transport can be obtained when the grid is co-linear with the expected net flow streamlines.

The orthogonal, staggered mesh is beneficial with respect to the overall stability of the method compared, e.g. with centroid-based schemes [14, 15], which justifies a larger mesh generating effort for orthogonal meshes [16]. Additionally, the semi-Lagrangian momentum advection scheme of

DOI: 10.1002/fld

UnTRIM has clear advantages compared with Eulerian methods affected by time step and spatial discretization limitations, especially when wetting and drying occur [1, 17, 18].

The described properties indicate UnTRIM as a general-purpose code applicable for both 2D, depth-averaged, and 3D hydrostatic or non-hydrostatic flows. The UnTRIM software structure with an efficient numerical kernel accessible through a rich user interface allows all steps of creative, practical code usage from simplest, non-generic user solutions up to embedding in complex modular software systems with, e.g. wave and morphodynamic models coupling.

It is noteworthy that the attractive properties of the Casulli formulation have motivated or influenced numerous similar code efforts, whereby SUNTANS [17], DELFIN [14], FINEL [19] or ELCIRC [20] can be named as examples without making any claims for completeness.

A parallel version of UnTRIM with the shared memory parallelization paradigm using OpenMP [21] standards exists since 2002. However, even when applying state-of-the-art massive parallel computers the maximum speedup factor of approx. 15 cannot be exceeded due to the limitations imposed by these parts of the code, which have to be executed serially (Amdahl's Law). Although the presently reached affordability of multi-core processors makes the further OpenMP version maintenance attractive, a better scalability of the code can be achieved using mesh partitioning methods and message-passing parallelization standards, such as MPI [22].

The domain decomposition method—understood in the context that large problems are divided into smaller ones through the introduction of usually artificial sub-domains of the geometrical domain—is an efficient means of introducing natural parallelism into a problem, which is considered computationally too intensive for a single processor. It is one of the most successfully applied approaches to achieve a good parallel speedup independently from the fact if some internal parallel properties of the algorithm itself are present or not. For hydrodynamic codes based on the finite elements, volumes or differences methods, it is almost synonymic with the partitioning of the computational mesh. A separate copy of the program is executed on each processor with data concerning a partition of the meshed domain, exchanging information between these sub-domains when necessary. The speedup factor then depends strictly on the communication efficiency with a growing number of applied processors.

Reaching a good scalability of the parallel code for computationally intensive applications without negatively affecting the advantageous features of the serial numerical scheme of UnTRIM is the main aim of this study. The main difficulty approached and solved without introducing any limitations to the parallel code is the semi-Lagrangian treatment of the advection based on the autonomous streamline tracking over the partitioned mesh.

This paper is structured as follows. In Section 2 the fundamental governing equations are provided for a reader unfamiliar with the UnTRIM conceptual model. The parallel implementation is described in detail in Sections 3 and 4, in which also the mesh and the numerical scheme of UnTRIM are presented so far as required for the understanding of the parallelization methodology. The new parallel semi-Lagrangian advection scheme is presented in Section 4.4. The results concerning in this case the verification of the accuracy and performance of the scheme are presented and discussed in Section 5. A summary, outlook and conclusions are provided in Sections 6 and 7.

2. GOVERNING EQUATIONS

The application domains of UnTRIM are geophysical, environmental free-surface water flows in moderate spatial and temporal scales, e.g. rivers, estuaries, lakes, smaller seas, etc. Therefore, the

DOI: 10.1002/fld

momentum conservation is described adequately by the incompressible, Reynolds-averaged 3D Navier–Stokes equations with the eddy viscosity concept, Boussinesq approximation and Coriolis parameters formulated for a Cartesian coordinate system. The continuity equation is taken in the form typical for the environmental free surface flows. The bottom and wind friction as well as the influence of atmospheric pressure are parametrized. Transport equations for scalars (species) such as temperature, salinity, suspended (and settling, such as sediments) or passively transported dissolved matter are solved as well. (In the framework of this paper the interaction between fluid and the bottom sediments causing morphodynamic changes is not taken into account.)

The model yields results in the form of transient variables: the free surface elevation $\eta(x, y, t)$, velocity components $u(x, y, z, t,)$, $v(x, y, z, t)$ and $w(x, y, z, t)$ in the horizontal directions x, y and the vertical z-direction, respectively (t is the time). Additionally, the normalized pressure $p(x, y, z, t)$ (i.e. pressure divided by the constant Boussinesq reference density ϱ_0) and the species concentrations $c(x, y, z, t)$ are delivered.

Denoting the variable water density as ϱ, the constant Coriolis parameter as f and the gravitational acceleration aligned with the z-direction as g, the momentum conservation equations can be expressed as follows:

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+w\frac{\partial u}{\partial z}-fv=-\frac{\partial p}{\partial x}+\nu^{\mathrm{h}}\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)+\frac{\partial}{\partial z}\left(\nu^{\mathrm{v}}\frac{\partial u}{\partial z}\right) \tag{1}$$

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+w\frac{\partial v}{\partial z}+fu=-\frac{\partial p}{\partial y}+\nu^{\mathrm{h}}\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right)+\frac{\partial}{\partial z}\left(\nu^{\mathrm{v}}\frac{\partial v}{\partial z}\right) \tag{2}$$

$$\frac{\partial w}{\partial t}+u\frac{\partial w}{\partial x}+v\frac{\partial w}{\partial y}+w\frac{\partial w}{\partial z}=-\frac{\partial p}{\partial z}+\nu^{\mathrm{h}}\left(\frac{\partial^2 w}{\partial x^2}+\frac{\partial^2 w}{\partial y^2}\right)+\frac{\partial}{\partial z}\left(\nu^{\mathrm{v}}\frac{\partial w}{\partial z}\right)-\frac{\varrho}{\varrho_0}g \tag{3}$$

The Boussinesq eddy viscosity components in the horizontal and vertical directions ν^{h} and ν^{v} can be delivered by an appropriate turbulence closure model as non-negative functions of space and time.

The volume conservation condition for incompressible flows states that the velocity divergence is equal to zero:

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}=0 \tag{4}$$

Making an important assumption for the conceptual model that the water elevation $\eta(x, y, t)$ is a single-valued function with respect to the vertical direction z (*height function method*) and using the kinematic condition at the free surface and the impermeability condition at the bottom one obtains the conservative form of the continuity equation for the elevation η:

$$\frac{\partial \eta}{\partial t}+\frac{\partial}{\partial x}\left[\int_{-h}^{\eta} u\,\mathrm{d}z\right]+\frac{\partial}{\partial y}\left[\int_{-h}^{\eta} v\,\mathrm{d}z\right]=0 \tag{5}$$

where $h(x, y)$ is the bathymetry measured from a reference water level, assumed constant in time in the framework of this paper. The bathymetry h is counted from the reference level 0 downwards, the free surface elevation η upwards. The total water depth is then $H(x, y, t)=h(x, y)+\eta(x, y, t)$.

DOI: 10.1002/fld

The characteristic feature of this model is the pressure treatment in Equations (1)–(3). It is decomposed into a sum of its hydrostatic and dynamic (non-hydrostatic) components. Indeed, the (global) pressure $p(x, y, z, t)$ can be expressed as the sum of atmospheric pressure at the free surface $p_a(x, y, t)$, the barotropic and the baroclinic contributions to the hydrostatic pressure and the dynamic pressure $q(x, y, z, t)$:

$$p(x, y, z, t) = p_a(x, y, t) + g[\eta(x, y, t) - z] + g \int_z^\eta \frac{\varrho(x, y, z, t) - \varrho_0}{\varrho_0} \, d\zeta + q(x, y, z, t) \tag{6}$$

Introducing (6) in the momentum equations (1)–(3), one obtains the form of the momentum equations eventually applied in the scheme:

$$\frac{du}{dt} - fv = -\frac{\partial p_a}{\partial x} - g\frac{\partial \eta}{\partial x} - g\frac{\partial}{\partial x}\left[\int_z^\eta \frac{\varrho - \varrho_0}{\varrho_0} \, d\zeta\right] - \frac{\partial q}{\partial x} + \nu^h \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + \frac{\partial}{\partial z}\left(\nu^v \frac{\partial u}{\partial z}\right) \tag{7}$$

$$\frac{dv}{dt} + fu = -\frac{\partial p_a}{\partial y} - g\frac{\partial \eta}{\partial y} - g\frac{\partial}{\partial y}\left[\int_z^\eta \frac{\varrho - \varrho_0}{\varrho_0} \, d\zeta\right] - \frac{\partial q}{\partial y} + \nu^h \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) + \frac{\partial}{\partial z}\left(\nu^v \frac{\partial v}{\partial z}\right) \tag{8}$$

$$\frac{dw}{dt} = -\frac{\partial q}{\partial z} + \nu^h \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2}\right) + \frac{\partial}{\partial z}\left(\nu^v \frac{\partial w}{\partial z}\right) \tag{9}$$

If the conditions for a hydrostatic approximation of the pressure apply, which is equivalent to the assumption that the vertical accelerations are negligible, Equation (9) can be neglected and the dynamic pressure vanishes ($q = 0$). In this case the vertical velocity component can be obtained from the incompressibility condition (4).

The shear boundary conditions for the momentum equations (7)–(9) at the bottom of the water body are specified as

$$\nu^v \frac{\partial u}{\partial z} = \gamma_B u, \quad \nu^v \frac{\partial v}{\partial z} = \gamma_B v \quad \text{at } z = -h \tag{10}$$

where γ_B is defined according to

$$\gamma_B = r_B \sqrt{u^2 + v^2} \tag{11}$$

where r_B is a non-negative bottom friction coefficient. At the free surface, the wind stresses can be prescribed as

$$\nu^v \frac{\partial u}{\partial z} = \gamma_T (u_a - u), \quad \nu^v \frac{\partial v}{\partial z} = \gamma_T (v_a - v) \quad \text{at } z = \eta \tag{12}$$

where u_a and v_a are the wind velocity components in the x- and y-directions, respectively. The coefficient γ_T is defined as

$$\gamma_T = r_T \sqrt{(u_a - u)^2 + (v_a - v)^2} \tag{13}$$

where r_T is a non-negative wind stress coefficient.

At open inflow boundaries Dirichlet boundary conditions for the velocity can be applied, also in the form of a given fluid volume flux. For open outflow boundaries, the radiation condition is

DOI: 10.1002/fld

applied:

$$\frac{\partial \eta}{\partial t}+\frac{\partial}{\partial x}\left[\int_{-h}^{\eta} u\,\mathrm{d}z\right]+\frac{\partial}{\partial y}\left[\int_{-h}^{\eta} v\,\mathrm{d}z\right]=\frac{\eta^*-\eta}{\tau_\mathrm{r}} \tag{14}$$

where $\eta^*(t)$ is the prescribed elevation and τ_r is a problem-specific relaxation time.

The mass conservation of scalars transported with the fluid (species) is expressed in the form of the transport equations for their concentration $c(x, y, z, t)$:

$$\frac{\partial c}{\partial t}+\frac{\partial (uc)}{\partial x}+\frac{\partial (vc)}{\partial y}+\frac{\partial [(w-w_\mathrm{s})c]}{\partial z}=\frac{\partial}{\partial x}\left(K^\mathrm{h}\frac{\partial c}{\partial x}\right)+\frac{\partial}{\partial y}\left(K^\mathrm{h}\frac{\partial c}{\partial y}\right)+\frac{\partial}{\partial z}\left(K^\mathrm{v}\frac{\partial c}{\partial z}\right) \tag{15}$$

The typical species are salinity, temperature, as well as suspended sediment with a non-zero settling velocity of w_s. Any passive substance transported with the fluid can also be treated in this manner. The horizontal and vertical diffusivity coefficients (K^h and K^v) can be prescribed as non-negative functions of time and space.

The boundary conditions for the scalar transport equation at the bottom and the free surface are specified in the form of mass fluxes:

$$-K^\mathrm{v}\frac{\partial c}{\partial z}+(w-w_\mathrm{s})c=\alpha_\mathrm{B}+\beta_\mathrm{B}(c_\mathrm{B}-c) \quad \text{at } z=-h \tag{16}$$

$$K^\mathrm{v}\frac{\partial c}{\partial z}-(w-w_\mathrm{s})c=\alpha_T+\beta_T(c_T-c) \quad \text{at } z=\eta \tag{17}$$

where α_B, β_B and α_T, β_T are prescribed non-negative parameters, independent from the concentrations c and defined at the bottom and the surface. c_B and c_T are the prescribed bottom and free surface concentrations. At the lateral open boundaries, species concentrations can be imposed.

The equation system is closed by an equation of state, which relates the water density ϱ to the concentration of each scalar variable c, which can influence the density (such as temperature and/or salinity):

$$\varrho(x, y, z, t)=\varrho(c(x, y, z, t)) \tag{18}$$

3. THE MESH

3.1. The unstructured orthogonal mesh

The mesh structure applied in UnTRIM reflects the assumptions met in the conceptual model and in the governing equations for environmental flows, however, without putting a strain on the requirements for the numerical scheme. The geometrical aspects to be considered are (1) the height function method for obtaining the water elevation $\eta(x, y, t)$ with respect to the vertical direction, (2) the typical properties of natural water bodies in which the water depth is mostly comparable or smaller than their horizontal extend, (3) the necessity to deal with complex horizontal boundaries which change their position due to water level changes (drying and wetting, i.e. tidal flats) as well as (4) arbitrary bathymetries in terms of bottom gradients. Therefore, a mesh that is unstructured in the horizontal direction, but structured in the vertical direction, with strictly horizontal layers of not necessarily constant thicknesses, is a good choice for most applications.

DOI: 10.1002/fld

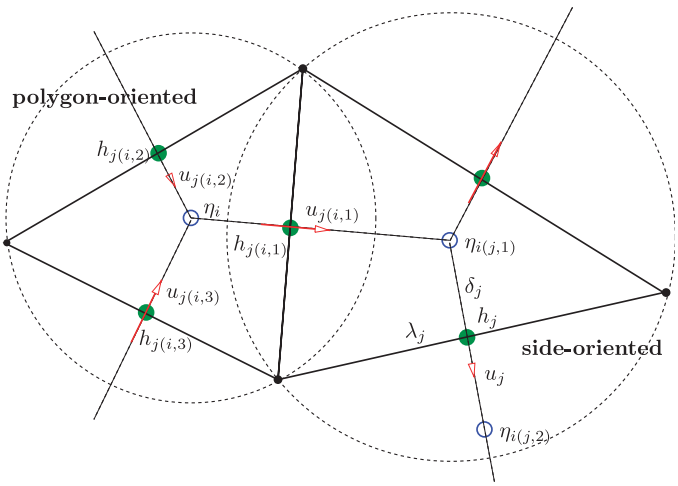

Figure 1. The horizontal staggered mesh. Although the water elevation η_i is defined in the middle of a polygon of a surface P_i, the normal velocity component u_j and the bathymetry h_j are defined on the polygon sides of length λ_j and segments between circumcentres δ_j orthogonal to them. All variables can be addressed using polygon-oriented (left) and side-oriented (right) connectivity tables.

The orthogonal mesh of UnTRIM consists of horizontal layers of prismatic cells with horizontal triangular or quadrilateral bases. The horizontal (i.e. covering the (x, y)-domain) unstructured orthogonal grid has the important property that the segments δ_j $(j=1,2,\ldots,N_s)$ joining the circumcentres of the mesh polygons P_i with S_i sides $(i=1,2,\ldots,N_p)$ must have an intersection with their common sides λ_j and be orthogonal to each other. The resulting horizontal mesh can be treated as a combination of a Delaunay tessellation with edges λ_j and its dual, a Voronoi diagram with circumcentre distances δ_j (Figure 1).

In the vertical direction (Figure 2) the mesh consists of horizontal cell layers k so that the distances between the mesh surfaces identified by $z_{k\pm1/2}$ are given by $\Delta z_k = z_{k+1/2} - z_{k-1/2}$ $(k=1,2,\ldots,N_z)$. In this manner the 3D mesh consists of prism layers of a constant height Δz_k, with horizontal faces formed by the orthogonal grid polygons P_i.

The discrete variables of the scheme are defined at staggered positions in the mesh and assumed constant over an object they are ordered to (Figure 1). The bathymetry h_j is defined constant over a polygon side j while the elevation of the free surface η_i^n for the nth time step is situated at the circumcentre of the polygon P_i and constant over its surface. The horizontal velocity $u_{j,k}^n$, treated in the scheme in the normal direction to each prism face, is defined at the intersections of the prism centres and their common faces, while the vertical velocity $w_{i,k+1/2}^n$ is defined at the circumcentres of the prism top and bottom surfaces (Figure 2). Finally, other variables such as hydrodynamic pressure $q_{i,k}^n$ and m species concentrations $c_{i,k,m}^n$ are defined in the centres of the prisms.

The addressing of variables defined on various points in the mesh is realized with connectivity tables for N_s sides and N_p polygons. The edges of the polygons can be accessed with an index $j(i,l)$, $1 \leqslant j(i,l) \leqslant N_s$, $l=1,2,\ldots,S_i$, and both polygons sharing the j-side with $i(j,s)$, $1 \leqslant i(j,s) \leqslant N_p$, where $s=1,2$ (Figure 1). The 3D connectivity tables follow the 2D pattern and

DOI: 10.1002/fld

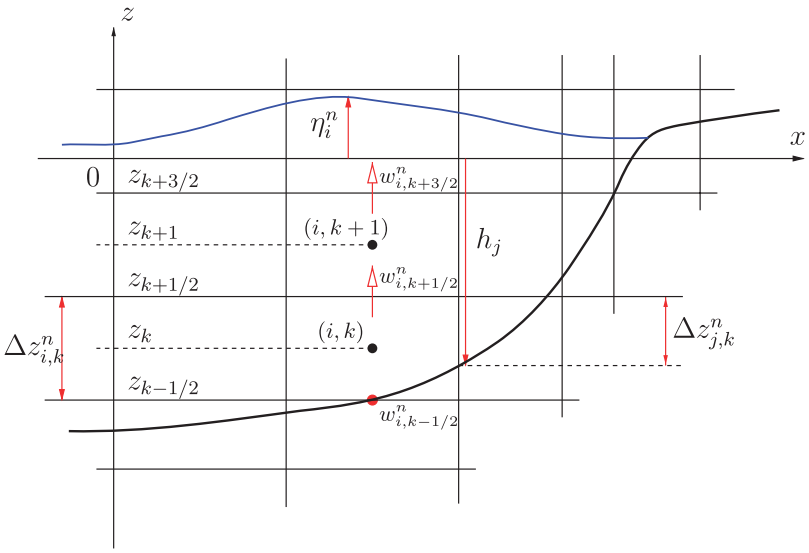

Figure 2. The vertical mesh structure. The bathymetry h is counted from the reference level downwards, and the free surface elevation η^n upwards. The dynamic pressure q^n and species concentrations c^n are defined in the cell centres (i,k) at levels z_k. The vertical velocity w^n is defined on the polygon upper and lower surfaces at the levels $z_{k\pm1/2}$.

the cells or sides in the water columns. This indirect addressing is a characteristic feature of unstructured meshes important for the parallel implementation.

3.2. Mesh partitioning

The description above suggests that in order to apply the domain decomposition method efficiently, mesh partitioning into horizontally balanced, static sub-domains is appropriate. The staggered positions of the discrete variables in the finite difference/volume scheme force an overlapping of these *mesh partitions*, using *halo cells* along vertical *interfaces* between partitions; see Figure 3.

The mesh decomposition is not a trivial task, taking into account that it is usually required that the resulting static mesh partitions should be equally large in terms of cell numbers taken into account in the computation (thus balancing the computational load on each processor) and the sub-domain interfaces possibly small in terms of neighbouring cell number (thus reducing the amount of data to be exchanged between the processors).

However, with this approach and for typical geophysical applications some imbalance in the computational effort per processor can occur. In vertical, the number of cells is variable due to the water levels varying in time, which is taken into account by varying the indices in loops sweeping vertical directions in the UnTRIM code. Furthermore, the movements of the free surface can result in drying and wetting of cells during the simulation, which influences the iterative equation solvers for the elevation and pressure as well.

The most efficient and available partitioning methods for non-trivial cases are based on the graph partitioning theory. In the parallel implementation described here, the mesh partitioning software for UnTRIM is based on the *Metis* and *ParMetis* libraries [23]. Although the software

DOI: 10.1002/fld

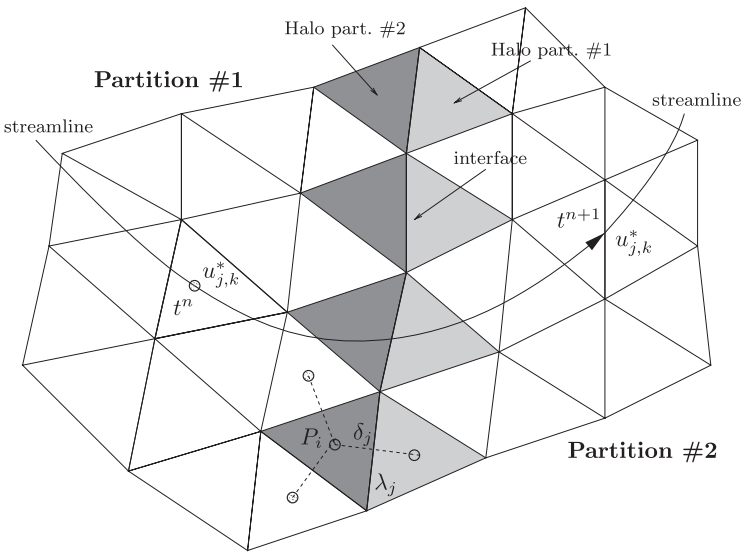

Figure 3. *Partitioned mesh*: to compute values in the internal halo cell P_i, values are required from the external halo cell behind the interface side λ_j crossed by the segment δ_j joining cell circumcentres. *Streamline tracking*: to obtain the next time step t^{n+1} advected value $u^*_{j,k}$, the streamline is followed one time step backward in time and $u^*_{j,k}$ is interpolated in the neighbour partition cell, where the traceback ends for t^n.

allows weighting the partition size according to arbitrary criteria, at this development stage the equal balancing of the partitions in terms of polygon numbers is applied in the verification test cases (Section 5).

4. UnTRIM PARALLEL IMPLEMENTATION

4.1. Numerical scheme outline

In this section the numerical scheme of UnTRIM is analysed in terms of the parallel implementation using the mesh partitioning method. We concentrate exclusively on the characteristic features of the numerical formulation relevant for the message-passing parallelization. For a complete and detailed reference, the publication of Casulli and Zanolli [11] or Casulli and Lang [24] is recommended.

UnTRIM algorithm applies a semi-implicit, fractional time-step integration scheme. In the first step, the provisional values for the velocity and the free surface elevation are obtained from Equations (7)–(8) and (5), where both variables are treated semi-implicitly (θ-method). For stability reasons, the bottom and free surface friction (10)–(12) as well as the vertical viscosity are treated implicitly. In the second fractional step, the velocity and the elevation are corrected by taking the dynamic pressure q into account so that the resulting velocity field is solenoidal and thus volume conservative.

DOI: 10.1002/fld

4.2. Halo swapping

An insight into the communication patterns required in the mesh partitioning method can be given studying exemplarily the first part of the algorithm. In this semi-implicit fractional step (θ-method) the provisional horizontal velocity field $\tilde{u}$ in the new time step n is computed taking the free surface η and dynamic pressure component q gradients as well as the vertical viscosity v^{v} from the horizontal momentum equations (7)–(8):

$$\tilde{u}_{j,k}^{n+1} = \mathbf{F}u_{j,k}^{n} - (1-\theta)\frac{\Delta t}{\delta_j}[g(\eta_{i(j,2)}^{n} - \eta_{i(j,1)}^{n}) + q_{i(j,2),k}^{n} - q_{i(j,1),k}^{n}] - \theta g \frac{\Delta t}{\delta_j}(\tilde{\eta}_{i(j,2)}^{n+1} - \tilde{\eta}_{i(j,1)}^{n+1}) + \frac{\Delta t}{\Delta z_{j,k}^{n}}\left[v_{j,k+1/2}^{\mathrm{v}} \frac{\tilde{u}_{j,k+1}^{n+1} - \tilde{u}_{j,k}^{n+1}}{\Delta z_{j,k+1/2}^{n}} - v_{j,k-1/2}^{\mathrm{v}} \frac{\tilde{u}_{j,k}^{n+1} - \tilde{u}_{j,k-1}^{n+1}}{\Delta z_{j,k-1/2}^{n}} \right], \quad k = m_j^n, m_{j+1}^n, \ldots, M_j^n \tag{19}$$

where $\mathbf{F}$ is an explicit finite difference operator:

$$\mathbf{F}u_{j,k}^{n} = \frac{[1-\theta(1-\theta)f^2\Delta t^2]u_{j,k}^{*} + f\Delta t v_{j,k}^{*}}{1+\theta^2 f^2 \Delta t^2} + \Delta t v^{\mathrm{h}} \Delta_h u_{j,k}^{*} - \Delta t \frac{p_{a,i(j,2)}^{n+\theta} - p_{a,i(j,1)}^{n+\theta}}{\delta_j} - g\frac{\Delta t}{\delta_j \varrho_0} \sum_{l=k}^{M_j} \omega_l [\varrho_{i(j,2)}^{n} - \varrho_{i(j,1)}^{n}] \Delta z_{j,l}^{n} \tag{20}$$

where $\omega_k = \frac{1}{2}$ and $\omega_l = 1,\ l = k+1, k+2, \ldots, M_j$. In order to achieve stability of the scheme, the implicitness factor θ must be in the range $0.5 \leqslant \theta \leqslant 1$. The range of indices k for the vertical finite difference stencils (from m_j to M_j^n, $1 \leqslant m_j \leqslant M_j^n \leqslant N_z$) changes due to the free surface dynamics. It must be stressed that in formulating (19)–(20) one profits from the fact that the momentum equations are invariant to the solid body rotation in the (x, y)-plane so that $u_{j,k}$ denotes the horizontal velocity component normal to the side j at the level k. The horizontal finite difference stencils, based on the circumcentre distances δ_j, make an intensive use of connectivity tables like $i(j, s)$.

The parallel treatment of the explicit operator $\mathbf{F}$ in Equation (20) is described in Sections 4.3.1 and 4.4. Concentrating on Equation (19), it can be noted that on a partitioned mesh, all terms concerning the *interface sides* j at a level k can be computed locally in a mesh partition if it is extended with *external halo cells* belonging to the neighbour partition and the values defined there are made available by communication between partitions. The same concerns the variables defined for central points i of *internal halo cells* being neighbours to the vertical partition interface; see Figure 3. A noteworthy fact is that the overlapping between the horizontal partitions can be limited only to the halo cells directly adjacent to the interface sides.

Therefore, in the parallel software implementation the external halo objects (polygons, edges, cells, sides) are taken into account in the object lists, sorted contiguously and included in the local connectivity tables for partitions at distinguishable positions. Values of variables for these external halo objects are computed in the neighbouring partitions (where they are internal cells, actually) and fetched via message-passing communication if necessary. In this manner all dependencies on the values addressed via connectivity tables from the horizontally oriented computational loops over internal cell or side indices are fulfilled. A characteristic feature of this approach is the fact that the *common interface sides* are treated as internal sides from the point of view of both

DOI: 10.1002/fld

the neighbouring sub-domains and the velocity is computed independently in these sub-domains without exchanging any values via communication.

A similar analysis can be done for the vertical momentum equation as well:

$$\tilde{w}_{i,k+1/2}^{n+1} = \mathbf{F} w_{i,k+1/2}^{n} - (1-\theta)\frac{\Delta t}{\Delta z_{i,k+1/2}^{n}}[q_{i,k+1}^{n} - q_{i,k}^{n}] + \frac{\Delta t}{\Delta z_{i,k+1/2}^{n}}\left[v_{i,k+1}^{\mathrm{v}} \frac{\tilde{w}_{i,k+3/2}^{n+1} - \tilde{w}_{i,k+1/2}^{n+1}}{\Delta z_{j,k+1}^{n}} - v_{i,k}^{\mathrm{v}} \frac{\tilde{w}_{i,k+1/2}^{n+1} - \tilde{w}_{i,k-1/2}^{n+1}}{\Delta z_{i,k}^{n}} \right]$$

$$k = m_i, m_i + 1, \ldots, M_i^n - 1 \tag{21}$$

where $\mathbf{F}$ is an explicit finite difference operator

$$\mathbf{F} w_{i,k+1/2}^{n} = w_{i,k+1/2}^{*} + \Delta t v^{\mathrm{h}} \Delta_h w_{i,k+1/2}^{*} \tag{22}$$

described in Sections 4.3.1 and 4.4. For each cell i, the set of Equations (21) forms a linear, tridiagonal system with unknowns from the same water column $\tilde{w}_{i,k+1/2}^{n+1}$, which is solved by a simple, direct method. All terms in Equations (22) concern one water column i, which confirms that the vertical partition interfaces have an additional advantage that all operations over vertical columns k of cells or sides can be performed locally, without the necessity of any additional communication.

4.3. Iterative or sub-stepping procedures

The iterative parts of the algorithm require a special care in the parallelization procedure. In most cases, exchanging data between partitions inside iterative loops is unavoidable, but the frequency of communication calls can be minimized. In the explicit part of the algorithm, the horizontal diffusion stability requirements force a sub-stepping treatment. In the implicit part of the algorithm, iterative methods appear in the solution of the wave equation for the provisional free surface. A similar situation occurs for the Poisson equation for the dynamic pressure and in the transport treatment.

4.3.1. The explicit treatment the horizontal velocity Laplacian. The horizontal velocity Laplacian $v^{\mathrm{h}} \Delta_h u_{j,k}^{*}$ in (20) and $v^{\mathrm{h}} \Delta_h w_{j,k+1/2}^{*}$ in (22) for the horizontal viscosity terms are discretized with the finite differences involving direct neighbour cells in the horizontal direction. This explicit approach introduces a mild stability constraint on the time step [6]:

$$\Delta t \leqslant \left[2 v^{\mathrm{h}} \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) \right]^{-1} \tag{23}$$

In order not to make it a constraint for the time step of the global scheme, an approach with sub-stepping in time is chosen. It means, for the parallel implementation, that at each sub-step the changing values of horizontal or vertical velocity for halo and interface cell faces must be exchanged in each iteration loop.

DOI: 10.1002/fld

4.3.2. The wave equation. The wave equation method is used to find the provisional position of the free surface and the horizontal velocity. This approach combines Equation (19) and the semi-implicit finite volume discretization of the vertically integrated continuity equation (5):

$$P_i\tilde{\eta}_i^{n+1} = P_i\eta_i^n - \theta\Delta t\sum_{l=1}^{S_i}\left[s_{i,l}\lambda_{j(i,l)}\sum_{k=m}^{M}\Delta z^n_{j(i,l),k}\tilde{u}^{n+1}_{j(i,l)}\right]$$

$$-(1-\theta)\Delta t\sum_{l=1}^{S_i}\left[s_{i,l}\lambda_{j(i,l)}\sum_{k=m}^{M}\Delta z^n_{j(i,l),k}u^n_{j(i,l)}\right] \tag{24}$$

where P_i is the area of ith cell. $s_{i,l}$ is a sign function, equal to $+1$ if the positive value of the velocity on the edge l corresponds to outflow, and equal to -1, if it corresponds to inflow to the ith water column.

Instead of combining (19) and (24) using the discretized forms of (10) and (12) for building a larger equation system (in the order of $N_zN_\mathrm{s}+N_p$), both equations are treated in a decoupled way, in order to find the water level first from a discrete wave equation and the provisional horizontal velocity later. This yields one linear system for N_p unknowns and a set of N_s independent tridiagonal systems of N_z equations. Leaving the derivation details to [11, 24] we quote here the final wave equation for the free surface to be solved in the matrix form:

$$P_i\tilde{\eta}_i^{n+1} - g\theta^2\Delta t^2\sum_{l=1}^{S_i}\frac{s_{i,l}\lambda_{j(i,l)}}{\delta_{j(i,l)}}[(\Delta\mathbf{Z})^\mathrm{T}\mathbf{A}^{-1}\Delta\mathbf{Z}]^n_{j(i,l)}(\tilde{\eta}^{n+1}_{i[j(i,l),2]} - \tilde{\eta}^{n+1}_{i[j(i,l),1]})$$

$$= P_i\tilde{\eta}_i^n - (1-\theta)\Delta t\sum_{l=1}^{S_i}s_{i,l}\lambda_{j(i,l)}[(\Delta\mathbf{Z})^\mathrm{T}\mathbf{U}]^n_{j(i,l)} - \theta\Delta t\sum_{l=1}^{S_i}s_{i,l}\lambda_{j(i,l)}[(\Delta\mathbf{Z})^\mathrm{T}\mathbf{A}^{-1}\mathbf{G}]^n_{j(i,l)} \tag{25}$$

Matrices $\tilde{\mathbf{U}}_j^{n+1}$ and $\Delta\mathbf{Z}_j^n$ contain horizontal velocity components and the vertical cell heights, respectively. $\mathbf{G}_j^n$ contains terms connected to the explicit operator $\mathbf{F}$, free surface and dynamic pressure gradients and shear boundary conditions. $\mathbf{A}_j^n$ contains terms with the vertical diffusion and shear boundary conditions. For parallelization it is enough to state that all the values in these matrices are already given or can be computed locally in partitions.

Equations (25) form a linear, sparse system of equations in the order of number of cells N_p for $\tilde{\eta}_i^{n+1}$, which is strongly diagonally dominant, symmetric and positive definite. It is solved using preconditioned conjugate gradient method in an iterative manner until the residual norm becomes smaller than a user-defined tolerance ε_η. This method requires a few communication calls for polygon-oriented values in the pre-conditioning stage, however only one complete polygon halo swapping of the values for dot products in the internal iterations of the solver, as well as computing global sums for, e.g. residual values.

With the provisional water level $\tilde{\eta}$ computed and communicated between sub-domains, the provisional horizontal velocity $\tilde{u}$ is found from a transformed form of Equation (19):

$$\mathbf{A}_j^n\tilde{\mathbf{U}}_j^{n+1} = \mathbf{G}_j^n - \theta g\frac{\Delta t}{\delta_j}(\tilde{\eta}^{n+1}_{i(j,2)} - \tilde{\eta}^{n+1}_{i(j,1)})\Delta\mathbf{Z}_j^n \tag{26}$$

which forms a set of N_s independent tridiagonal systems of N_z linear equations for the same water column over an edge j. With all matrices computed and updated via communication previously, this part can be done in a separate way in all sub-domains, exchanging the final results with neighbour partitions only once later.

DOI: 10.1002/fld

4.3.3. The discrete Poisson equation. The central part of the second fractional step of UnTRIM algorithm concerning the computational effort is the discrete Poisson equation. It delivers the dynamic (non-hydrostatic) pressure correction $\tilde{q}$, which in combination with the provisional free surface elevation $\tilde{\eta}_i^{n+1}$ yields the global pressure:

$$p_{i,k}^{n+1} = g(\tilde{\eta}_i^{n+1} - z_k) + \tilde{q}_{i,k}^{n+1} \tag{27}$$

Omitting the equation for $k = M$ for simplicity, for the remaining levels k one can express

$$g\theta^2\Delta t^2 \left[\sum_{l=1}^{S_i} s_{i,l}\lambda_{j(i,l)}\Delta z_{j(i,l),k}^n \frac{\tilde{q}_{i[j(i,l),1],k}^{n+1} - \tilde{q}_{i[j(i,l),2],k}^{n+1}}{\delta_{j(i,l)}} + P_i \left(\frac{\tilde{q}_{i,k}^{n+1} - \tilde{q}_{i,k+1}^{n+1}}{\Delta z_{i,k+1/2}^n} - \frac{\tilde{q}_{i,k-1}^{n+1} - \tilde{q}_{i,k}^{n+1}}{\Delta z_{i,k-1/2}^n} \right) \right]$$

$$= g\theta\Delta t P_i(\tilde{w}_{i,k-1/2}^{n+1} - \tilde{w}_{i,k+1/2}^{n+1}) - g\theta\Delta t \sum_{l=1}^{S_i} s_{i,l}\lambda_{j(i,l)}\Delta z_{j(i,l),k}^n \tilde{u}_{j(i,l),k}^{n+1}$$

$$k = m_i, m_i+1, \ldots, M_i - 1 \tag{28}$$

Equations (28) form a system of $N_p N_z$ linear equations solved by preconditioned conjugate gradient solver in an iterative manner. As in the case of the wave equation (25) the iterations of the solvers in sub-domains are strongly coupled. However, the exchange of the values for partial results in each iteration for the 3D halo cells is done only once. Compared with the equation set (25) the number of iterations is usually relatively large, so that the level of coupling can be probably reduced in order to iterate possibly independently in the partitions.

4.3.4. The final stages. In the final projection stage, there are no iterative procedures, but it is described in this place for the sake of completeness. With all provisional values computed and with values exchanged over the overlapping areas, all computations concerning final values can be performed locally. The formulae for the dynamic pressure q and the elevation η as well as the final velocity field (u, w) are

$$q_{i,k}^{n+1} = \tilde{q}_{i,k}^{n+1} - \tilde{q}_{i,M}^{n+1}, \quad k = m_i, m_i+1, \ldots, M_i - 1 \tag{29}$$

$$\eta_i^{n+1} = \tilde{\eta}_i^{n+1} + \frac{\tilde{q}_{i,M}^{n+1}}{g} \tag{30}$$

$$u_{j,k}^{n+1} = \tilde{u}_{j,k}^{n+1} - \theta\frac{\Delta t}{\delta_j}(\tilde{q}_{i(j,2),k}^{n+1} - \tilde{q}_{i(j,1),k}^{n+1}) \tag{31}$$

$$w_{i,k+1/2}^{n+1} = w_{i,k-1/2}^{n+1} - \frac{1}{P_i}\sum_{l=1}^{S_i} s_{i,l}\lambda_{j(i,l)}\Delta z_{j(i,l),k}^n u_{j(i,l),k}^{n+1}, \quad k = m_i, m_i+1, \ldots, M_i - 1 \tag{32}$$

The sum in the continuity equation (32) above is computed for each cell column concerning all surrounding edges, which is done locally in partitions.

Finally, the total depth H at each cell side j is computed from

$$H_j^{n+1} = \max[0, h_j + \eta_{i(j,1)}^{n+1}, h_j + \eta_{i(j,2)}^{n+1}] \tag{33}$$

which also determines if an edge becomes dry or wet. This determination of the cell side status can be done locally in partitions, communicating the pre-condition for the dry/wet state of cells (i.e. values of η and geometry changes due to the free surface movements) before.

DOI: 10.1002/fld

4.3.5. The transport equation. The transport equation (15) is solved using a conservative, semi-implicit finite volume scheme with a min–max property, using flux limiter functions. A time step sub-cycling approach is applied in order to avoid stability restrictions on the time step in the whole model (for details refer to [13, 24]). Consequently, in each sub-cycle a set of N_p linear, tridiagonal systems of equations with N_z unknowns being scalar concentrations defined in the middle of the cells in the same water column is solved applying a direct method. In the parallel implementation, in each sub-cycle the scalar concentrations as well as equation coefficients in the overlapping areas have to be updated between neighbouring partitions, resulting in two halo cell swaps per sub-iteration. Additionally, the time sub-step for sub-iterations must be found as the minimum value in all sub-domains.

4.4. Parallel streamline tracking

Neglecting for the sake of clarity the influence of fluid density ϱ, horizontal viscosity ν^{h} and atmospheric pressure p_{a}, the explicit finite difference operators $\mathbf{F}$ from Equations (20) and (22) can be expressed in a simple form containing only the advection and the Coriolis force terms:

$$\mathbf{F}u^n_{j,k} = \frac{[1-\theta(1-\theta)f^2\Delta t^2]u^*_{j,k} + f\Delta t v^*_{j,k}}{1+\theta^2 f^2 \Delta t^2} \tag{34}$$

$$\mathbf{F}w^n_{i,k+1/2} = w^*_{i,k+1/2} \tag{35}$$

The values of $u^*_{j,k}$ and $w^*_{i,k+1/2}$, respectively, are obtained using the semi-Lagrangian method for the advection ($v^*_{j,k}$ is the local tangential component for Coriolis terms). The values for a head side (j,k) at t^{n+1} are found following the streamline (i.e. characteristic curve, Lagrangian trajectory) backwards in time until the foot position for t^n is found and interpolating the value there from the surrounding prism sides for this previous time step (Figure 3). For convenience, the term *traceback* is introduced to describe this procedure including the data required for the actual streamline backtracking, the interpolation and the finally obtained results.

The semi-Lagrangian advection methods are awkward to treat in the communication pattern of partition neighbourhood relationships described in Section 4.2. The first idea—to extend the overlapping halo cell areas into neighbourhood partitions so far as the expected Courant numbers ($Cr_{j,k} = u^n_{j,k}\Delta t/\delta_j$) dictate—allows performing the streamline backtracking locally in one partition, but imposes some arbitrarily set limitation on the time step (or forces to sub-stepping with smaller time steps) and is inevitably connected with unnecessarily growing amounts of data to be exchanged via communication for larger Courant numbers [3, 4]. Indeed, if the Courant numbers could be kept low enough so that it could be guaranteed that the backtracking does not leave the partitions extended with halo cells at any time, no special treatment of them would have been necessary. However, at this stage another component of the method comes into play, namely the velocity value interpolation at the foot position. Owing to the fact that the interpolation quality determines the level of the numerical diffusivity of the scheme [25], UnTRIM implements the interpolation in polygon patches if the polygon neighbours are available, which allows identical accuracy in the serial and parallel case only for the internal partition cells, and not the external halo cells (Figure 3). In order to avoid differences in accuracies between runs with different numbers of partitions, the halo overlapping would have to be increased even more.

A completely different approach would be to give up the semi-Lagrangian treatment of the advection terms in UnTRIM and apply an alternative advection scheme at the cost of imposing

DOI: 10.1002/fld

some limitation on the time step, but allowing a straightforward parallel implementation. However, the application of modern finite difference Eulerian schemes for unstructured meshes (e.g. [18]) has been excluded so far due to their adverse properties for general-purpose codes; see e.g. [17]. An implementation of an implicit advection scheme for unstructured meshes would be simpler to realize in the case of a decomposed domain, but would probably change the numerical properties of the overall scheme and has not been considered.

Therefore, an algorithm has been developed for the parallel streamline tracking in which the tracebacks leaving partitions they originate from are treated as autonomous, self-contained objects. The data type describing these tracebacks is also defined as a specific MPI data type for efficient message passing using arrays of such objects. If the streamline backtracking leaves the given partition (i.e. crosses the common interface sides), the traceback in question gets an identifier describing its origins—the partition and the head position (i, j, k) there—as well as the position in the neighbour partition it is passing into. The traceback object also contains fields for the interpolated values at the foot of the trajectory. All leaving tracebacks are sent to and implanted in the partitions they are to enter for a further treatment. If a next interface between partitions is met while backtracking in the neighbour partition, the autonomous object is passed to the further neighbours, and so on, until all trajectory feet are found. When it happens, all finalized tracebacks are sorted according to their origin partitions in their identifiers and sent back there using global (*all-to-all*) communication methods. In the final stage, the interpolated values contained in the received lost traceback objects are applied where they belong—at their head positions.

The main advantage of the new algorithm is that it reduces the parallel overhead to the treatment of the 'lost' tracebacks only and is in a more intensive use only when larger flow Courant numbers nearby the partition interfaces occur.

4.5. Boundary conditions

The parallel implementation brings a complication for prescribing boundary conditions: in the process of the domain decomposition, the open boundaries can be divided between partitions. For boundaries with imposed water levels, Equation (14), this is the simple issue of recognizing which boundary belongs to which partition. However, in the case of velocity boundary conditions in the form of imposed fluxes, in order to apply the correct values in each stretch of the divided boundary, an additional communication is required. This is realized for a general case of numerous open boundaries by definition of MPI sub-communicators in processor groups sharing open boundaries with imposed fluxes. This adds only a minimal amount of communication to the scheme.

5. RESULTS

5.1. Verification

Taking results of the serial code as reference, verification of a parallel code implementation is based on presenting evidence that obtained numerical results do not differ between runs with different numbers of partitions and that the code execution speedup grows with the number of processors applied. In the mesh partitioning method, the unavoidable, but usually very small, discrepancies between runs with varying processor numbers are caused by changes in the number and sequence of the floating-point operations due to the different conditioning of the iterative equation solvers

DOI: 10.1002/fld

and changing iteration numbers there. Some differences can also be caused by the variability in reaching thresholds triggering different algorithm parts (e.g. for drying and wetting treatment, min/max principles). Numerous examples from the *UnTRIM validation document* [24] as well as from the BAW consultancy work [2, 26, 27] have been absolved by the parallel implementation. In the following text two representative test cases for the hydrodynamic part of the algorithm (one schematic and one real-life application) are presented and discussed, taking also into account the formulation of recommendations for maintaining the scheme accuracy and performance in typical applications.

The accuracy and parallel performance investigations have been done applying the state-of-the-art shared-memory massive parallel compute servers of the BAW, SGI *Altix* machines with 1600 MHz Intel *Itanium-2* processors with 6 MB secondary cache. The computations were made in a normal, multi-user server operating modus, however, using CPU-sets guaranteeing relatively good reproducibility of the achieved execution times (never more than 5% discrepancies between runs with variable machine load). Therefore, it is assumed that hardware quality cannot have a negative influence on the performance of parallel codes adequate for runs up to 250 processors.

5.2. Accuracy

The verification example concerning the short waves entering a harbour basin demonstrating UnTRIM ability to model wave reflection and diffraction in a complex geometry is a demanding example for testing the parallel implementation due to the fact that all parts of the non-hydrostatic hydrodynamic algorithm must be applied in this case [24, 28].

A harbour basin with a constant depth of $H = 10\,\mathrm{m}$ and the horizontal geometry presented in Figure 4 is meshed applying 13 096 triangular polygons with 19 834 edges. The typical edge length in the mesh interior is in the range $5\,\mathrm{m} < \lambda_j < 5.6\,\mathrm{m}$ and circumcentre distances in $2.8\,\mathrm{m} < \delta_j < 3.75\,\mathrm{m}$. In the vertical direction the mesh consists of 32 layers in the range between -2.0 and 10 m, the upper 12 layers with a thickness of 0.25 m and the remaining of 0.5 m. The resulting 3D mesh consists of 418 208 prisms. At the harbour basin entrance, the free surface is agitated by a sinusoidal wave with a period of 12 s and an amplitude of 1 m. The time step is chosen to be $\Delta t = 0.25\,\mathrm{s}$, and the simulation time is $t = 90\,\mathrm{s}$. The entering waves are diffracted at the left entrance corner of the basin and reflected from the diagonal basin wall to the right and the wave-breaker inside the harbour. By $t = 90\,\mathrm{s}$ the entering, diffracted and reflected waves are superimposed, producing a complicated wave pattern, which is taken for the comparison of the differences between parallel runs with 1 and 4 or 48 processors. Four values for both the solver accuracies, $\varepsilon_\eta = \varepsilon_q = \varepsilon$, namely 10^{-4}, 10^{-6}, 10^{-8} and 10^{-10} are applied.

The resulting 12 comparison cases are presented in Table I. Although for the larger precision ε the number of iterations in the solvers for the water elevation i_η and the dynamic pressure i_q grows, the residual values r_η and r_q computed as a sum from all partitions in the parallel case do not change to a large precision with the varying number of processors n, resulting in the same number of iterations. There is a tendency that the maximum differences in the computed values for elevation η_{diff}, normal velocity u_{diff} and dynamic pressure q_{diff} grow slightly with the number of processors n, but remaining at a very similar level independently from the precision of the solvers. This is a clear indication that these differences are proportional to the length of interfaces between mesh partitions, i.e. to the amount of the communication. However, it must be noted that the level of these discrepancies is thoroughly negligible for practical purposes.

DOI: 10.1002/fld

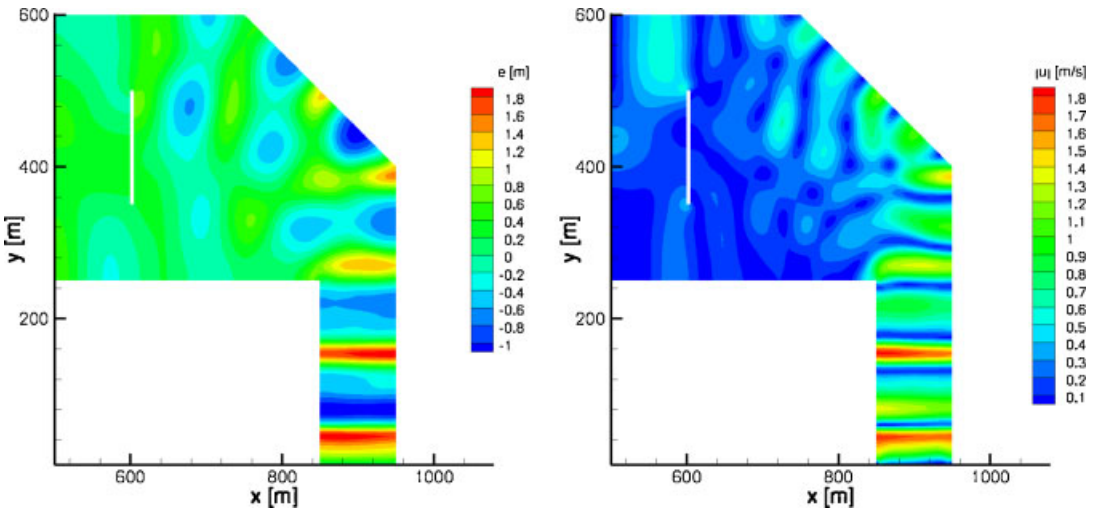

Figure 4. The test case *short waves in a harbour basin*. The wave pattern in the harbour basin of a constant depth after $t=90$ s of simulation time, shown in terms of water levels e and the absolute value of the vertically averaged velocity $|u|$.

Table I. Accuracy of the model *short waves in a harbour basin*.

n	ε	i_η	r_η	i_q	r_q	η_{diff}	u_{diff}	q_{diff}
1	10^{-4}	5	0.57847440E−05	24	0.95682396E−04			
4	10^{-4}	5	0.57847440E−05	24	0.95682396E−04	1.065E−14	1.261E−13	9.592E−14
48	10^{-4}	5	0.57847440E−05	24	0.95682396E−04	1.221E−14	8.970E−14	9.814E−14
1	10^{-6}	6	0.19502884E−06	33	0.58604017E−06			
4	10^{-6}	6	0.19502884E−06	33	0.58604017E−06	7.993E−15	1.171E−13	8.704E−14
48	10^{-6}	6	0.19502884E−06	33	0.58604017E−06	8.215E−15	1.327E−13	8.082E−14
1	10^{-8}	7	0.71237680E−08	40	0.79228882E−08			
4	10^{-8}	7	0.71237680E−08	40	0.79228882E−08	1.265E−14	1.221E−13	1.287E−13
48	10^{-8}	7	0.71237680E−08	40	0.79228882E−08	1.421E−14	1.526E−13	1.261E−13
1	10^{-10}	9	0.69746699E−11	47	0.90761763E−10			
4	10^{-10}	9	0.69746699E−11	47	0.90761763E−10	2.264E−14	1.876E−13	1.616E−13
48	10^{-10}	9	0.69746699E−11	47	0.90761763E−10	1.376E−14	2.075E−13	1.088E−13

5.3. *Performance*

If t_n is the computation time applying n processors, the parameters describing the parallel performance gain compared with a run with m processors ($m \leqslant n$, usually $m=1$), such as the cost c, speedup s_m, efficiency e_m are defined here as follows:

$$c(n)=nt_n, \quad s_m(n)=t_m/t_n, \quad e_m(n)=s_m(n)/n \tag{36}$$

As a representatively large model for a convincing performance test, a model of a 14 km-long lowland stretch of the main channel of the River Elbe by Coswig [29] resolved with a high-quality,

DOI: 10.1002/fld

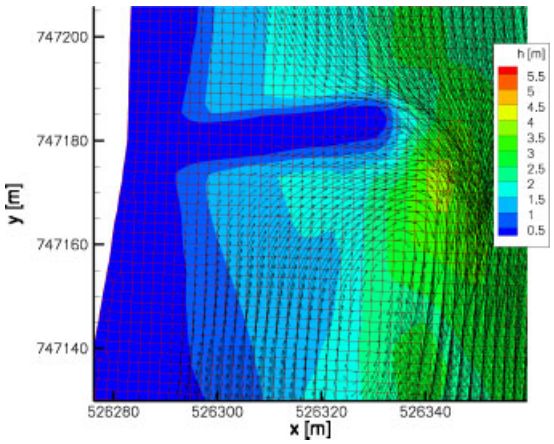

Figure 5. A fragment of the high-resolution mesh of the model *The River Elbe by Coswig*.

fine, almost regular mesh of quadrilateral polygons with $\delta_j \approx 2\,\mathrm{m}$ and in the 3D case 22 horizontal levels with $\Delta z_k = 0.5\,\mathrm{m}$ is presented. For this resolution, the base mesh has 738 485 polygons and the 3D mesh 8 006 838 cells (Figure 5).

The model is initialized with imposed water levels known from the measurements all over the area and steered with the constant flux of $360\,\mathrm{m}^3/\mathrm{s}$ at the inflow and a constant water level of 58.42 m at the outflow. For these conditions, typical for a mean flow in this river stretch, the resulting shoreline must be reproduced from a pattern of dry and wet polygons on the narrow foreland, which are set anew in each time step, taking into account a threshold of $\Delta z_{\min} = 5\,\mathrm{mm}$ representing a minimum water depth below which a polygon or edge is treated as dry. In the runs with various number of partitions, the percentage of wet polygons may vary around the mean value of 77% at the end of the computation, with the discrepancies growing with the number of processors—e.g. for 192 processors this percentage varies already between ca. 30 and 100%. UnTRIM is applied in 2D and 3D, non-hydrostatic mode. In all cases, 300 time steps of $\Delta t = 2.0\,\mathrm{s}$ are computed starting from the initial conditions described above. The solver accuracy for the water level is set to $\varepsilon_\eta = 10^{-6}$ and for the dynamic pressure to $\varepsilon_q = 10^{-9}$.

The performance results for the 2D case are presented in Table II for the processor range 1–192 and for the 3D case in Table III for the range 8–192. The 2D results show a super-linear speedup relative to one processor s_1, revealing suboptimal serial runs by larger meshes. Similar effects can be observed in the speedup data relative to 8 processors s_8 for the computationally much more intensive 3D case. In order to interpret the data, the speedup is scaled to the number of processors by which the parallel efficiency e_1 or e_8 of the code reaches a maximum, i.e. for $m = 96$ processors. The relative efficiency e_{96} diminishes below 96 processors, which can be explained by cache misses occurring through the indirect addressing in larger unstructured meshes. The relatively smaller efficiency losses above 128 processors arise from the growing relative amount of the halo cells causing an increase in the communication, as well as from growing discrepancies between wetting percentage of the partitions. However, as tests with other models confirm, the latter factor is important only when these discrepancies grow much more drastically causing larger imbalances in the computational load, which can especially be noted when completely dry and completely wet partitions occur.

DOI: 10.1002/fld

Table II. Performance of the 2D model *The Elbe River by Coswig*.

n	t_n (s)	c (s)	s_1 (n)	e_1 (n)	s_{96} (n)	e_{96} (n)	Halo (%)	Wet (%)
1	12043.3	12043.3	1.000	1.000	0.745	0.745	0.00	77.77
2	5654.8	11309.6	2.130	1.065	1.588	0.794	0.03	77.12 – 78.43
4	2668.5	10673.8	4.513	1.128	3.364	0.841	0.09	74.75 – 79.48
8	1309.9	10479.8	9.194	1.149	6.853	0.857	0.20	73.94 – 82.67
16	616.9	9869.6	19.524	1.220	14.554	0.910	0.43	69.59 – 83.99
32	289.7	9270.7	41.570	1.299	30.987	0.968	0.95	66.10 – 88.43
48	194.3	9326.0	61.986	1.291	46.205	0.963	1.44	65.68 – 88.37
64	142.6	9128.8	84.433	1.319	62.938	0.983	1.90	56.06 – 89.76
96	93.5	8977.3	128.786	1.342	96.000	1.000	2.87	42.05 – 90.75
128	70.9	9070.0	169.960	1.328	126.692	0.990	3.57	49.12 – 92.31
160	57.9	9274.2	207.774	1.299	154.879	0.968	3.99	43.66 – 92.10
192	53.1	10186.9	226.990	1.182	169.203	0.881	4.46	29.60 – 100.0

Table III. Performance of the 3D model *The Elbe River by Coswig*.

n	t_n (s)	c (s)	s_8 (n)	e_8 (n)	s_{96} (n)	e_{96} (n)
8	15718.4	125747.2	8.000	1.000	6.428	0.803
16	7752.0	124031.6	16.221	1.014	13.033	0.815
32	3654.9	116956.4	34.405	1.075	27.643	0.864
48	2436.8	116964.9	51.604	1.075	41.461	0.864
64	1709.3	109395.2	73.566	1.149	59.106	0.924
96	1052.4	101030.4	119.486	1.244	96.000	1.000
128	821.0	105090.1	153.160	1.197	123.055	0.961
160	665.1	106412.1	189.072	1.182	151.908	0.949
192	587.4	112778.3	214.079	1.115	172.000	0.896

In order to investigate the influence of the new parallel streamline algorithm (Section 4.4) on the overall speedup of the code, runs with two different time steps are compared. For the mean flow in the middle of the river channel with a representative velocity of approx. 1 m/s, by $\Delta t = 0.5$ s the Courant numbers are small enough so that only the tracebacks stemming directly from the interface between partitions are treated. By $\Delta t = 2.0$ s one has already additional hundreds (2D) or thousands (3D) of tracebacks leaving the partition interiors to be dealt with by the new parallel streamline tracking algorithm. In Table IV 2D and 3D, non-hydrostatic computations are contrasted additionally with run results when the advection step is switched off (physically incorrect, but numerically possible). The data reveal that in the given error range the speedup is only slightly dependent on the fact, if the parallel advection algorithm is applied in its full extent or not (columns 2s,a, 0.5s,a) or the advection is not taken into account at all (columns 0.5s,na and 2s,na).

For studying the influence of the growing amount of halo the swapping on the speedup in the available range of processors, the verification case *short waves in a harbour basin* (Section 5.2) is adequate, because the mesh is relatively smaller and no wetting and drying of mesh polygons occur. The results for (unphysical) two-dimensional 2D and three-dimensional hydrostatic 3Dhy, as well as (physical) non-hydrostatic case 3Dnh, which in this context differ strongly in the computational cost, are presented in Table V. In all cases $\varepsilon_\eta = \varepsilon_q = 10^{-10}$.

DOI: 10.1002/fld

Table IV. The model *The Elbe River by Coswig*, comparison between the achieved speedup (scaled for 96 processors) according to the effort required for the parallel advection scheme.

	2D				3D			
n	2s,a	0.5s,a	2s,na	0.5s,na	2s,a	0.5s,a	2s,na	0.5s,na
8	6.85	6.84	6.43	6.47	6.43	6.71	6.11	6.51
16	14.55	14.52	13.89	13.88	13.03	13.65	12.45	13.18
32	30.99	30.23	29.72	29.20	27.64	28.77	26.46	28.18
48	46.21	45.13	44.76	44.21	41.46	42.53	39.35	41.72
64	62.94	62.06	61.59	60.95	59.11	59.59	57.82	58.87
96	96.00	96.00	96.00	96.00	96.00	96.00	96.00	96.00
128	126.69	128.40	128.27	129.81	123.06	122.78	124.25	123.93
160	154.88	159.28	159.94	162.46	151.91	152.58	155.16	154.28
192	169.20	175.80	175.42	178.95	172.00	172.64	178.60	175.01

Table V. Performance of the model *short waves in a harbour basin*.

	t_n (s)			$s_1(n)$			
n	2D	3Dhy	3Dnh	2D	3Dhy	3Dnh	Halo (%)
1	54.5	985.6	2651.1	1.0	1.0	1.0	0.0
2	27.3	507.3	1280.7	2.0	1.9	2.1	1.06
4	14.1	257.2	563.4	3.9	3.8	4.7	2.63
8	7.7	133.4	272.1	7.1	7.4	9.7	4.27
12	5.6	90.8	186.2	9.8	10.9	14.2	5.90
16	4.7	71.8	146.2	11.6	13.7	18.1	7.62
24	3.8	49.8	103.0	14.5	19.8	25.7	9.99
32	3.5	39.7	83.2	15.8	24.8	31.8	12.02
48	3.1	28.5	59.9	17.8	34.6	44.3	15.02
64	3.6	24.2	53.9	15.1	40.8	49.2	17.80
96	4.4	19.1	42.8	12.5	51.6	62.0	22.39

It can be roughly estimated that for the 2D case a breakup of the speedup with the increasing number of processors appears above ca. 5% of halo cells in partitions, in 3D hydrostatic case above ca. 10% and in the most computationally intensive non-hydrostatic case above ca. 15%. These values can also be confirmed with other test cases.

As a conclusion, it can be stated that for the parallel implementation of UnTRIM the recommendations for users concerning the accuracy and performance limits are typical for the domain decomposition method: when the computational load can be kept balanced between partitions, a larger percentage of halo cells should be avoided.

6. SUMMARY AND OUTLOOK

6.1. Summary of the parallel algorithm

The parallel implementation of the UnTRIM algorithm follows the mesh partitioning method for the finite volume and finite difference parts of the scheme. The partitioning of the horizontal base mesh

DOI: 10.1002/fld

for well-balanced partitions in terms of number of polygons and shortest possible interface lengths is based on the *Metis* and *ParMetis* libraries [23]. The exchange of data concerns overlapping halo areas of the neighbouring mesh partitions and can be reduced only to the objects concerning the halo cells directly adjacent to the vertical prism sides along the vertical interfaces. The initial stage of the computation deals with preparing partitioned mesh and data structures optimized for the halo-swapping efficiency with 2D and 3D objects treated separately and sorted into possibly contiguous data fields. This is especially important for all strongly coupled iterative parts of the algorithm, where communication is necessary in each iteration.

This approach leads to a limited set of *point-to-point communications* between directly neighbouring mesh partitions. The *parallel overhead*—i.e. costs of the communication between processors and the effort required to prepare data for sending and applying the received values—is diminished during the actual run by limiting the amount of exchanged data only to the required 2D and 3D halo objects (cells, sides, edges, polygons) and pre-sorting the required fields for efficiency. The *global communication* is necessary in the case of the new object-oriented parallel algorithm for the streamline tracking in the explicit advection treatment with the semi-Lagrangian method. The parallel overhead is limited only to the treatment of these tracebacks that actually leave the partitions. The *sub-communication* in processor groups is sporadically necessary for treatment of boundaries with prescribed fluxes, with a minimal amount of data to be exchanged.

In order to maintain the good parallel performance of the code, the recommendations typical for domain decomposition methods apply, with the percentage of the halo cells and the computational load balancing between partitions as the determining and limiting factors for the speedup. However, if a few simple recommendations are followed, even in the computationally most intensive cases the reached scalability of the whole algorithm is appropriate for dealing with larger, highly resolved models aimed for real-world hydro-engineering studies.

6.2. *Outlook*

Further improvements in the overall performance and accuracy are expected by advances in the UnTRIM algorithm and its software implementation. The super-linear speedup reveals that there is a possibility of performance increase for the lower number of partitions by modifying UnTRIM software implementation in order to use the processors with a larger secondary cache more efficiently. Advances are expected also by balancing the computational load using a hybrid parallel execution mode (MPI and OpenMP) and by further developments in the practical applicability of partitioning methods taking criterions concerning, e.g. expected water levels during the simulations. Additionally, maintaining accuracy and reproducibility of the results between runs with different numbers of processors is to be investigated with regard to local and global wet/dry thresholds and min/max principles in these parts of the serial algorithm which are being modified presently.

7. CONCLUSIONS

A message-passing parallel version of UnTRIM based on the mesh partitioning method has been developed without compromising the properties of the serial code. Scalability appropriate for dealing with larger models is reached due to the communication methods appropriately designed for the significant parts of the algorithm and minimizing the amount of data exchanged between the processors. The new autonomous parallelized streamline backtracking scheme does not influence

DOI: 10.1002/fld

the overall scalability of the scheme and is free of any limitations due to the parallel implementation. The computations with standard verification test cases and larger real-world models confirm that the presented implementation allows an efficient and accurate application of this code on state-of-the-art high-performance computers for computationally intensive hydraulic engineering applications.

ACKNOWLEDGEMENTS

The author thanks numerous colleagues at BAW who helped directly or indirectly by this study, whose name list is unfortunately too long to present here. An exception is being made for Regina Patzwahl and Günther Lang for making their test cases available for this publication, Carsten Thorenz for his numerical support and Thomas Lege for making it possible to finish this work. Last but not least, the continuous support from Vincenzo Casulli and the interest of international participants of UnTRIM workshops are thankfully acknowledged.

REFERENCES

1. Croce R, Griebel M, Schweitzer M. A parallel level-set approach for two-phase flow problems with surface tension in three space dimensions. Preprint 157, Sonderforschungsbereich 611, Universität Bonn, 2004.
2. Strybny J, Thorenz C, Croce R, Engel M. A parallel 3D free surface Navier–Stokes solver for high performance computing at the German waterways administration. *Proceedings of the Seventh International Conference on Hydroscience and Engineering* (*ICHE 2006*), Philadelphia, PA, U.S.A., 2006. Available from: http://idea.library.drexel.edu/handle/1860/732.
3. Hinkelmann R. Parallelisierung eines Lagrange-Euler-Verfahrens für Strömungs-und Stofftransportprozesse in Oberflächegewässern. *Ph.D. Thesis*, Universität Hannover, Institut für Strömungsmechanik und ERiB, Bericht Nr. 51/1997, 1997.
4. Kräutle S. A Navier–Stokes solver based on CGBI and the method of characteristics. *Ph.D. Thesis*, Naturwissenschaftliche Fakultäten, Friedrich-Alexander-Universität Erlangen-Nürnberg, 2001.
5. Casulli V. Semi-implicit finite difference methods for the two-dimensional shallow water equations. *Journal of Computational Physics* 1990; **86**:56–74.
6. Casulli V, Cheng R. Semi-implicit finite difference methods for three-dimensional shallow water flow. *International Journal for Numerical Methods in Fluids* 1992; **15**:629–648.
7. Casulli V, Cattani E. Stability, accuracy and efficiency of a semi-implicit method for three-dimensional shallow water flow. *Computers and Mathematics with Applications* 1994; **27**(4):99–112.
8. Cheng R, Casulli V, Gartner J. Tidal, Residual, Intertidal Mudflat (TRIM) model and its applications to San Francisco Bay, California. *Estuarine*, *Coastal and Shelf Science* 1993; **36**:235–280.
9. Casulli V. A semi-implicit finite difference method for non-hydrostatic, free surface flows. *International Journal for Numerical Methods in Fluids* 1999; **30**:425–440.
10. Casulli V, Walters R. An unstructured grid, three dimensional model based on the shallow water equations. *International Journal for Numerical Methods in Fluids* 2000; **32**:331–348.
11. Casulli V, Zanolli P. Semi-implicit numerical modelling of non-hydrostatic free-surface flows for environmental problems. *Mathematical and Computer Modelling* 2002; **36**:1131–1149.
12. Cheng R, Casulli V. Evaluation of the UnTRIM model for 3-D tidal circulation. *Proceedings of the 7th International Conference on Estuarine and Coastal Modeling*, St. Petersburg, FL, U.S.A., 2001; 628–642.
13. Casulli V, Zanolli P. High resolution methods for multidimensional advection–diffusion problems in free-surface hydrodynamics. *Ocean Modelling* 2004; **10**:137–151.
14. Ham D. On techniques for modelling coastal and ocean flow with unstructured meshes. *Ph.D. Thesis*, Faculty of Civil Engineering and Geosciences, Delft University of Technology, Delft, The Netherlands, 2006.
15. Ham D, Kramer S, Stelling G, Pietrzak J. The symmetry and stability of unstructured mesh C-grid shallow water models under the influence of Coriolis. *Ocean Modelling* 2007; **16**:47–60.
16. Lippert C, Sellerhoff F. Efficient generation of orthogonal unstructured grids. *Proceedings of the Seventh International Conference on Hydroscience and Engineering* (*ICHE 2006*), Philadelphia, PA, U.S.A., 2006. Available from: http://idea.library.drexel.edu/handle/1860/732.

DOI: 10.1002/fld

17. Fringer O, Gerritsen M, Street R. An unstructured-grid, finite-volume, nonhydrostatic, parallel coastal simulator. *Ocean Modelling* 2006; **14**:139–173.
18. Perot B. Conservation properties of unstructured staggered mesh schemes. *Journal of Computational Physics* 2000; **159**:58–89.
19. Labeur R, Pietrzak J. A fully three dimensional unstructured grid non-hydrostatic finite element coastal model. *Ocean Modelling* 2005; **10**:17–28.
20. Zhang Y, Baptista A, Myers E. A cross-scale model for 3D baroclinic circulation in estuary-plume-shelf systems: I. Formulation and skill assessment. *Continental Shelf Research* 2004; **24**:2187–2214.
21. OpenMP Architecture Review Board. *OpenMP Fortran Application Program Interface*, Version 1.1. November 1999. Available from: http://www.openmp.org/.
22. Message Passing Interface Forum. *MPI*: *A Message Passing Interface Standard*, *Version 1.1*. University of Tennessee, Knoxville, 12 June 1995. Available from: http://www-unix.mcs.anl.gov/mpi/.
23. Karypis G, Kumar V. Multilevel k-way partitioning scheme for irregular graphs. *Journal of Parallel and Distributed Computing* 1998; **48**(1):96–129. http://www.cs.umn.edu/~metis.
24. Casulli V, Lang G. Mathematical model UnTRIM, validation document, Version 1.0. *Technical Report*, Bundesanstalt für Wasserbau, Hamburg, June 2004. Available from: http://www.baw.de/vip/abteilungen/wbk/Methoden/hnm/untrim/PDF/vd-untrim-2004.pdf.
25. Malcherek A. *Numerische Methoden der Strömungsmechanik*. Lecture Script, Bundesanstalt für Wasserbau. Außenstelle Küste, Hamburg, 1999. Available from: http://www.baw.de/vip/abteilungen/wbk/Methoden/hnm/nummeth/numerik.pdf.
26. Lege T, Stamm J, Abel D. 2D-Finite-element-modelling of the lower Rhine River. *QNET-CFD-Newsletter* 2003; **2**(3).
27. Lege T, Alexy M, Kellermann J. Three-dimensional flow fields as a prerequisite for sediment transport modelling in lowland rivers. *Proceedings of the IAHR Fifth International Symposium on Environmental Hydraulics* (*ISEH V*), Phoenix, AZ, U.S.A., 2007.
28. Jankowski JA. A non-hydrostatic model for free surface flows. *Ph.D. Thesis*, Universität Hannover, Institut für Strömungsmechanik und ERiB, Bericht Nr. 56/1999, 1999.
29. Patzwahl R, Jankowski J, Lege T. Very high resolution numerical modelling for inland waterway design. Submitted to *International Conference on Fluvial Hydraulics* (*River Flow 2008*), Izmir, Turkey, 2008. Available from: http://riverflow2008.org.

DOI: 10.1002/fld