Parallel Simulation of Dual Lithology Sedimentation

Wenjie Wei, Stuart R. Clark, Xing Cai, and Are Magnus Bruaset

Center for Biomedical Computing, Simula Research Laboratory, Norway

Simulating the evolution of marine basins is challenging due to the complex interplay between erosion, deposition and mass flows of sediment. Uncertainties in the transportation modes and flow properties also require a large number of trial computations using different model settings and coefficients, see e.g. [3]. Parallel computing is thus not only indispensable for achieving high spatial and temporal resolution, but also of great importance for handling the repeated computations.

The present study is a preliminary investigation about the applicability of parallel computing to this subject, including the choice of suitable numerical strategies and their parallel implementation and execution on multicore-based clusters.

Mathematical Model

We consider here a dual-lithology (i.e., sand and silt) sedimentation scenario. If diffusion is considered to be the driving force, the following two nonlinear partial differential equations apply:

$$\frac{\partial h}{\partial t} = \frac{1}{C_s}\nabla\cdot(\alpha s\nabla h) + \frac{1}{C_m}\nabla\cdot(\beta(1-s)\nabla h), \tag{1}$$

$$\frac{\partial s}{\partial t} + s\frac{\partial h}{\partial t} = \frac{1}{C_s}\nabla\cdot(\alpha s\nabla h), \tag{2}$$

where the unknowns $s(x,y,t)$ and $h(x,y,t)$ denote, respectively, the fractional quantity of sand and the the height of bathymetry of the basin. In addition, $C_s(x,y)$ and $C_m(x,y)$ are the concentrations, $\alpha(x,y)$ and $\beta(x,y)$ are the diffusion coefficients, of sand and silt, respectively. Equations (1)-(2) are solved with given initial conditions, and the boundary conditions are normal derivatives of s and h. Fig. 1 and Fig.2 show the evolution of the basin with an area of $220 \times 120km^2$ over 4000,000 years.

Figure 1. An initial condition for h (s is initially assumed to be constant everywhere)

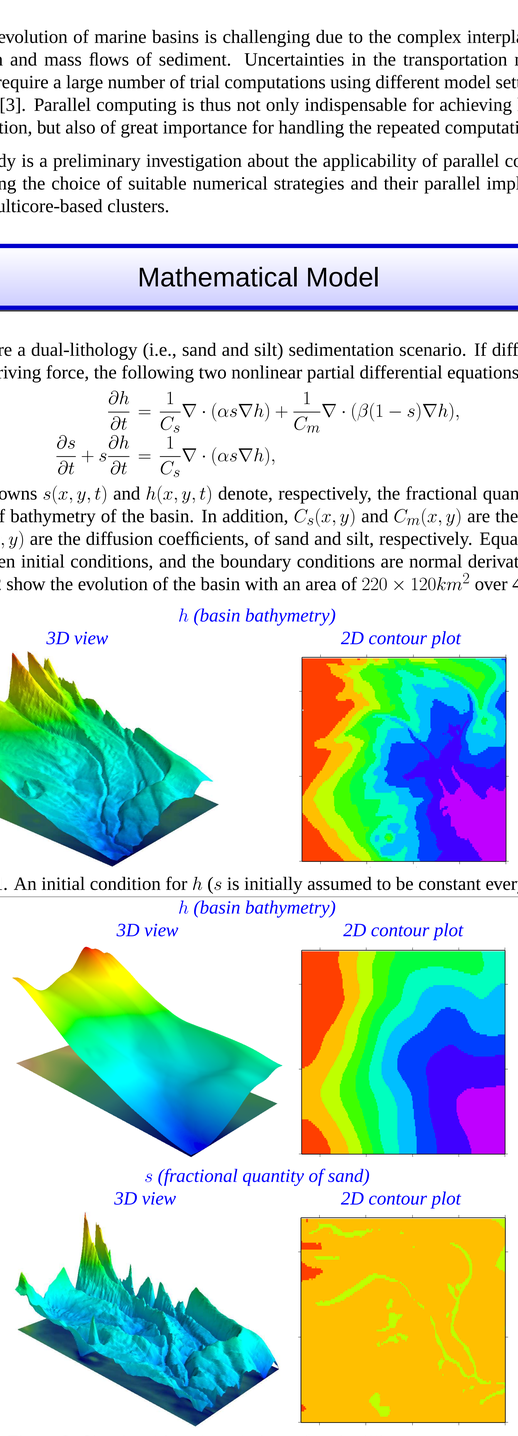

Figure 2. Simulated h and s solutions after 4 million years

Numerical Methods

Temporal discretization of (1)-(2) can be done as follows:

$$\frac{h^{\ell+1}-h^{\ell}}{\Delta t} = \frac{1}{C_s}\nabla\cdot(\alpha s^{\ell}\nabla h^{*}) + \frac{1}{C_m}\nabla\cdot(\beta(1-s^{\ell})\nabla h^{*}), \tag{3}$$

$$\frac{s^{\ell+1}-s^{\ell}}{\Delta t} + s^{*}\left(\frac{h^{\ell+1}-h^{\ell}}{\Delta t}\right) = \frac{1}{C_s}\nabla\cdot(\alpha s^{*}\nabla h^{\ell+1}). \tag{4}$$

Fully explicit scheme: $h^{*} = h^{\ell}$ and $s^{*} = s^{\ell}$. No need to solve linear systems.
Semi-implicit scheme: $h^{*} = h^{\ell+1}$ and $s^{*} = s^{\ell+1}$. Solve two separate linear systems per time step.

Due to strict stability requirement of $\Delta t = \mathcal{O}(\Delta x^2)$, the explicit scheme is not applicable to cases where high spatial resolution is desired. The semi-implicit scheme, although having to solve two linear systems per time step, will be the method of choice.

Implementation

Both the explicit scheme and the semi-implicit scheme are implemented. Parallelism arises from dividing the spatial solution domain into subdomains, while letting one MPI process to handle one subdomain. The solutions s and h are both distributed among processors, and so are the linear systems distributed when the semi-implicit method is used. We have used the Trilinos software package [2] for parallel solution of the involved linear systems.

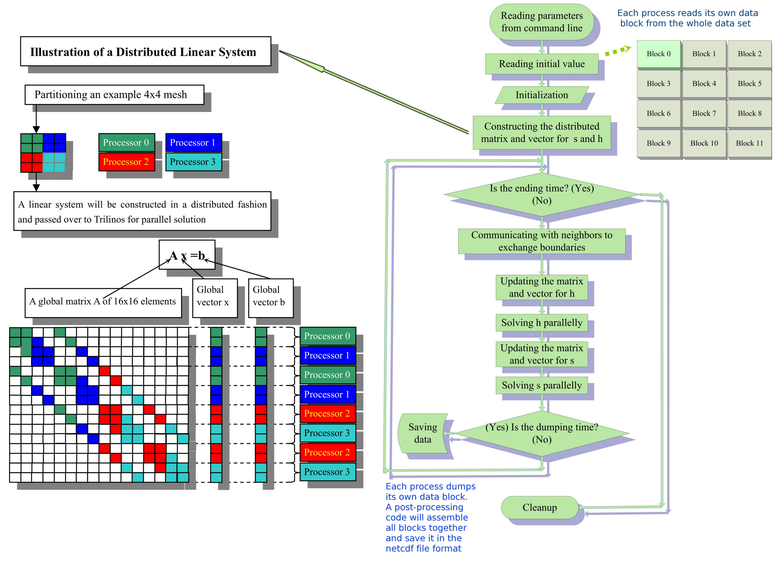

Figure 3. The flow chart and distributed data structure of the semi-implicit code

Performance

The following two tests were done on stallo.uit.no[1].

Small test case
Spatial problem size: 400×400. Time length: 40,000,000 years

Compute nodes	Total cores	Explicit Method (Speedup)	Semi-implicit Method (Speedup)
1	1	357.47	180.51
1	2	183.13 (1.95)	121.24 (1.49)
1	4	95.11 (3.76)	83.42 (2.16)
1	8	50.33 (7.10)	20.77 (8.69)
2	16	33.5 (10.67)	14.4 (12.54)

Table 1. Comparison between explicit and semi-implicit schemes. Time step size is 2,010 years for the explicit scheme, 62,5000 years for the semi-implicit scheme.

Huge test case
Spatial problem size: $10,000 \times 10,000$. Time length: 2,400,000 years

Compute nodes	Total cores	Whole running time (Speedup)
30	240	1420.19
60	480	927.17 (1.53)
90	720	591.78 (2.40)
120	960	581.69 (2.44)

Table 2. Scalability of the semi-implicit code. Time step size: 1000 years.

Discussion and Work in Progress

We found the scalability is mainly affected by inter-domain communication, which in the present code is not stable. Work for figuring out possible reasons is in progress. In the next step, the more complex model of three-lithology will be adopted. Concentrations parameters Cs and C_m will be allowed to vary over both time and space as well.

Acknowledgment. The NOTUR computing facilities have been used to conduct the numerical simulations reported in this poster.

References

[1] Stallo cluster. http://www.notur.no/hardware/stallo/.

[2] The Trilinos Project Home Page. http://http://trilinos.sandia.gov/.

[3] S.R. Clark, A.M. Bruaset, T.O. Sømme, and T.M. Løseth. Probabilistic handling of uncertainty in diffusion-based numerical models of erosion and deposition. 2009, in prep.

Emails: wenjie@simula.no (Wenjie Wei)