John von Neumann Institute for Computing

Parallel Simulation of Tsunamis Using a Hybrid Software Approach

X. Cai, G.K. Pedersen, H.P. Langtangen, S. Glimsdal

published in

Parallel Computing:
Current & Future Issues of High-End Computing,
Proceedings of the International Conference ParCo 2005,
G.R. Joubert, W.E. Nagel, F.J. Peters, O. Plata, P. Tirado, E. Zapata (Editors),
John von Neumann Institute for Computing, Jülich,
NIC Series, Vol. **33**, ISBN 3-00-017352-8, pp. 383-390, 2006.

http://www.fz-juelich.de/nic-series/volume33

Parallel Simulation of Tsunamis Using a Hybrid Software Approach

X. Cai[ab], G. K. Pedersen[c], H. P. Langtangen[ab], S. Glimsdal[ab]

[a]Simula Research Laboratory, P.O. Box 134, N-1325 Lysaker, Norway

[b]Department of Informatics, University of Oslo, P.O. Box 1080, Blindern, N-0316 Oslo, Norway

[c]Department of Mathematics, Mechanics Division, University of Oslo, P.O. Box 1053, Blindern, N-0316 Oslo, Norway

Simulation of tsunamis in large ocean domains, such as the Indian Ocean, presents a huge computational and software challenge, which the traditional parallel software alone can not meet. To effectively use the computational resources, we should allow different mathematical models, different numerical methods, and different mesh types/resolutions in different areas of the vast ocean domain. This ensures that complex mathematical models and high mesh resolution are only used where necessary. Such a parallel hybrid tsunami simulator calls for a very flexible software approach that can mix different models, methods and meshes, maybe even incorporate "alien software". This paper thus explains how this can be achieved by combining overlapping domain decomposition and object-oriented programming. Some preliminary simulation results of the Indian Ocean Tsunami are also provided to demonstrate the applicability of the proposed software approach.

1. Introduction

The tragic event of the Indian Ocean Tsunami on December 26, 2004 has once again grimly reminded us about the importance of tsunami modeling and simulation. This challenging research field spans a wide spectrum and involves many interplayed topics, such as earthquake modeling, ocean wave propagation and coastal modeling. The main focus of the present paper is to investigate how to effectively simulate the propagation of tsunami over a vast ocean. Although the topic of water wave propagation has been subject to extensive research, the particular case of simulating the Indian Ocean Tsunami presents a new challenge because of the huge size of the computations. Moreover, many complex features need to be considered in ocean wave propagation, such as the locally rapidly changing bathymetry, dispersion, nonlinear effects, and complicatedly shaped coastlines.

To obtain a balance between numerical accuracy and computational efficiency, we should only apply advanced numerical techniques and high mesh resolutions to small areas where necessary, while resorting to simpler numerical techniques and coarser meshes in the remaining areas. In terms of mathematical models, this means a choice between the most widely used linear wave equations and the more complex Boussinesq wave equations, which involve both weak dispersion and nonlinear effects. Perhaps also the Navier-Stokes equations may sometimes be used in certain small areas. In terms of numerics, there is a choice between finite differences, finite volumes, and finite elements. The finite difference method (FDM) results in the fastest computations, at least on a uniform mesh of rectangular shape. The finite volume method (FVM) or the finite element method (FEM), on the other hand, run slower but are better adapted to unstructured meshes and resolution variations needed for treating, e.g., areas of rapidly changing ocean bottom topography.

Considering the fact that there already exist many software packages for wave simulation, and that parallel computing must be used for detailed ocean modeling, an ideal tsunami simulator should be built by a hybrid software approach. More specifically, the entire ocean domain is divided into many subdomains, and independent solvers are assigned to the subdomains. Each subdomain solver has its own mathematical model, together with the matching numerical method and subdomain mesh. Therefore, different subdomains may use different mathematical models, different numerical methods, different meshes and even different software! Of course, to make the above hybrid divide-and-conquer simulator to work, we need a numerical framework overseeing all the subdomains. This can be achieved by the overlapping domain decomposition (DD) strategy [5,15], which was originally designed for using a same mathematical model everywhere, but can be extended to act as an "umbrella" for the subdomains in the hybrid tsunami simulator. The communication between the subdomains is controlled by the DD framework and implemented as exchanging messages using MPI [7,10]. In respect of programming, the framework of DD can be implemented in an object-oriented style [1], making it easier to encompass different types of subdomain solvers and to adopt "alien software" when necessary.

The remaining text of the paper is organized as follows. Section 2 explains two mathematical models commonly used for simulating wave propagation and the additive Schwarz scheme needed as the numerical foundation of the parallel hybrid tsunami simulator. Section 3 then concentrates on the implementation aspect, based on object-oriented programming. Thereafter, Section 4 presents some preliminary parallel simulation results of the Indian Ocean Tsunami. Finally, some concluding remarks and comments about future work are given in Section 5.

2. Mathematics and Numerics

2.1. Boussinesq Water Wave Equations

The nonlinear Boussinesq water wave equations can be used to simulate ocean waves; see e.g. [14, 11,2,16]. In comparison with the standard linear wave equations, the Boussinesq equations can model weakly dispersive and nonlinear waves. There exist several variants of the Boussinesq equations, among which we will consider the following two coupled partial differential equations:

$$\frac{\partial \eta}{\partial t} + \nabla \cdot (H + \alpha\eta)\nabla\phi + \epsilon H\left(\frac{1}{6}\frac{\partial \eta}{\partial t} - \frac{1}{3}\nabla H \cdot \nabla\phi\right)\nabla H = 0, \tag{1}$$

$$\frac{\partial \phi}{\partial t} + \frac{\alpha}{2}\nabla\phi \cdot \nabla\phi + \eta - \frac{\epsilon}{2}H\nabla \cdot \left(H\nabla\frac{\partial\phi}{\partial t}\right) + \frac{\epsilon}{6}H^2\nabla^2\frac{\partial \phi}{\partial t} = 0, \tag{2}$$

where the primary unknowns are the surface elevation $\eta(x, y, t)$ and the depth averaged velocity potential $\phi(x, y, t)$. In the above equations $H(x, y)$ denotes the water depth, and the constants ϵ and α are used to control the magnitude of dispersion and nonlinearity, respectively. With $\epsilon = \alpha = 0$ we recover the standard linear wave equations from (1)-(2).

A standard numerical strategy for solving (1)-(2) first carries out the temporal discretization, with centered differences on a staggered grid in time [13]:

$$\frac{\eta^\ell - \eta^{\ell-1}}{\Delta t} + \nabla \cdot \left(\left(H + \alpha\frac{\eta^{\ell-1} + \eta^\ell}{2}\right)\nabla\phi^{\ell-\frac{1}{2}} + \epsilon H\left(\frac{1}{6}\frac{\eta^\ell - \eta^{\ell-1}}{\Delta t} - \frac{1}{3}\nabla H \cdot \nabla\phi^{\ell-\frac{1}{2}}\right)\nabla H\right) = 0,$$

$$\frac{\phi^{\ell+\frac{1}{2}} - \phi^{\ell-\frac{1}{2}}}{\Delta t} + \frac{\alpha}{2}\nabla\phi^{\ell-\frac{1}{2}} \cdot \nabla\phi^{\ell+\frac{1}{2}} - \frac{\epsilon H}{2}\nabla \cdot \left(H\frac{\nabla\phi^{\ell+\frac{1}{2}} - \nabla\phi^{\ell-\frac{1}{2}}}{\Delta t}\right) + \frac{\epsilon H^2}{6}\frac{\nabla^2\phi^{\ell+\frac{1}{2}} - \nabla^2\phi^{\ell-\frac{1}{2}}}{\Delta t} = -\eta^\ell.$$

Here, index ℓ denotes the discrete time levels and Δt is the time step size. Note that η is sought at integer time levels (ℓ) and ϕ is sought at half-integer time levels ($\ell + \frac{1}{2}$). The remaining part of the numerical scheme is to carry out the spatial discretization of the above two semi-discretized equations at each time step, using FDM or FEM, and solve for η^ℓ and $\phi^{\ell+\frac{1}{2}}$. The readers are referred to [9,4] for more details.

2.2. The Parallel Multi-Subdomain Strategy

As we can see in the preceding text, the computational task at time step ℓ is to find η^{ℓ} and $\phi^{\ell+\frac{1}{2}}$ based on the solutions from the previous time step: $\eta^{\ell-1}$ and $\phi^{\ell-\frac{1}{2}}$. To incorporate parallelism, we use the additive Schwarz scheme [5,15], which is an overlapping DD method. The entire ocean domain Ω is first decomposed into a set of *overlapping* subdomains Ω_s, $1 \leq s \leq P$. Overlapping zones are present between neighboring subdomains. In the context of solving discretized Boussinesq equations at time step ℓ, the additive Schwarz scheme is transformed into the following parallel numerical strategy:

- Set $\eta_s^{\ell,0} = \eta^{\ell-1}|_{\Omega_s}$, where $\eta^{\ell-1}|_{\Omega_s}$ denotes the restriction of the global solution $\eta^{\ell-1}$ (from time step $\ell - 1$) onto subdomain s.

- Carry out the following Schwarz iterations for $k = 1, 2, 3, \ldots$ until convergence of η among the subdomains:

  1. On each subdomain find an improved local solution $\eta_s^{\ell,k}$ based on $\eta_s^{\ell,k-1}$ and $\phi^{\ell-\frac{1}{2}}|_{\Omega_s}$.
  2. Compose a temporary global solution $\eta^{\ell,k}$ by "sewing together" the latest subdomain solutions $\left\{\eta_s^{\ell,k}\right\}$. In each overlapping zone between two or more neighboring subdomains, averaging between different subdomain solutions is enforced. In case neighboring subdomains have different mesh resolutions in an overlapping zone, interpolation between the subdomain meshes is used in the averaging.

- Set $\phi_s^{\ell+\frac{1}{2},0} = \phi^{\ell-\frac{1}{2}}|_{\Omega_s}$ and carry out the Schwarz iterations with respect to ϕ as above.

It should be noted that the above numerical strategy extends the multi-subdomain strategy proposed in [4]. Here, the adopted mathematical model and/or numerical method may differ from subdomain to subdomain, so are the type and resolution of the subdomain meshes. During each Schwarz iteration, the process of solving $\eta_s^{\ell,k}$ and $\phi_s^{\ell+\frac{1}{2},k}$ on subdomain s is decided by the subdomain solver independently. For example, some subdomains may adopt the linear wave equations, i.e., $\epsilon = \alpha = 0$ in (1)-(2), and thus find $\eta_s^{\ell,k}$ and $\phi_s^{\ell+\frac{1}{2},k}$ by an explicit updating scheme, whereas other subdomains may consider dispersion and/or nonlinearity in the Boussinesq equations and thus need an implicit solution scheme. The above parallel multi-subdomain strategy also differs from the classical additive Schwarz method [5,15] in that there does not always exist a global linear system coupling all $\eta_s^{\ell,k}$ or $\phi_s^{\ell+\frac{1}{2},k}$.

Although the subdomain solvers are mostly independent of each other, exchange of subdomain solutions within the overlapping zones is necessary for obtaining global convergence. Thus a global administrator is needed to synchronize the pace of the subdomain solvers during the task of "sewing together" subdomain solutions at the end of each Schwarz iteration. Since different subdomains normally reside on different processors, the global administrator initiates the inter-processor communication in form of message passing. No subdomain is allowed to proceed to the next Schwarz iteration before all its neighbors have received needed information.

Checking the convergence of the Schwarz iterations is another major task of the global administrator. More specifically, after the message passing phase is finished at the end of each Schwarz iteration, each subdomain checks the difference between $\eta^{\ell,k}|_{\Omega_s}$ and $\eta^{\ell,k-1}|_{\Omega_s}$ (recall that $\eta^{\ell,k}$ is the result of "sewing together" $\eta_s^{\ell,k}$ from all the subdomains). If $\left\|\eta^{\ell,k}|_{\Omega_s} - \eta^{\ell,k-1}|_{\Omega_s}\right\|$ is small enough, subdomain s sends a flag indicating local convergence to the global administrator. Otherwise a flag of no local convergence is sent. It is only when *all* the subdomains have reported local convergence that the global administrator deems that global convergence is reached and stops the Schwarz iterations.

3. A Parallel Hybrid Tsunami Simulator

3.1. Making Use of a Generic DD Framework

As has been explained in Section 2.2, the multi-subdomain parallelization strategy uses additive Schwarz iterations as the numerical foundation. It should be noted that such a parallelization strategy is not only applicable to tsunami simulation, but is generic for building many other parallel partial differential equation solvers. Therefore, object-oriented programming has been adopted to implement a parallel DD framework, reusable for many occasions. For details we refer the readers to [3], and it suffices for the present paper to say that the global administrator (see Section 2.2) can be implemented as class `Administrator` and a generic subdomain solver as class `SubdomainSolver`. The essence is that common tasks, such as domain partitioning, inter-subdomain communication and control of Schwarz iterations, are implemented as member functions in `Administrator`, whereas `SubdomainSolver` defines a set of virtual member functions constituting a generic interface of any particular subdomain solver. MPI is used inside `Administrator` for inter-subdomain communication, such that the generic DD framework is portable on all parallel computers.

3.2. Coupling FDM and FEM in a Hybrid Simulator

We have, as the starting point for our parallel hybrid tsunami simulator, two different pieces of serial software: a flexible Boussinesq solver written in C++ and a legacy Fortran 77 code. The C++ Boussinesq solver is implemented as class `Boussinesq` in the Diffpack programming environment [6,12]. This C++ solver uses finite elements and handles both unstructured and uniform meshes. When dispersion and/or nonlinearity are considered in (1)-(2), the two resulting linear systems at each time step can be solved by a variety of linear solvers provided by Diffpack. The legacy Fortran 77 code is a set of subroutines which are much less flexible in that only uniform meshes are allowed, the equations are discretized by FDM, and the linear solver is fixed as the alternating line-version of the SSOR method. However, the main advantage of the Fortran 77 code is its computational efficiency. The code is also reliable and well tested over two decades.

To incorporate both serial Boussinesq solvers as subdomain solvers into a parallel hybrid tsunami simulator, we have created two light-weight new classes:

`SubdomainBQFEMSolver` and `SubdomainBQFDMSolver`

Here, class `SubdomainBQFEMSolver` is implemented as subclass of both `SubdomainSolver` and `Boussinesq`, so that it inherits the computational functionality from `Boussinesq` while becoming recognizable by the generic `Administrator` as a subdomain solver in the generic DD framework. Similarly, class `SubdomainBQFDMSolver` is derived from `SubdomainSolver` and acts as a wrapper of the Fortran 77 subroutines. Finally, another new class `HybridBQSolver` is derived as subclass of `Administrator`, so that some tsunami specific functionality can be added on top of generic DD functionality.

4. Preliminary Results of Indian Ocean Tsunami

In this section we present some preliminary simulation results of the Indian Ocean Tsunami on December 26, 2004. It should be emphasized that the following results are only meant to demonstrate the applicability of the hybrid software approach to tsunami simulation. Only relatively low mesh resolution has been used so far. Ongoing research is currently testing the suitable mathematical model and mesh resolution in different regions of the Indian Ocean, in preparation of doing really large-scale parallel hybrid tsunami simulations.

4.1. The Ocean Domain

Our computational domain Ω covers the entire Indian Ocean from the coastlines of Sumatra in the west to the African coastlines in the east. The size of Ω is 8096.08km$\times$$4777.65$km. Figure 1 shows a plot of the ocean bottom topography $H(x, y)$, built from publicly available bathymetry data.

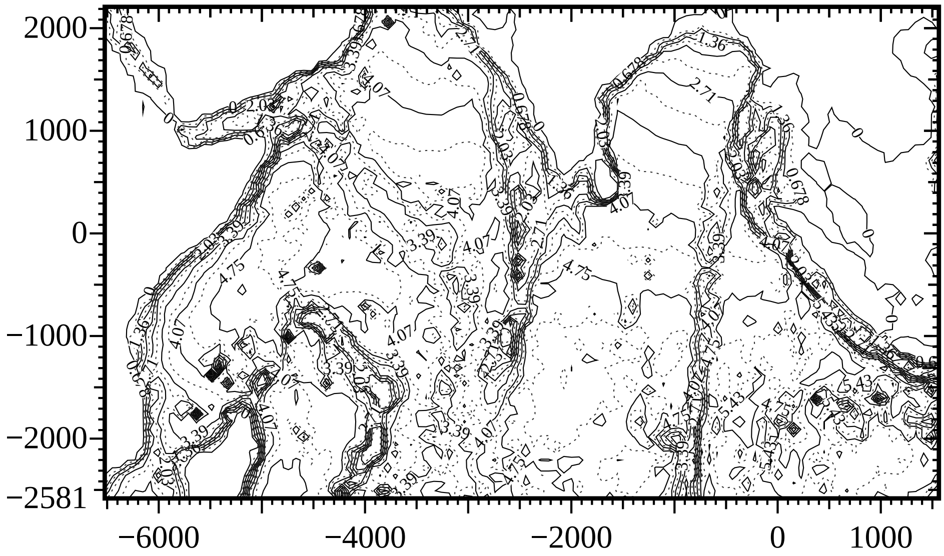

Figure 1. Contour curves representing the water depth (in kilometers) of the entire solution domain. The position $(0, 0)$ represents the epicenter of the submarine earthquake that generated the tsunami.

4.2. Simulation on a Global Uniform Mesh

As a simple test case we use a coarse uniform 1093×645 mesh, where each cell is approximately 7.4km$\times$7.4km. The entire solution domain is divided into $4 \times 4 = 16$ subdomains, which all use the Fortran 77 FDM code as the subdomain solver. Only dispersion is considered, i.e., $\epsilon = 1$ and $\alpha = 0$ in (1)-(2). Figure 2 shows a particular initial condition (see [8]) and two snapshots of the simulated η solution in a zoomed-in region around the epicenter.

4.3. Adaptive Mesh Refinement in the Malacca Strait

Close to the epicenter of the earthquake, such as in the Malacca Strait, the water is extremely shallow, so higher mesh resolution is obviously needed. To test this possibility, we carry out local adaptive mesh refinement in a focused rectangular region around the epicenter. The left plot in Figure 3 shows a zoomed-in plot of the resulting unstructured finite element mesh. The focused region after the adaptive mesh refinement is further partitioned into 17 smaller subdomains that use `SubdomainBQFEMSolver`. The remaining 15 subdomains use `SubdomainBQFDMSolver`, as in Section 4.2. Inside the overlapping zones, the mesh points of the FEM subdomains may not match those of the neighboring FDM subdomains, thus interpolation is required when `HybridBQSolver` "sews together" the subdomain solutions at the end of each Schwarz iteration, see Section 2.2. A

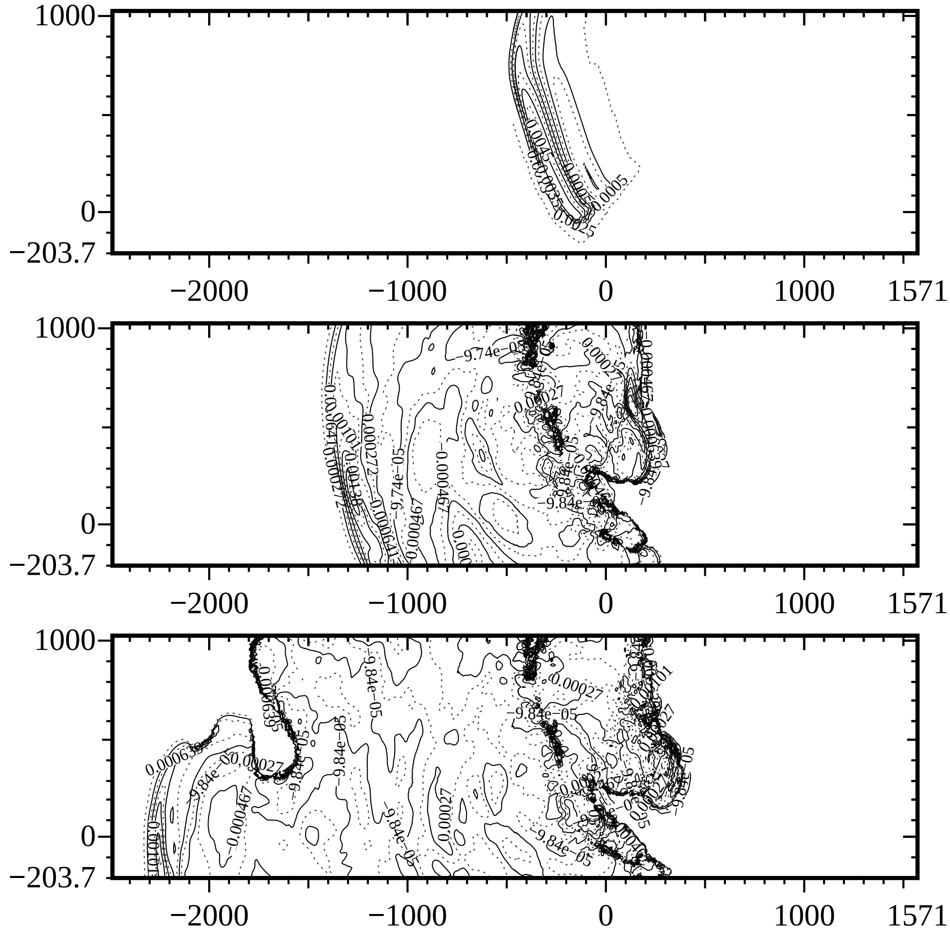

Figure 2. Contour curves of the η solution that are zoomed into a region around the epicenter. Top plot: initial condition. Middle plot: simulated η after approximately 1.4 hours. Bottom plot: simulated η after approximately 2.8 hours.

new simulation is thus run using this setup of hybrid subdomain solvers. Figure 4 shows a snapshot of η approximately 2.8 hours after the earthquake. In comparison with the bottom plot in Figure 2, we can see that adaptively refined finite element meshes produce more local details of η around the epicenter.

5. Concluding Remarks

The simulations reported in Section 4 are rather a proof of concept for the hybrid tsunami simulator. They demonstrate that the resulting parallel simulator is capable of adopting different numerical methods and subdomain meshes in different areas of the ocean domain. We emphasize once again that overlapping domain decomposition and object-oriented programming are the two main ingredients in the hybrid simulator. Meaningful simulations will be carried out in the future using a much finer mesh resolution than that used in Section 4. Nevertheless, for most areas where uniform subdomain meshes are appropriate, FDM should be used for the computational efficiency. For other areas, such as in shallow regions and near the coastlines, locally unstructured finite element fine meshes must be used, thus requiring FEM in the subdomain solvers.

Since a FEM solver is typically an order of magnitude slower than a FDM solver, more (and

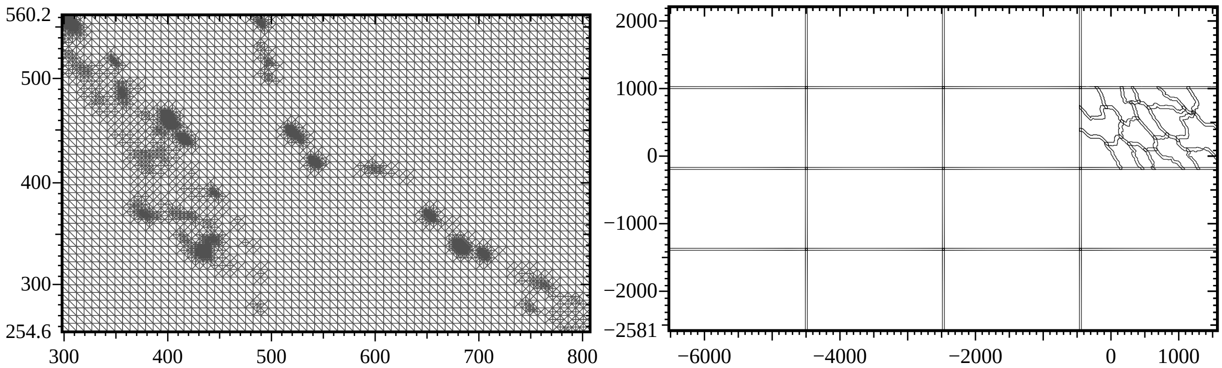

Figure 3. Adaptively refined finite element mesh in a zoomed-in area in the Malacca Strait, and an unstructured repartitioning of the region where adaptive mesh refinement has been carried out.

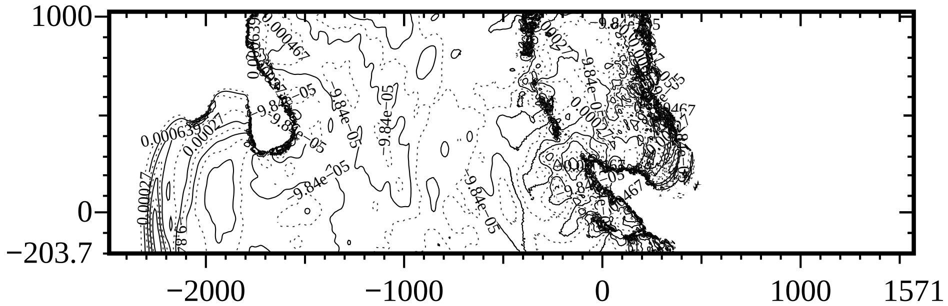

Figure 4. Contour curves of the simulated η solution, zoomed into an area around the epicenter; approximately 2.8 hours after the earthquake.

smaller) subdomains should be used in areas with locally refined meshes, see the right plot in Figure 3. This will help to maintain a reasonable level of load balance between the subdomains. More precisely, a few test time steps can be run on a relatively small number of processors and the CPU time is measured on each subdomain. The CPU time ratio between a FEM subdomain and a standard FDM subdomain gives the number of smaller subdomains into which the FEM subdomain should be further decomposed. When all the FEM subdomains are further decomposed, production simulations can be run on the newly extended set of subdomains, which may differ greatly in terms of area size and number of mesh points but are relatively balanced with respect to the computational speed.

References

[1] J. J. Barton and L. R. Nackman. *Scientific and Engineering C++ – An Introduction with Advanced Techniques and Examples*. Addison-Wesley, 1994.

[2] Marine Accident Investigation Branch. Report on the investigation of the man overboard fatality from the angling boat Purdy at Shipwash Bank off Harwich on 17 July 1999. Technical Report 17/2000, Marine Accident Investigation Branch, Carlton House, Carlton Place, Southampton, SO15 2DZ, 2000.

[3] X. Cai. Overlapping domain decomposition methods. In H. P. Langtangen and A. Tveito, editors, *Advanced Topics in Computational Partial Differential Equations – Numerical Methods and Diffpack Programming*, pages 57–95. Springer, 2003.

[4] X. Cai, G. K. Pedersen, and H. P. Langtangen. A parallel multi-subdomain strategy for solving Boussi-

nesq water wave equations. *Advances in Water Resources*, 28:215–233, 2005.

[5] T. F. Chan and T. P. Mathew. Domain decomposition algorithms. In *Acta Numerica 1994*, pages 61–143. Cambridge University Press, 1994.

[6] Diffpack Home Page. http://www.diffpack.com.

[7] Message Passing Interface Forum. MPI: A message-passing interface standard. *Internat. J. Supercomputer Appl.*, 8:159–416, 1994.

[8] S. Glimsdal, G. Pedersen, K. Atakan, C. B. Harbitz, H. P. Langtangen, and F. Løvholt. Propagation of the Dec. 26, 2004 Indian Ocean Tsunami: Effects of dispersion and source characteristics. *Int. J. Fluid Mech. Research*, 2005. To appear.

[9] S. Glimsdal, G. K. Pedersen, and H. P. Langtangen. An investigation of overlapping domain decomposition methods for one-dimensional dispersive long wave equations. *Advances in Water Resources*, 27:1111–1133, 2004.

[10] W. Gropp, E. Lusk, and A. Skjellum. *Using MPI – Portable Parallel Programming with the Message-Passing Interface*. The MIT Press, 2nd edition, 1999.

[11] M. Hamer. Solitary killers. *New Scientist*, August:18–19, 1999.

[12] H. P. Langtangen. *Computational Partial Differential Equations - Numerical Methods and Diffpack Programming*. Texts in Computational Science and Engineering. Springer, 2nd edition, 2003.

[13] G. Pedersen. Three-dimensional wave patterns generated by moving disturbances at transcritical speeds. *J. Fluid Mech.*, 196:39–63, 1988.

[14] G. Pedersen and H. P. Langtangen. Dispersive effects on tsunamis. In *Proceedings of the International Conferance on Tsunamis, Paris, France*, pages 325–340, 1999.

[15] B. F. Smith, P. E. Bjørstad, and W. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, 1996.

[16] S.-B. Woo and P. L.-F. Liu. Finite-element model for modified Boussinesq equations. I: Model development. *J. Waterway, Port, Coastal and Ocean Eng.*, 130(1):1–16, 2004.