Making Hybrid Tsunami Simulators in a Parallel Software Framework

Xing Cai[1,2] and Hans Petter Langtangen[1,2]

[1] Simula Research Laboratory, P.O. Box 134, N-1325 Lysaker, Norway,
`{xingca,hpl}@simula.no`
[2] Department of Informatics, University of Oslo, P.O. Box 1080 Blindern, N-0316 Oslo, Norway

Abstract. Extending the philosophy of additive Schwarz algorithms, we adopt a hybrid framework that allows different subdomains to use different mathematical models, spatial discretizations, local meshes, and serial codes. This hybrid software framework is implemented using object-oriented techniques, such that existing serial codes are easily reused after being equipped with the standard interface of a generic subdomain tsunami solver. The resulting hybrid parallel tsunami simulator thus has full flexibility and extensibility.

1 Motivation

Computing the wave propagation is a fundamental task in tsunami simulation. When an entire ocean is the solution domain, the computational task becomes extremely challenging, both due to the huge amount of computations needed and due to the fact that different physics apply in different regions. For example, the effect of dispersion is important for modeling wave propagation over an vast region with large water depth. In regions where water depth rapidly changes or close to the coastlines, nonlinear effects become important.

Among commonly used mathematical models, we have the following Boussinesq water wave equations:

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

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

where η and ϕ are the primary unknowns denoting, respectively, the water surface elevation and velocity potential. The water depth H is assumed to be a function of the spatial coordinates x and y. In Equations (1)-(2) the weak effect of dispersion and nonlinearity is controlled by the two dimensionless constants ϵ and α, respectively. For more mathematical and numerical details, we refer to [4, 3, 2]. Note that by choosing $\epsilon = \alpha = 0$, we recover the widely used linear shallow water equations:

$$\frac{\partial \eta}{\partial t} + \nabla \cdot (H \nabla \phi) = 0, \tag{3}$$

$$\frac{\partial \phi}{\partial t} + \eta = 0. \tag{4}$$

Regarding the numerical algorithms, the details of Equations (1)-(2) are often resolved by unstructured meshes and finite element discretization, whereas structured meshes and finite differences are commonly used for (3)-(4). Consequently, the software codes implementing these two types of numerical algorithms are different in code complexity. High-level programming languages (such as C++) are typically used to program finite elements, whereas low-level languages (such as F77 and C) are used for finite differences.

2 Parallelization by a Subdomain-Based Approach

Parallel computing is essential for simulating wave propagation over an entire ocean, because a huge number of degrees of freedom are often needed. As we have discussed above, different physics are valid in different regions, calling for a computationally resource-aware parallelization. More specifically, in regions where nonlinear and/or dispersive effects are important, existing serial software for Boussinesq equations (1)-(2) should be applied. Likewise can existing serial software for linear shallow water equations (3)-(4) be used in the remaining regions.

Such a parallelization strategy is most easily realized by using subdomains, such that the entire spatial domain Ω is decomposed into a set of overlapping subdomains $\{\Omega_s\}_{s=1}^{P}$. Mathematically, this idea of parallelization was first conceived in the additive Schwarz algorithms, see [5]. In a generic setting, where a partial differential equation (PDE) is expressed as $\mathcal{L}_{\Omega}(u) = f_{\Omega}$, the Schwarz algorithm consists of an iterative process generating $u^0, u^1, \ldots, u^k$ as a series of approximate solutions. During Schwarz iteration k, each subdomain first *independently* updates its local solution through

$$\mathcal{L}_{\Omega_s}(u_s^k) = f_{\Omega_s}^{k-1}. \tag{5}$$

Note that $f_{\Omega_s}^{k-1}$ refers to a right-hand side which is restricted within Ω_s and depends on the latest global approximation u^{k-1}. Then, the new global solution u^k is composed by "sewing together" the subdomain local solutions $u_1^k, u_2^k, \ldots, u_P^k$.

Equation (5) thus opens for the possibility of using different local solvers in different subdomains. Taking the idea of additive Schwarz one step further, we can also apply different mathematical models in different subdomains. Therefore, different serial codes may be deployed regionwise.

3 An Object-Oriented Framework

To facilitate a rapid implementation of a hybrid parallel tsunami simulator as described above, we resort to object-oriented programming techniques. As described in [1], a generic library of Schwarz algorithms for PDEs can consist of two generic components: `class SubdomainSolver` and `class Administrator`. The generic class of `SubdomainSolver` declares a generic interface of all subdomain PDE solvers. The generic interface is simply a set of virtual member functions without concrete implementation. On the other hand, the generic class of `Administrator` implements a common set of functions, such as checking global convergence and invoking inter-subdomain communication, which are independent of specific PDEs.

An object-oriented framework for hybrid parallel tsunami simulators can be illustrated by the following case study.

4 Case Study

We have two existing serial software codes, an advanced C++ finite element solver named `class Boussinesq` applicable for unstructured meshes, and a legacy F77 finite difference code applicable for uniform meshes. The purpose is to build a hybrid parallel tsunami simulator based on these two codes. Two light-weight new classes are thus programmed:

`SubdomainBQFEMSolver` and `SubdomainBQFDMSolver`

Here, class `SubdomainBQFEMSolver` uses double inheritance, as subclass of both `SubdomainSolver` and `Boussinesq`, so that it inherits the computational functionality from `Boussinesq` and is recognized by the generic `Administrator` as a subdomain solver. Similarly, class `SubdomainBQFDMSolver` is derived from `SubdomainSolver` and at the same time “wraps up” the F77 subroutines of the legacy code. Finally, another new light-weight class `HybridBQSolver` is derived as subclass of `Administrator`, so that some tsunami specific functionality can be added on top of the generic Schwarz functionality.

Using such a hybrid software framework, a parallel tsunami simulator can be built for the 2004 Indian Ocean tsunami. The entire spatial domain and the wave depth distribution are depicted in Figure 1. Moreover, the figure also shows that different types of local meshes (uniform subdomain meshes and adaptively refined subdomain meshes) and different spatial discretizations (finite differences and finite elements) can be freely deployed in different regions.

5 Concluding Remarks

We have explained a hybrid software framework for parallelizing and, at the same time, combining existing serial codes. Such a parallelization strategy is numerically inspired by the additive Schwarz algorithms, while implementationally enabled by object-oriented programming techniques.

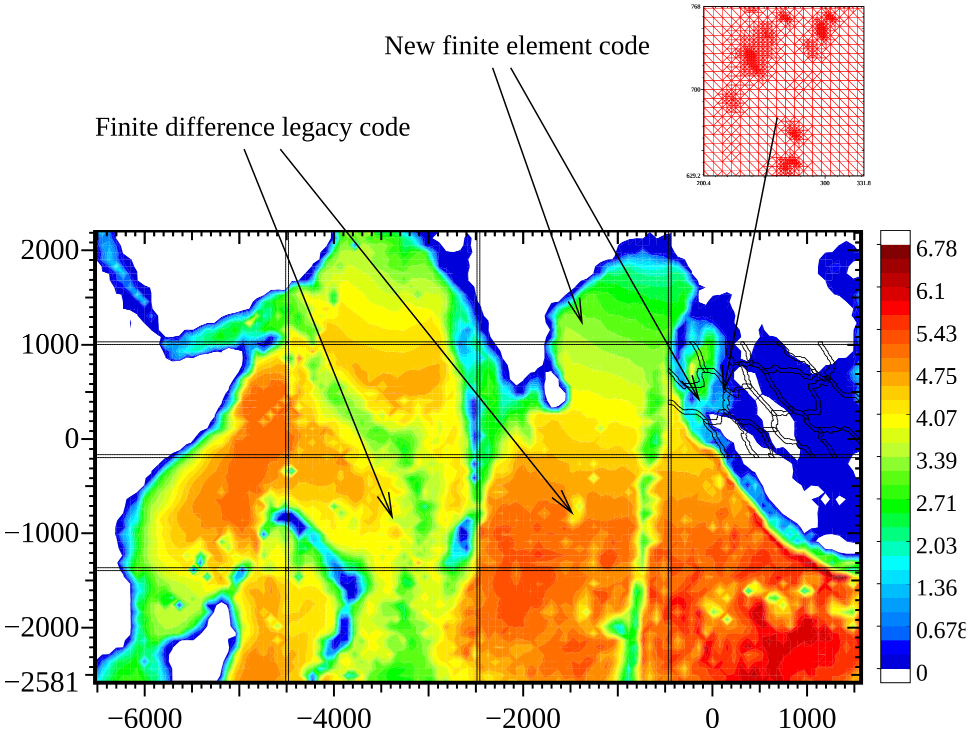

Fig. 1. An example of partitioning the Indian Ocean domain into a mixture of rectangular and complex-shaped subdomains. Finite differences or finite elements are chosen by each subdomain to carry out the spatial discretization

References

1. 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.
2. X. Cai, G. K. Pedersen, and H. P. Langtangen. A parallel multi-subdomain strategy for solving Boussinesq water wave equations. *Advances in Water Resources*, 28:215–233, 2005.
3. 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.
4. G. Pedersen and H. P. Langtangen. Dispersive effects on tsunamis. In *Proceedings of the International Conferance on Tsunamis, Paris, France*, pages 325–340, 1999.
5. B. F. Smith, P. E. Bjørstad, and W. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, 1996.