Further developments of UnTRIM: parallel implementation and its verification

Jacek A. Jankowski[*]

Abstract

This contribution deals with further developments of *UnTRIM*, an unstructured-grid, three-dimensional, semi-implicit finite difference – finite volume model for the shallow water equations [5]. Attractive numerical properties of the method like its robustness due to the unlimited stability and successes in practical applications spawned efforts aimed at making the available code fit for the high performance computing in order to address larger, complex problems in hydraulic engineering.

The paper concentrates on the parallel implementation of the program, based on the domain decomposition method and message passing, which has been achieved without negatively affecting any of the properties of the serial code. A special attention is paid to a new, autonomous parallel streamline tracking algorithm, which allows using semi-Lagrangian methods in decomposed meshes without compromising the scalability of the code.

The new developments have been carefully verified not only with the numerous 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, where the hydrodynamics can be coupled with wave, transport and morphodynamical models.

The presented achievements pair robust and efficient numerical methods with the state-of-the art high performance computing know-how in order to deliver a solid base for computationally intensive hydraulic engineering applications.

1 Introduction

1.1 The numerical scheme

The hydraulic engineer applying numerical methods for solving practical environmental problems is nowadays confronted with a multitude of various readily available models which outbid each other with numerous features and applicability claims. In this situation experienced practitioners almost intuitively prefer simpler, but general purpose codes which not only cover the aimed application domain, but also clearly guarantee efficiency, robustness and accuracy of the numerical scheme.

Since ca. 1990 the *TRIM* family of finite difference models for structured meshes has been systematically developed by Casulli and his co-workers, with a special attention not only to the proven numerical stability, accuracy and efficiency of the scheme, but also to its practical applicability [3, 2]. All these features are inherited by *UnTRIM*, the extension of the final *TRIM* algorithm [1] for unstructured meshes in order to deal with complex boundaries more flexibly and allow for local mesh refinements [5, 6].

UnTRIM is a practical scheme for solving the three dimensional shallow water equations with a semi-implicit, fractional step time integration, a finite difference/volume spatial discretisation and a semi-Lagrangian treatment of advection using an unstructured, orthogonal grid. The aimed application domain are three-dimensional, non-hydrostatic environmental free surface flows including species transport. The orthogonal mesh consists of horizontal layers of prismatic cells with horizontal triangular or qudrilateral bases. In the particular case, when only one layer is defined, the algorithm is consistent with the two dimensional, vertically integrated shallow water equations. The drying and flooding of computational cells is included in a natural way.

The governing momentum equations together with the equation for the free surface and the incompressibility condition are treated by a relatively simple and efficient semi-implicit, fractional step 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 imply a mild stability criterion; for modelling of internal waves their speed is also a limiting factor for the time step. For barotropic flows, when the horizontal viscosity is neglected, the scheme is unconditionally stable.

[*] Department of Hydraulic Engineering in Inland Areas, Federal Waterways Engineering and Research Institute (BAW), Kussmaulstr. 17, 76187 Karlsruhe, Germany. E-mail:jacek.jankowski@baw.de

For regular meshes the discretisation error is second-order in space, as well as in the time when the semi-implicit scheme is equally balanced between time levels. For irregular meshes or fully implicit computations, the discretisation error grows, but can be diminished when the polygon dimensions vary gradually in the domain.

The choice of the orthogonal, staggered mesh is beneficial in respect to the overall stability of the method compared e.g. to the centroid-based schemes [8], which justifies the larger mesh generating effort for the strictly orthogonal meshes [12]. Additionaly, the applied semi-Lagrangian momentum advection scheme has clear advantages for general purposes compared to Eulerian methods affected by time step and spatial discretisation limitations, especially when wetting and drying occurs [7, 15].

The *UnTRIM* software structure with an efficient numerical kernel accessible via a clear 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 morphodynamical models coupling.

It is noteworthy that the attractive properties of Casulli formulation have motivated or influenced numerous very similar code efforts, whereby *Elcirc* [17], *Suntans* [7], *Delfin* [9] or *Finel* [11] can be named as examples without a claim for completeness.

1.2 Message-passing parallelism

In order to profit form the enormous growth of available computational resources of modern parallel computers an additional, specific effort is required from the code developers. Unfortunately, only in exceptional cases codes are designed as parallel programs from a scratch (e.g. NaSt3D [16]), normally a serial code must be adapted for parallel execution, which may result in some cases in non-trivial changes in the numerical formulation.

A parallel version of *UnTRIM* with the shared memory parallelization paradigm using OpenMP standards [14] exists since 2002. The achieved parallel speed-up up to 10 and overall performance of the code in terms of load balancing is excellent as for OpenMP, but required that time high-end machines allowing a larger number of processors to adress a common shared memory. For this parallelisation method the speed-up factor is limited by this part of the code, which has to be executed serially (Amdahl's law). Although the appearance of affordable multicore processors makes the further OpenMP version maintenance still attractive, larger speed-up factors can be efficiently achieved using domain decomposition methods and message passing parallelization standards, like MPI [13]. In the message-passing parallelism with domain decomposition, a separate copy of the program is executed almost independently on each processor of the parallel machine with data concerning a given part of the global meshed domain, exchanging information between subdomains (partitions) when necessary. The speed-up factor depends strictly on the communication efficiency, which is the leading thought of the work presented here.

2 UnTRIM parallelization

2.1 Governing equations

The lack of space forces to assumption that the reader is familiar with the governing three-dimensional shallow water equations with their boundary conditions typical for environmental flows and concentrate exclusively on the exemplary features of the mesh and the numerical formulation relevant for the message-passing parallelization. For the complete reference of the method, see [5, 6] or the *UnTRIM validation document* [4].

2.2 The mesh

The horizontal (i.e. covering the (x,y)-domain of computation) unstructured orthogonal grid has the important property that the segments δ_j $(j = 1,2,...,N_s)$ joining the circumcenters of the mesh polygons P_i $(i = 1,2,...,N_p)$ have an intersection with their common sides λ_j and are orthogonal to each other. The connectivity tables for the sides and N_p polygons having S_i sides allow addressing the objects of the mesh, so that the sides of the polygon can be identified with an index $1 \leq j(i,l) \leq N_s$, $l = 1,2,...,S_i$, and the both polygons sharing the j-side can be identified with $1 \leq i(j,m) \leq N_p$, where $m = 1,2$. In the vertical direction the mesh consists of horizontal levels k, so that the distances between the mesh surfaces identified by $z_{k+1/2}$ are given by $\Delta z_k = z_{k+1/2} - z_{k-1/2}$ $(k = 1,2,...,N_z)$. In this way the three-dimensional mesh consists of horizontal layers of prism with the height of Δz_k with horizontal faces being the orthogonal grid polygons P_i. Presently, only qudrangles and triangles are implemented for the horizontal base mesh.

The discrete variables of the scheme are defined at staggered positions. The bathymetry h_j is defined constant over a polygon side j. The elevation of the free surface η_i^n for the n-th time step is spatially situated at the circumcentre of the polygon P_i and assumed constant there. The velocity $u_{j,k}^n$, treated in the scheme as the normal component to each prism face, is defined on the intersections of the prism centers and their common faces and assumed constant over them. Finally, other variables, like hydrodynamic pressure $q^n_{i,k}$ and m species concentration $q^n_{i,k,m}$ are defined in the centers of the prisms.

The description above suggests that in order to apply the domain decomposition method efficiently, mesh partitioning into horizontally balanced 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 so-called halo (ghost) cells along the vertical interfaces between the partitions.

2.3 Halo swapping

The last statement can be confirmed by the numerical implementation analyse. An exemplary insight can be given studying the first part of the algorithm for the horizontal velocity component only. In this fractional and semi-implicit 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 ν^v from the horizontal momentum equation:

$$\begin{aligned}\tilde{u}^{n+1}_{j,k} &= \mathbf{F}u^n_{j,k} - (1-\theta)\frac{\Delta t}{\delta_j}\left[g(\eta^n_{i(j,2)} - \eta^n_{i(j,1)}) + q^n_{i(j,2),k} - q^n_{i(j,1),k}\right] \\ &- \theta\frac{\Delta t}{\delta_j}(\tilde{\eta}^{n+1}_{i(j,2)} - \tilde{\eta}^{n+1}_{i(j,1)}) + \frac{\Delta t}{\Delta z^n_{j,k}}\left[\nu^v_{j,k+\frac{1}{2}}\frac{\tilde{u}^{n+1}_{j,k+1} - \tilde{u}^{n+1}_{j,k}}{\Delta z^n_{j,k+\frac{1}{2}}} - \nu^v_{j,k-\frac{1}{2}}\frac{\tilde{u}^{n+1}_{j,k} - \tilde{u}^{n+1}_{j,k-1}}{\Delta z^n_{j,k-\frac{1}{2}}}\right], \\ & k = m^n_j, m^n_{j+1}, \ldots, M^n_j\end{aligned} \tag{1}$$

where $\mathbf{F}$ is an explicit finite difference operator described in the section 2.4. Note that in vertical, m^n_j to M^n_j describe the variable range of the k indices for the vertical finite difference stecils, $1 \le m^n_j \le M^n_j \le N_z$, which may change due to the bottom and the free surface dynamics. In horizontal, the finite difference stencils in (1) based on the circumcentre distances δ_j make intensive use of the connectivity tables (like $i(j,m)$). It means that when all values for the time step n are updated via communication, all terms concerning central prism points i on the both sides of a side j at a level k can be computed locally in a subdomain extended with halo (ghost) cells provided for each side on all interfaces. This confirms that it is appropriate to partition the horizontal base mesh so that the interfaces are vertical. Then all computations over vertical columns of cells or edges can be then performed locally as well.

It is clear, that the overlapping between the horizontal partitions can be limited to the (halo) cells directly adjacent to the prism sides. The external halo cell objects (polygons, sides, edges, cells) are placed at the end of the appropriate object lists and included in the connectivity tables for subdomains. In this way the computational loops in a given subdomain do not reach the ghost cells indices range, but the ghost cell values are addressed from these loops. Their actual values are computed in the neighbouring subdomains (where they are simply internal cells) and delivered via message passing communication to the neighbour(s) when necessary, and vice versa.

Therefore, the main part of the message-passing parallel implementation is based on the domain decomposition method with overlapping mesh subdomains, which is appropriate for the applied FD/FV scheme. This leads to a set of point-to-point communications between neighbouring mesh partitions. The parallel overhead – costs of the communication between processors and the effort required to prepare data for sending and applying the received values – is diminished by minimalising the overlapping of the meshes and the amount of exchanged data (especially in the iterative parts of the algorithm not discussed here). The efficient partitioning of the horizontal mesh is based on the *Metis* and *ParMetis* libraries [10]. An additional effort is concerned with preparing partitioned mesh and data structures optimised for the halo-swapping efficiency.

2.4 Parallel streamline tracking

Assuming for simplicity a constant fluid density and atmospheric pressure, the explicit finite difference operator $\mathbf{F}$ from (1) concerns the advection, Coriolis (parameter f) and the horizontal diffusion (coefficient ν^h) terms can be written as:

$$\mathbf{F}u^n_{j,k} = u^*_{j,k} + \Delta t f v^*_{j,k} + \Delta t \nu^h \Delta_h u^*_{j,k} \tag{2}$$

While the horizontal Laplacian Δ_h is discretised with the finite differences, the value of $u^*_{j,k}$ ($v^*_{j,k}$ is the local tangential component) is obtained using the semi-Lagrangian method for the advection (i.e. the method of characteristics). The values for a head side (j,k) at t^{n+1} are obtained following the streamline (characteristic curve, Lagrangian trajectory) backward in time until the foot position for t^n is found and interpolating the value there from the surrounding prism sides for this time step.

The semi-Lagrangian advection methods are awkward to treat in the comunication pattern of partition neigbourhood relationships described in the section 2.3. In order to perform the streamline backtracking just like in the serial case in one partition only, the overlapping areas should extent into neighbourhood partitions so far as the expected Courant numbers dictate. This is adverse to the communication efficiency, and hard to deal with for large Courant numbers. Therefore, an algorithm has been developed for the parallel streamline tracking, free of any implementation limitations, in which the tracebacks leaving given partitions are treated as autonomous objects.

[...describe the new algorithm here...]

3 Results

3.1 Verification

The verification of a parallel implementation is straighforward – it based on direct comparisons of results obtained with different number of applied processors compared to each other and especially to the serial execution with the original code. Numerous examples from the *UnTRIM validation document* [4] and additional ones being presently documented were applied.

[...describe the differences here...]

3.2 Speedup

The speedup investigations have been performed using the state-of-the art parallel compute servers of the BAW, shared-memory SGI *Altix* machines with 1600 Mhz Intel *Itanium-2* procesors with 6 MB secondary cache. The computations were made in a normal, multi-user server operating modus, however in a queue allowing application of 1-128 processors using CPU-sets guaranteeing relatively good reproductability of the achieved execution times.

Table 1: Speedup data for the model *The Elbe River by Coswig* in the 2D modus

np	exhalo%	time[s]	effort[s]	spup	effic	rspup	reffic
1	0.00	11952.4	11952.4	1.000	1.000	0.763	0.763
2	0.03	5721.1	11442.2	2.089	1.045	1.594	0.797
4	0.12	2721.5	10886.0	4.392	1.098	3.351	0.838
8	0.12	1351.8	10814.4	8.842	1.105	6.747	0.843
16	0.50	641.9	10270.4	18.620	1.164	14.208	0.888
32	0.89	303.3	9705.6	39.408	1.231	30.069	0.940
48	0.95	198.8	9542.4	60.123	1.253	45.875	0.956
64	2.03	145.9	9337.6	81.922	1.280	62.509	0.977
96	2.99	95.0	9120.0	125.815	1.311	96.000	1.000
128	3.42	73.0	9344.0	163.732	1.279	124.932	0.976

[...describe the speedup here...]

4 Conclusions

A message-passing parallel version of *UnTRIM* based on the domain decomposition method was developed without compromising the properties of the serial code. A good scalability 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 parallelised streamline tracking scheme does not influence the scalability. In order to maintain the good parallel performance of the code the recommendations typical for domain decomposition methods apply – in addition to appropriate partitioning of the mesh the percentage of the halo cells to be communicated and the iterations in the equation solvers, diffusion and transport schemes should be kept low. The computations with numerous verification test cases and larger real-world models confirm that the presented solution allows an efficient and accurate *UnTRIM* application on the state-of-the art high performance computers for computationally intensive hydraulic engineering applications.

Table 2: Speedups relative to 96 processors reached with the model *The Elbe River by Coswig* for 2D and 3D computations

	2D	2D	2D	2D	3D	3D	3D	3D
	adv	adv	nadv	nadv	adv	adv	nadv	nadv
np	2s	0.5s	2s	0.5s	2s	0.5s	2s	0.5s
1	0.76	0.78	0.72	0.74				
2	1.59	1.61	1.51	1.52				
4	3.35	3.38	3.22	3.23				
8	6.75	6.87	6.54	6.64	6.41	6.78	6.17	6.66
16	14.21	14.39	13.92	13.94	12.90	13.59	12.61	13.53
32	30.07	30.45	30.09	29.51	27.66	28.72	26.83	28.53
48	45.88	45.34	45.13	44.24	41.82	42.64	40.94	42.26
64	62.51	62.44	61.42	61.08	59.40	59.98	59.03	59.85
96	96.00	96.00	96.00	96.00	96.00	96.00	96.00	96.00
128	124.93	126.35	124.48	127.95	123.59	122.27	122.46	122.65

References

[1] Casulli, V., 1999. A semi-implicit finite difference method for non-hydrostatic, free surface flows. *International Journal for Numerical Methods in Fluids*, **30**, 425–440.

[2] Casulli, V., and Cattani, E., 1994. Stability, accuracy and efficiency of a semi-implicit method for three-dimensional shallow water flow. *Computers Math. Applic.*, **27**(4), 99–112.

[3] Casulli, V., and Cheng, R., 1992. Semi-implicit finite difference methods for three-dimensional shallow water flow. *International Journal for Numerical Methods in Fluids*, **15**, 629–648.

[4] Casulli, V., and Lang, G., 2004. Mathematical model UnTRIM, validation document. Technical report, Bundesanstalt für Wasserbau, Hamburg. Version June 2004 (1.0), 78 pp., http://www.baw.de/vip/abteilungen/wbk/Methoden/hnm/untrim/PDF/vd-untrim-2004.pdf.

[5] Casulli, V., and Walters, R., 2000. An unstructured grid, three dimensional model based on the shallow water equations. *International Journal for Numerical Methods in Fluids*, **32**, 331–348.

[6] Casulli, V., and Zanolli, P., 2002. Semi-implicit numerical modelling of non-hydrostatic free-surface flows for environmental problems. *Mathematical and Computer Modelling*, **36**, 1131–1149.

[7] Fringer, O., Gerritsen, M., and Street, R., 2006. An unstructured-grid, finite-volume, nonhydrostatic, parallel coastal simulator. *Ocean Modelling*, **14**, 139–173.

[8] Ham, D., Kramer, S., Stelling, G., and Pietrzak, J., 2007. The symmetry and stability of unstructured mesh c-grid shallow water models under the influence of coriolis. *Ocean Modelling*, **16**, 47–60.

[9] Ham, D., Pietrzak, J., and Stelling, G., 2005. A scalable unstructured grid 3-dimensional finite volume model for the shallow water equations. *Ocean Modelling*, **10**, 153–169.

[10] Karypis, G., and Kumar, V., 1998. Multilevel k-way partitioning scheme for irregulat graphs. *Journal of Parallel and Distributed Computing*, **48**(1), 96–129. http://www.cs.umn.edu/˜metis.

[11] Labeur, R., and Pietrzak, J., 2005. A fully three dimensional unstructured grid non-hydrostatic finite element coastal model. *Ocean Modelling*, **10**, 17–28.

[12] Lippert, C., and Sellerhoff, F., 2006. Efficient generation of orthogonal unstructured grids. In *Proceedings of the Seventh International Conference on Hydroscience and Engineering (ICHE 2006)*, Philadelphia, PA, USA.

[13] Message Passing Interface Forum, 1995. *MPI: A Message Passing Interface Standard*. University of Tennessee, Knoxville. Version 1.1, June 12, 1995, 231 pp., http://www-unix.mcs.anl.gov/mpi/.

[14] OpenMP Architecture Review Board, 1999. *OpenMP Fortran Application Program Interface*. Version 1.1 November 1999. 76 pp., http://www.openmp.org/.

[15] Perot, B., 2000. Conservation properties of unstructured staggered mesh schemes. *Journal of Computational Physics*, **159**, 58–89.

[16] Strybny, J., Thorenz, C., Croce, R., and Engel, M., 2006. A parallel 3D free surface Navier-Stokes solver for high performance computing at the German waterways administration. In *Proceedings of the Seventh International Conference on Hydroscience and Engineering (ICHE 2006)*, Philadelphia, PA, USA.

[17] Zhang, Y., Baptista, A., and Myers, E., 2004. A cross-scale model for 3D baroclinic circulation in estuary-plume-shelf systems: I. Formulation and skill assessment. *Continental Shelf Research*, **24**, 2187–2214.