Domain decomposition in shallow-water modelling for practical flow applications

Mart Borsboom[1], Menno Genseberger[1], Bas van ’t Hof[2], and Edwin Spee[1]

1 Introduction

For the simulation of flows in rivers, lakes, and coastal areas for the executive arm of the Dutch Ministry of Infrastructure and the Environment the shallow-water solver SIMONA is being used [1]. Applications range from operational forecasting of flooding of the Dutch coast [3] and big lakes [7], to the assessment of primary water defences (coast, rivers, and lakes). These applications require a robust and efficient modelling framework with extensive modelling flexibility and good parallel performance.

About two decades ago, a parallel implementation of SIMONA was developed [10, 11] based on domain decomposition with maximum overlap. In the same period, non-overlapping domain decomposition with optimized coupling was considered for Delft3D-FLOW [2], a shallow-water solver that is numerically very similar to SIMONA. More recently, ideas of the latter were adapted for incorporation in SIMONA for enhanced modelling flexibility and parallel performance. This will be the subject of the present paper.

The paper is organized as follows. The numerical approach for modelling shallow-water flow as implemented in SIMONA is outlined in section 2. In section 3 we show how domain decomposition has been incorporated and which refinements have been made. The parallel performance of the modified method is illustrated in section 4 for two practical flow problems from civil engineering.

2 ADI-type shallow-water solvers

The shallow-water equations consist of a depth-integrated continuity equation and two horizontal momentum equations. Vertical momentum is replaced by the hydrostatic pressure assumption, i.e., the vertical variation of the pressure is assumed to depend solely on hydrostatic forces as determined by the position of the free surface. For the numerical solution of the shallow-water equations SIMONA applies a so-called alternating direction implicit (ADI) method to integrate the equations numerically in time, using an orthogonal staggered grid with horizontal curvilinear coordinates ξ and η [1].

[1] Deltares, Delft, The Netherlands,
e-mail: {Mart.Borsboom}{Menno.Genseberger}{Edwin.Spee}@deltares.nl
[2] VORtech Computing, Delft, The Netherlands,
e-mail: bas.vanthof@vortech.nl

In the ADI method, each time step is split in two stages of half a time step. In the first stage, the water-level gradient is taken implicitly in the ξ-momentum equation and explicitly in the η-momentum equation. The mass fluxes in the continuity equation are taken implicitly/explicitly in ξ- and η-direction as well, allowing the implicit terms to be combined to uncoupled tridiagonal systems of equations in ξ-direction for the water level at the intermediate time level. In contrast, the evaluation of the horizontal convection terms and viscosity terms are respectively explicit and implicit in the ξ- and η-momentum equation. In the second stage of the time step, the implicit and explicit discretisations are interchanged. For stability, derivatives in vertical direction and the bottom friction term are always integrated implicitly.

The ADI method requires the use of fairly small time steps to avoid excessive splitting errors:

$$\frac{u\Delta t}{\Delta x_\xi} \le O(1)\ ,\ \frac{v\Delta t}{\Delta x_\eta} \le O(1)\ ,\ \frac{\sqrt{gh}\Delta t}{\Delta x_\xi} \le O(10)\ ,\ \text{and}\ \frac{\sqrt{gh}\Delta t}{\Delta x_\eta} \le O(10)\ . \qquad (1)$$

Here, Δx_ξ, Δx_η are the grid sizes and u, v the velocities in the two horizontal curvilinear coordinate directions ξ and η, Δt is the time step, h the local water depth, and g the acceleration due to gravity ($\sqrt{gh}$ is the shallow-water wave celerity). Because of the conditions (1), the discretized equations to be solved have a fairly high diagonal dominance horizontally. This enables the use of semi-explicit iterative methods horizontally, such as red-black Jacobi to solve implicit convection and viscosity. For the same reason, horizontal domain decomposition with explicit coupling, if designed properly, can be very efficient. We remark that in the vertical direction grid sizes $\ll \Delta x_\xi$, Δx_η are used and the systems of equations are much stiffer. Vertical derivatives are therefore always integrated implicitly in time.

3 Domain decomposition techniques for ADI-type shallow-water solvers

About two decades ago, a parallel implementation of SIMONA was developed [10, 11] using a multi-domain version of the ADI method with Dirichlet-Dirichlet coupling and maximum overlap to ensure fast convergence. This approach is still applied in the 2006 version of SIMONA. Later on, for modelling flexibility, the possibility to use different grid resolutions per subdomain has been introduced. For such a situation it is not that easy to deal with an overlap between subdomains. Therefore, the overlap was removed. This concerns the overlap of the physical area of the subdomains, i.e., the area containing the inner grid cells. For the implementation of boundary conditions and coupling conditions, virtual grid cells were added outside the physical areas along boundaries and DD interfaces. So although the subdomains do not overlap, the subdomain grids do. Unfortunately, a Dirichlet-Dirichlet coupling with minimal overlap (only the virtual grid cells overlap) has a very slow rate of convergence. See also panel (b) of Fig. 1. By re-using ideas from a non-overlapping domain decomposition approach with optimized coupling for Delft3D-FLOW [2], the good convergence behavior has been restored. This approach is implemented since 2010 in SIMONA.

To illustrate how convergence errors due to domain decomposition propagate from one subdomain to another in a multi-domain ADI-type shallow-water solver, we consider a uniform grid of size Δx_ξ, a uniform depth h, and assume a small surface elevation ζ and flow velocity u. The implicit systems in the ξ-direction at the first half time step from t^n to $t^{n+1/2}$ are then of the form (discretized continuity equation and momentum equation):

$$\frac{\zeta_i^{n+1/2}-\zeta_i^n}{\Delta t/2}+h\frac{u_{i+1/2}^{n+1/2}-u_{i-1/2}^{n+1/2}}{\Delta x_\xi}=\dots\ ,\quad \frac{u_{i+1/2}^{n+1/2}-u_{i+1/2}^{n}}{\Delta t/2}+g\frac{\zeta_{i+1}^{n+1/2}-\zeta_i^{n+1/2}}{\Delta x_\xi}=\dots\ . \tag{2}$$

At the second half time step from $t^{n+1/2}$ to t^{n+1}, equations in η-direction (j-index) are obtained. By eliminating $u_{i+1/2}^{n+1/2}$, the two equations (2) can be combined to:

$$\zeta_i^{n+1/2}-CFL^2\left(\zeta_{i+1}^{n+1/2}-2\,\zeta_i^{n+1/2}+\zeta_{i-1}^{n+1/2}\right)=\dots\ , \tag{3}$$

with CFL number $CFL=\sqrt{gh}\Delta t/(2\Delta x_\xi)$.

To study the behavior of (3) in a DD framework, we consider the homogeneous equation that is satisfied by the DD convergence error $\delta\zeta_i^{n+1/2,m}=\zeta_i^{n+1/2,m}-\zeta_i^{n+1/2}$, with $\zeta_i^{n+1/2}$ the solution that is sought and $\zeta_i^{n+1/2,m}$ its iteratively determined approximation at iteration m:

$$\delta\zeta_i^{n+1/2,m}-CFL^2\left(\delta\zeta_{i+1}^{n+1/2,m}-2\,\delta\zeta_i^{n+1/2,m}+\delta\zeta_{i-1}^{n+1/2,m}\right)=0\ . \tag{4}$$

The inhomogeneous perturbation of $\delta\zeta_i^{n+1/2,m}$ comes from the boundaries of the subdomains where information is updated explicitly (Schwarz algorithm). Equation (4) determines how that information spreads across a subdomain and reaches the opposite subdomain boundary. This becomes clear from the solution of (4), which is of the form:

$$\delta\zeta_i^{n+1/2,m}=C_{LR}\lambda^i+C_{RL}\lambda^{-i}\ , \tag{5}$$

with $\lambda=(CFL^2+1/2-\sqrt{CFL^2+1/4})/CFL^2$. The solution consists of the superposition of two modes: one decaying from left to right and one decaying from right to left. Panel (a) in Fig. 1 illustrates this for a subdomain of 8 grid cells at $CFL=2$ (green), $CFL=5$ (red), and $CFL=10$ (blue). For $CFL\ll 1$, we have $\lambda\approx 1/CFL^2$. At such a high decay rate per grid cell, which is due to the large diagonal dominance of (4), a Dirichlet-Dirichlet coupling is efficient. For $CFL\gg 1$, however, we have $\lambda\approx 1-CFL^{-1}$ and hence a much lower decay rate. A Dirichlet-Dirichlet coupling is then not efficient anymore, unless a large overlap is used to compensate for the low decay rate. This is illustrated in panel (b) and (c) of Fig. 1.

A much larger DD convergence speed is obtained by only transfering from left to right (right to left) the information that is moving in that direction. This is realized by the coupling:

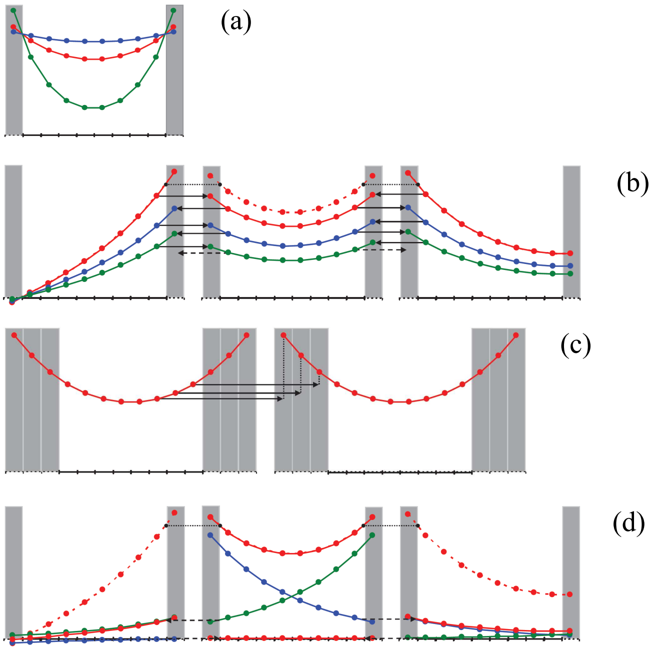

Fig. 1 Behavior of convergence error $\delta\zeta_i^{n+1/2,m}$ in subdomains consisting of 8 inner grid cells (white) and 1, 2, or 3 added virtual grid cells (grey) that overlap with inner grid cells of neighbouring subdomains: (a) inside a subdomain at $CFL = 2$ (green), $CFL = 5$ (red), and $CFL = 10$ (blue); (b) across 3 subdomains at $CFL = 5$ with Dirichlet boundary condition left, Neumann boundary condition right, and multiplicative Schwarz Dirichlet-Dirichlet coupling with minimal overlap in between (red, blue, green indicate subsequent DD iterations); (c) enhancement of DD convergence with Dirichlet-Dirichlet coupling when using a larger overlap (increasingly longer dotted lines indicate error reduction for 1-, 2-, and 3-cell overlap); (d) across 3 subdomains with optimized multiplicative Schwarz based on the decomposition of the convergence error (red lines) in its two solution modes (blue and green lines), cf. (5). Note that in (b, c) the arrows indicate the transfer of Dirichlet values from an inner grid cell to a virtual grid cell; in (d) the arrows indicate the transfer of optimized coupling information from interface to interface.

$$\begin{aligned}(CFL+1/2)\delta\zeta_{i_R}^{n+1/2,m+1} &- (CFL-1/2)\delta\zeta_{i_R+1}^{n+1/2,m+1} \\ = (CFL+1/2)\delta\zeta_{i_L-1}^{n+1/2,m} &- (CFL-1/2)\delta\zeta_{i_L}^{n+1/2,m} \,, \end{aligned} \tag{6}$$

with i_R the index of the left virtual grid cell of the subdomain right of the DD interface under consideration, and with i_L the index of the right virtual grid cell of the subdomain left. Notice the explicit nature of the coupling: the solution of domain L at previous iteration m determines the value (right-hand side of (6)) of the condition to be imposed at the left boundary of domain R during next iteration $m+1$ (left-hand side of (6)). An equivalent procedure is used for the transfer of coupling information in the other direction, from domain R to domain L.

Panel (d) of Fig. 1 illustrates the high DD convergence rate that can be obtained with an optimized coupling; the convergence speed is about as high as would be obtained with a Dirichlet-Dirichlet coupling with maximum overlap (of half a subdomain, cf. panel (c)). However, because of the overlap, the amount of work per iteration in the latter would be twice as large. Furthermore, as mentioned before, it can not be combined easily with local grid refinements for which the grid cells in the overlap do not coincide, contrary to the situation in panel (c).

The fast DD convergence speed that for diagonally dominant problems can be obtained with an optimized explicit local DD coupling (optimized Schwarz), and the link with absorbing boundary conditions, is well known [8, 5, 4, 9, 6]. Because the splitting applied in the ADI method leads to independent 1D problems, we have the advantage that the optimized coupling can not only easily be determined for constant Δx_ξ and h, as we did here, but also for the general case, by means of the LU decomposition of the resulting tridiagonal systems that are of the form (3), but with space- and time-varying coefficients. The bidiagonal L-matrices describe the decay of the solution in increasing i- (or j-) direction. Their last rows determine the combinations of pairs of ζ's at the subdomain interface (one ζ in a virtual grid cell, the other ζ in the adjacent inner grid cell) that do not specify this part of the solution, and hence only specify solution modes decaying in decreasing i- (or j-) direction. Transfering these combinations in decreasing i- (or j-) direction across DD interfaces (the variable-coefficient generalization of (6)) therefore ensures maximum DD convergence speed. Likewise for the bidiagonal U-matrices and the exchange of coupling information in the other direction.

4 Applications

There are many application areas of SIMONA. Here we present two examples. First we show the effect of the optimized coupling without overlap for a schematic model of the river Waal in the Netherlands. This schematic model has a simple geometric shape such that load balancing is straightforward. Second we show the parallel performance of the approach for DSCM, a huge real-life hydrodynamic model in which both load balancing and number of unknowns are an issue.

For the experiments we considered the following hardware:

- H4 linux-cluster at Deltares, nodes interconnected with Gigabit Ethernet, each node contains 1 AMD dual-core Athlon X2 5200B processor with 2.7 GHz per core,
- H4+ linux-cluster at Deltares, nodes interconnected with Gigabit Ethernet, each node contains 1 Intel quad-core i7-2600 processor with 3.4 GHz per core and hyperthreading (so effectively 8 threads are used on 4 cores), and
- Lisa linux-cluster at SURFsara, nodes interconnected with Infiniband, each node contains 2 Intel quad-core Xeon L5520 processors with 2.3 GHz per core.

On the H4 linux-cluster both the 2006 and 2010 version of SIMONA were used. On the H4+ and Lisa linux-cluster the 2010 version of SIMONA was used. Recall (see section 3) that the 2006 version uses Dirichlet-Dirichlet coupling and maximum overlap where the 2010 version uses optimized coupling without overlap.

4.1 Schematic model of river Waal

To study the effect of lowering the groynes on design flood level, in [12] a schematised river reach was used that was based on characteristic dimensions of river Waal in the Netherlands. Here, for the performance tests we will use the detailed model of [12] in which the groynes are represented as bed topography (see Fig. 2).

The detailed model is a symmetrical compound channel of 30 km length including floodplain (width of 1200 m) and main channel (width of 600 m). We apply a depth averaged version of SIMONA. The floodplain is schematised with grid cells of 2 m x 4 m and the main channel with grid cells of 2 m x 2 m, resulting in more than 9 million unknowns. A time step of 0.015 minutes is used, resulting in 12000 time steps for the 3 hour simulation that we consider here for the performance tests.

From Fig. 2 it can be observed that, in general, SIMONA scales well. Furthermore, on the H4 linux-cluster the 2010 version of SIMONA is about 20-30 % faster than the 2006 version. This additional work can be explained from the overlap in the 2006 version which is not in the 2010 version (see section 3). The difference in performance for the 2010 version of SIMONA on H4, Lisa, and H4+ linux-cluster is because of the different hardware.

4.2 Next generation Dutch Continental Shelf Model (DCSM)

The current generation of nested SIMONA models used for predicting water levels along the Dutch coast in an operational mode (see [3]) already require high performance computing. At the Lisa linux-cluster parallel performance of the 2010 version of SIMONA was tested for a next generation version of the DCSM (North Sea and adjacent region of the North Atlantic). This 3D (10 layer) higher resolution model includes salinity and temperature stratification processes which are essential for simulating among others the spread of the freshwater Rhine plume along the Dutch coast. This new model requires a huge computational effort but simulation times cannot increase for operational purposes. Although the North Sea model has an irregular geometry which is not ideal for scalability, performance tests at Lisa showed linear scalability up to 100 processors. The left panel of Fig. 3 shows the partitioning of the domain in 96 subdomains of (about) the same number of grid cells that is obtained by applying orthogonal recursive bisection (ORB). The right panel shows the parallel performance on the Lisa linux-cluster as a function of the number of subdomains and cores, for partitionings in strips and by means of ORB. The results show an optimal speed-up for the ORB partitioning and a small decay in performance for the larger strip decompositions. The latter is due to the shape of the strips. The strips become very thin with widths of less than a dozen grid cells as the number of domains increases, which affects the validity of the applied local coupling optimization.

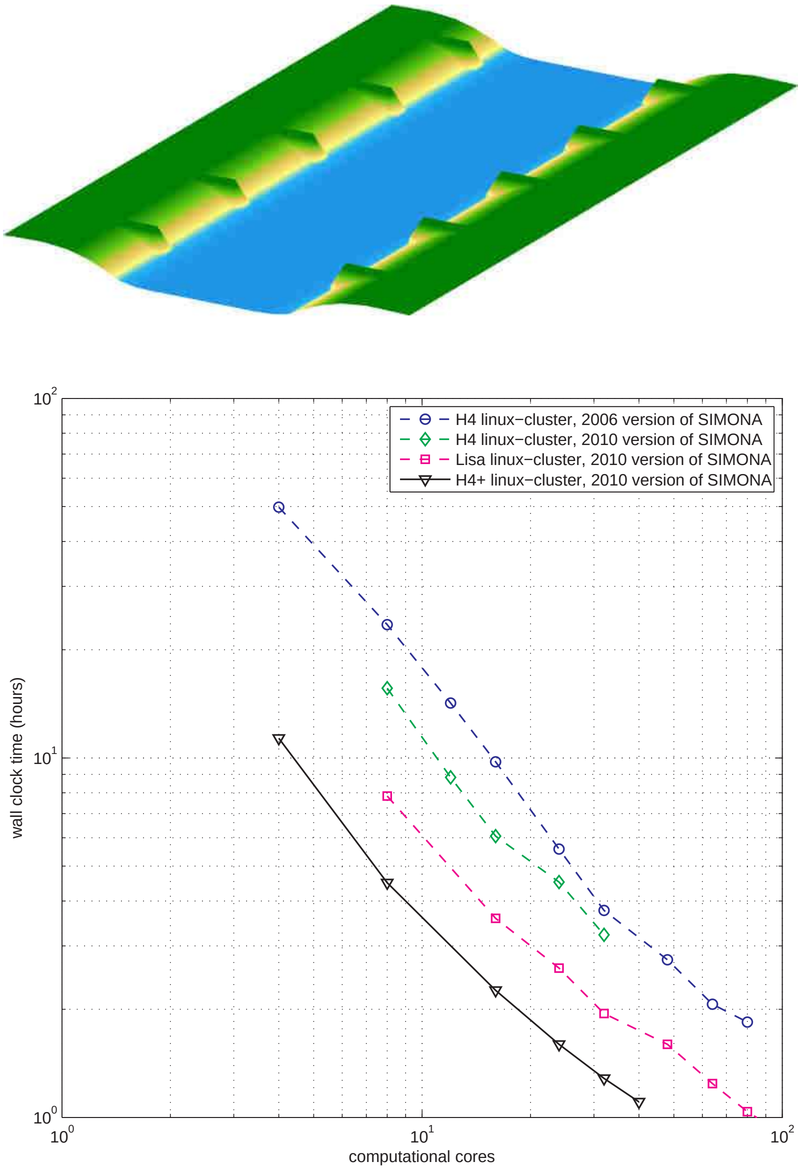

Fig. 2 Schematic model of river Waal: an excerpt of the model including part of the floodplain (top), parallel performance for different versions of SIMONA and on different hardware (bottom).

Acknowledgements We thank SURFsara (www.surfsara.nl) for their support in using the Lisa linux-cluster.

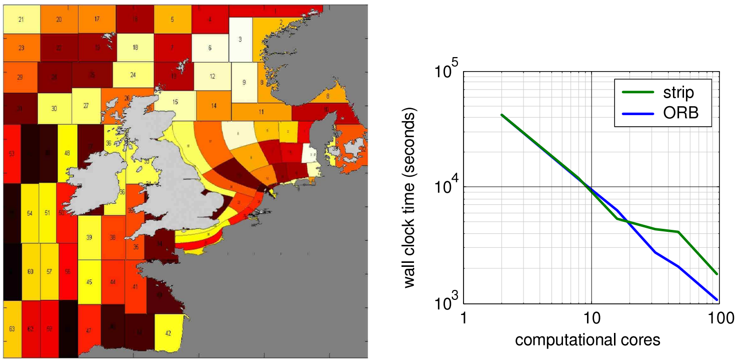

Fig. 3 DCSM. Left: partitioning of computational domain in 96 subdomains using the orthogonal recursive bisection (ORB) method. Right: parallel performance on Lisa linux-cluster for partitionings in vertical strips and ORB partitionings.

References

1. SIMONA WAQUA/TRIWAQ - two- and three-dimensional shallow-water flow model. (2012) URL {http://apps.helpdeskwater.nl/downloads/extra/simona/release/doc/techdoc/waquapublic/sim1999-01.pdf}
2. De Goede, E.D., Groeneweg, J., Tan, K.H., Borsboom, M.J.A., Stelling, G.S.: A domain decomposition method for the three-dimensional shallow water equations. Simulation Practice and Theory **3**, 307–325 (1995)
3. De Kleermaeker, S.H., Verlaan, M., Kroos, J., Zijl, F.: A new coastal flood forecasting system for the Netherlands. In: T. Van Dijk (ed.) Hydro12 Conference, Rotterdam, The Netherlands, 2012. Hydrographic Society Benelux (2012). URL {http://proceedings.utwente.nl/246}
4. Dolean, V., Lanteri, S., Nataf, F.: Convergence analysis of additive Schwarz for the Euler equations. Appl. Numer. Math. **49**(2), 153–186 (2004)
5. Engquist, B., Zhao, H.K.: Absorbing boundary conditions for domain decomposition. Appl. Numer. Math. **27**(4), 341–365 (1998)
6. Gander, M.J.: Optimized Schwarz methods. SIAM J. Numer. Anal. **44**(2), 699–731 (2006)
7. Genseberger, M., Smale, A., Hartholt, H.: Real-time forecasting of flood levels, wind driven waves, wave runup, and overtopping at dikes around Dutch lakes. In: F. Klijn, T. Schweckendiek (eds.) 2nd European Conference on FLOODrisk Management, Rotterdam, The Netherlands, 2012, Comprehensive Flood Risk Management, pp. 1519–1525. Taylor & Francis Group (2013)
8. Japhet, C., Nataf, F., Roux, F.X.: Extension of a coarse grid preconditioner to non-symmetric problems. In: J. Mandel, C. Farhat, X.C. Cai (eds.) Domain Decomposition Methods 10, *Contemporary Mathematics*, vol. 218, pp. 279–286. AMS (1998)
9. Maday, Y., Magoulès, F.: Absorbing interface conditions for domain decomposition methods: A general presentation. Comput. Meth. Appl. Mech. Eng. **195**(29–32), 3880–3900 (2006)
10. Roest, M.R.T.: Partitioning for parallel finite difference computations in coastal water simulation. Ph.D. thesis, Delft University of Technology, The Netherlands (1997)
11. Vollebregt, E.A.H.: Parallel software development techniques for shallow water models. Ph.D. thesis, Delft University of Technology, The Netherlands (1997)
12. Yossef, M.F.M., Zagonjolli, M.: Modelling the hydraulic effect of lowering the groynes on design flood level. Tech. Rep. 1002524-000, Deltares (2010)