Progress Report: Parallelization of ADCIRC3D

Mary F. Wheeler, Clint Dawson, Srinivas Chippada, Monica Martinez and Victor Parr

CRPC-TR98765
March 1998

Center for Research on Parallel Computation
Rice University
6100 South Main Street
CRPC - MS 41
Houston, TX 77005

Submitted September 1998; Also available as CEWES MSRC Technical Report 98-11

Progress Report: Parallelization of ADCIRC3D

Mary F. Wheeler, Clint Dawson, Srinivas Chippada,
Monica Martinez and Victor Parr
Texas Institute for Computational and Applied Mathematics
University of Texas at Austin
Austin, TX 78712

INTRODUCTION.

ADCIRC is a finite element shallow water hydrodynamic flow model based on methodology developed over a period of years by Gray, Kinnmark, Kolar, Luettich, Lynch, Westerink and others; see, for example (Lynch and Gray 1979; Kinnmark 1985; Drolet and Gray 1988; Gray 1989; Luettich *et al* 1991; Westerink *et al* 1992; Kolar *et al* 1994). This simulator has been tested on numerous field applications and is one of the primary shallow water simulators in use at WES. We have recently joined forces in an attempt to mathematically analyze the model, revisit the time-stepping strategies and spatial discretization methods used, and improve its performance by parallelizing it for distributed memory parallel computers.

The ADCIRC model uses a Galerkin finite element discretization on unstructured triangular meshes, based on a reformulation of the shallow water equations into a wave equation for the elevation combined with the standard momentum equations for velocity. The wave equation formulation is used to prevent numerical oscillations which often occur in finite element discretizations of the original primitive form of the continuity equation (Walters 1983). This formulation also allows for the use of the same approximating spaces for elevation and velocity; elevation and velocities are approximated by piecewise linears in the model. A three-level time-stepping scheme is used, with nonlinear terms and coupling terms between the two equations handled explicitly. Due to the wave equation formulation, a system of linear equations must be solved at each time step. However, this system is sparse, symmetric and diagonally dominant, thus iterative techniques such as Jacobi-preconditioned conjugate gradient provide accurate solutions in a few iterations.

We have developed a parallel version of the ADCIRC model, referred to as PADCIRC. This simulator uses domain (data) decomposition to achieve parallelism. We also employ a Hilbert space-filling curve technique for partitioning the finite element mesh among the processors. These details and performance data are given below.

MATHEMATICAL AND NUMERICAL FORMULATION.

Let $\xi(\boldsymbol{x}, t)$ denote the free surface elevation over a reference plane and $h_b(\boldsymbol{x})$ the bathymetric depth under that reference plane, with $H = \xi + h_b$ the total water column. Let $\boldsymbol{u} = (U, V)(\boldsymbol{x}, t)$ denote the depth-averaged horizontal velocities.

Letting $\boldsymbol{q} = \boldsymbol{u}H$, the 2-dimensional shallow water equations, in operator form (Kinnmark 1985; Luettich *et al* 1991), are the primitive continuity equation

$$\mathrm{L}(\xi, \boldsymbol{u}; h_b) \;\equiv\; \frac{\partial \xi}{\partial t} + \nabla\cdot\boldsymbol{q} = 0, \tag{1}$$

and the primitive non-conservative momentum equations (NCME),

$$\begin{aligned}\mathbf{M}(\xi, \boldsymbol{u}; \Phi) \;\equiv\; & \frac{\partial}{\partial t}\boldsymbol{u} + (\boldsymbol{u}\cdot\nabla)\,\boldsymbol{u} + \tau_{bf}\boldsymbol{u} \\ & + \boldsymbol{k} \times f_c\boldsymbol{u} + g\nabla\xi - \frac{1}{H}E_h\Delta\boldsymbol{q} \\ & - \frac{1}{H}\boldsymbol{\tau}_{ws} + H\boldsymbol{\mathcal{F}} = 0,\end{aligned} \tag{2}$$

where $\Phi = (h_b, \tau_{bf}, f_c, g, E_h, \boldsymbol{\tau}_{ws}, p_a, \eta)$. In particular, $\tau_{bf}(\xi, \boldsymbol{u})$ is a bottom friction function, $\boldsymbol{k}$ is a unit vector in the vertical direction, f_c is the Coriolis parameter, g is acceleration due to gravity, E_h is the horizontal eddy diffusion/dispersion (constant) coefficient, $\boldsymbol{\tau}_{ws}$ is the applied free surface wind stress relative to the reference density of water, and $\boldsymbol{\mathcal{F}} = (\nabla p_a - g\nabla\eta)$, where $p_a(\boldsymbol{x}, t)$ is the atmospheric pressure at the free surface relative to the reference density of water, and $\eta(\boldsymbol{x}, t)$ is the Newtonian equilibrium tide potential relative to the effective Earth elasticity factor.

The ADCIRC model replaces the primitive continuity equation with the Generalized Wave Continuity Equation (GWCE), which is obtained by differentiating (1) in time, differentiating the conservative form of (2) in space, multiplying (1) by a parameter $\tau_\circ$ and adding the result, thus obtaining

$$\begin{aligned}\frac{\partial^2\xi}{\partial t^2} + \tau_\circ\frac{\partial \xi}{\partial t} - \nabla\cdot\Bigg[\nabla\cdot\left(\frac{1}{H}\boldsymbol{q}^2\right) \\ -(\tau_\circ - \tau_{bf})\boldsymbol{q} + (\boldsymbol{k} \times f_c\boldsymbol{q}) + Hg\nabla\xi \\ + E_h\nabla\frac{\partial\xi}{\partial t} - \boldsymbol{\tau}_{ws} + H\boldsymbol{\mathcal{F}}\Bigg] = 0.\end{aligned} \tag{3}$$

Let Ω denote the computational domain, and let $0 \le t \le T$ denote a time interval. We supplement the GWCE-NCME with initial conditions

$$\begin{aligned}&\xi(\boldsymbol{x}, 0) = \xi_0(\boldsymbol{x}),\; \tfrac{\partial\xi}{\partial t}(\boldsymbol{x}, 0) = -\nabla\cdot\boldsymbol{v}(\boldsymbol{x}, 0), \\ &U(\boldsymbol{x}, 0) = U_0(\boldsymbol{x}),\; V(\boldsymbol{x}, 0) = V_0(\boldsymbol{x}),\end{aligned}$$

and, for simplicity, with homogeneous Dirichlet boundary conditions on U, V and ξ. The ADCIRC model accounts for general land and sea boundary conditions.

We define a continuous-time Galerkin approximations to ξ, $\boldsymbol{q}$ to be the mappings $\xi_h(\boldsymbol{x}, t) \in \mathcal{S}^h$, $\mathbf{U}_h(\boldsymbol{x}, t) \in \boldsymbol{\mathcal{S}}^h$ for each $t > 0$ satisfying

$$\begin{aligned}
&\left(\frac{\partial^2 \xi_h}{\partial t^2}, v\right) + \tau_\circ \left(\frac{\partial \xi_h}{\partial t}, v\right) + \left(\nabla \cdot \left\{\frac{1}{H_h}\mathbf{U}_h^2\right\}, \nabla v\right) \\
&\quad + ((\tau_{bf} - \tau_\circ)\mathbf{U}_h, \nabla v) + (\boldsymbol{k} \times f_c \mathbf{U}_h, \nabla v) \\
&\quad + (H_h g \nabla \xi_h, \nabla v) + E_h \left(\nabla \frac{\partial \xi_h}{\partial t}, \nabla v\right) \\
&\quad - (\boldsymbol{\tau}_{ws}, \nabla v) + (H\boldsymbol{\mathcal{F}}, \nabla v) \\
&\quad = 0, \qquad \forall v \in \mathcal{S}^h(\Omega),
\end{aligned} \tag{4}$$

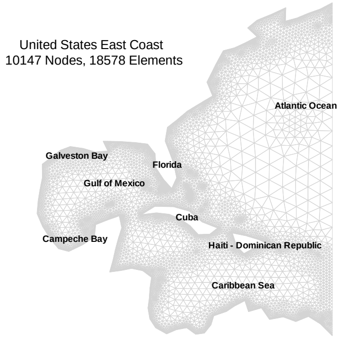
$$\begin{aligned}
&\left(\frac{\partial \mathbf{U}_h}{\partial t}, \boldsymbol{w}\right) + \left(\nabla \cdot \left\{\frac{1}{H_h}\mathcal{Q}^2\right\}, \boldsymbol{w}\right) + (\tau_{bf}\mathbf{U}_h, \boldsymbol{w}) \\
&\quad + (\boldsymbol{k} \times f_c \mathbf{U}_h, \boldsymbol{w}) + (H g \nabla \xi_h, \boldsymbol{w}) \\
&\quad + E_h (\nabla \mathbf{U}_h, \nabla \boldsymbol{w}) - (\boldsymbol{\tau}_{ws}, \boldsymbol{w}) \\
&\quad + (H_h \boldsymbol{\mathcal{F}}, \nabla v) = 0, \qquad \forall \boldsymbol{w} \in \boldsymbol{\mathcal{S}}^h(\Omega).
\end{aligned} \tag{5}$$

Equations (4)-(5) can be discretized in time in several ways. For (4), ADCIRC employs a three-level scheme centered at time t^k. Linear terms are handled implicitly, nonlinear terms are handled explicitly. The momentum equations are centered at $t^{k+1/2} = (t^{k+1} + t^k)/2$ except for nonlinear terms which are evaluated at t^k. Given a solution at t^k, (4) is solved updating ξ_h to time t^{k+1}. The velocities are then updated. When diffusion is included, linear systems of equations must be solved in both (4) and (5). However both of these matrices are time-independent, symmetric and diagonally dominant. When diffusion is neglected ($E_h = 0$), a linear system of equations must be solved only for determining ξ_h^{k+1}. A diagonally-preconditioned conjugate gradient iterative method is used for solving these systems.

PARALLEL IMPLEMENTATION AND RESULTS.

We have developed PADCIRC Version 1.0, a parallelization of version 24 of ADCIRC, which solves the two-dimensional equations described above. This version of PADCIRC has been executed on an Intel Paragon, IBM SP-2 and networks of workstations. The code was parallelized using a data decomposition strategy. A preprocessor code was developed which takes the global input file and splits it into separate input files to be read by each processor in the multiprocessor environment. The decomposition is performed by nodes; i.e., each processor is responsible for the unknowns at certain vertices in the triangular mesh. For nodes on a boundary between subdomains, the elements connected to these nodes are shared by the processors. Thus some overlap in computing is performed; however the changes to the original code were few. Only a few message-passing calls were added. Moreover, no global arrays are required. Each processor only needs enough memory to

Figure 1: United States east coast grid: 18578 elements, 10147 nodes



store arrays corresponding to the mesh on that processor. The values of the unknowns at neighboring nodes are passed using MPI message-passing routines.

In our first attempt at parallelization, a simplistic data decomposition was used, based on the ordering of unknowns provided by the mesh generation package. Given 1000 nodes and 10 processors, the first 100 nodes were sent to processor 1, nodes 101-200 sent to processor 2, and so forth. Below we refer to this approach as the node-ordering (NO) decomposition. This approach resulted in some subdomains having very high surface-to-volume ratios for most of the cases we have considered. The performance of this approach for tidal simulation of Galveston Bay and the eastern coast of the United States is given in (Chippada *et al* 1996).

In order to improve the parallel efficiency of our simulator, we have employed a Hilbert space-filling curve (Sagan 1994) decomposition strategy. In this approach, a space-filling curve is drawn through the mesh which maps nodes in the mesh to the interval $[0, 1]$ on the real line. The curve is constructed so that it passes through each node only once and enforces nearest neighbor groupings. Once the curve is constructed, the domain decomposition is performed by evenly partitioning the interval $[0, 1]$. Below we refer to this approach as the space-filling curve (SFC) decomposition. More details on this approach to domain decomposition for finite element computations can be found in (Patra and Oden 1995).

One application of interest is modeling tides in the western North Atlantic Ocean (Westerink *et al* 1994). This domain includes the eastern U.S. seaboard and the Gulf of Mexico. A typical finite element mesh is given in Figure (1).

In Figure (2), we show a mesh partitioning based on the node-ordering decomposition as discussed above, assuming 4 processors. Notice the long, thin regions which result. These subdomains have very poor surface-to-volume ratios, ranging from 6.0 to 9.3%. Here surface-to-volume ratio is defined as the number of nodes on the processor boundary of the

Figure 2: NO decomposition: 18578 elements, 10147 nodes

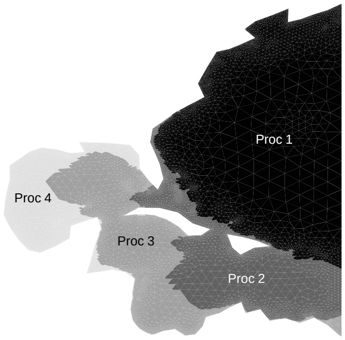
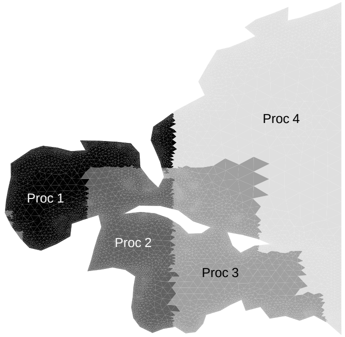


Figure 3: SFC decomposition: 18578 elements, 10147 nodes



subdomain divided by the total number of nodes in the subdomain.

In Figure (3), we show a mesh partitioning based on the space-filling curve. In this case the subdomains are much better proportioned. This is reflected in the surface-to-volume ratios, which range from 1.0 to 3.2 % in this case.

We tested the parallel efficiency of the code on an Intel Paragon for both types of decomposition above. The MPI message passing library was employed. In this simulation we took 30 second time steps and simulated a period of 1 day. We neglected diffusive effects. The linear solver for the elevation took roughly 3-5 iterations per time step to converge to a relative tolerance of 10^{-6}. The CPU times for 2, 4, 8, 16, and 32 processors are given in the table below. Notice that the differences in CPU times between the two approaches is particularly noticeable as the number of processors is increased. The speed-up factor between 2 and 32 processors is 7.57 for the node-ordering decomposition and 10.41 for the space-filling curve decomposition.

No. Processors	NO	SFC
2	7355.5	7283.9
4	3970.6	3682.9
8	2188.1	1891.4
16	1370.5	1159.1
32	971.7	699.7

We next present results for the same computational domain but with a refined grid. This grid consists of 31435 nodes and 58369 elements. CPU times for both partitioning algorithms are given in the following table.

No. Processors	NO	SFC
2	31424.0	30795.5
4	16446.7	15472.1
8	8817.1	7792.3
16	4934.5	4003.1
32	3032.4	2110.9

In this case, the speed-up factor between 2 and 32 processors is 10.4 for the node-ordering decomposition and 14.6 for the space-filling curve decomposition. The latter is close to the "optimal" factor of 16.

Finally, in Figure (4), we plot the speed-up for the same problem on a 16 processor IBM SP2, using the space-filling curve decomposition. Note that the speed-up is again close to optimal.

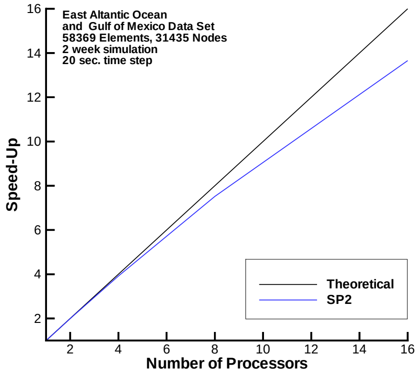
CURRENT AND FUTURE WORK

Above we have described the parallelization of Version 24 of ADCIRC. Recently, the developers of ADCIRC have added several new features to the code, including wetting and drying, more general land boundary conditions, and the capability to model three-dimensional variations in velocity. This version (Version 33.04) of the code has its own preprocessor which allows the user to incorporate and/or remove certain features in the program, depending on the problem, the machine to be used, etc. Because of the additional flexibility of the code, new input files have also been added.

The parallelization of Version 33.04 is proceeding similarly to the strategy described above, in particular:

- A preprocessor, ADCPREP, is being developed which partitions the domain using a space-filling curve, splits the global input files into processor-dependent input files,

Figure 4: Speedup vs. number of processors, IBM SP2



and sets up the communication table for message passing. A preliminary version of ADCPREP is working and has been tested on data sets provided by Joannes Westerink and Rick Luettich. Some additional work is needed in the preprocessor to handle certain types of boundary conditions which are new to the code.

- Message passing calls are being placed in the code and also a "parallel" option is being added to the ADCIRC preprocessor. We expect to have a working version of the parallel code before the end of this fiscal year.

- We are coordinating our efforts with the ADCIRC development team so that the next version of ADCIRC will include the parallel option; all future versions of ADCIRC will then be easily parallelizable.

REFERENCES

Chippada, S., C. N. Dawson, M. Martinez and M. F. Wheeler, 1995. "Finite element approximations for the system of shallow water equations, Part I: Continuous time a priori error estimates," to appear in SIAM J. Numer. Anal.

Chippada, S., C. N. Dawson, M. Martinez and M. F. Wheeler, 1996. "Parallel computing for finite element models of surface water flow," *Computational Methods in Water Resources XI*, Computational Mechanics Publications, Southampton, U.K.: 63-70.

Drolet, J. and W. G. Gray, 1988. "On the well-posedness of some wave formulations of the shallow water equations," *Advances in Water Resources 11*: 84–91.

Gray, W. G., 1989. "A finite element study of tidal flow data for the North Sea and English Channel," *Advances in Water Resources 12*: 143–154.

Kinnmark, I.P.E., 1985. *The Shallow Water Wave Equations: Formulation, Analysis and Applications*, vol. 15 of Lecture Notes in Engineering, Springer-Verlag, New York.

Kolar, R. L., J. J. Westerink, M. E. Canterkin and C. A. Blain, 1994. "Aspects of nonlinear simulations using shallow-water models based on the wave continuity equation," *Computers Fluids 23*; 523–538.

Luettich, R. A., J. J. Westerink, and N. W. Scheffner, 1991."*ADCIRC*: An advanced three-dimensional circulation model for shelves, coasts, and estuaries," Tech. Report 1, Department of the Army, U.S. Army Corps of Engineers, Washington, D.C. 20314-1000, December.

Lynch, D. R. and W. G. Gray, 1979. "A wave equation model for finite element tidal computations," *Computer and Fluids 7*: 207–228.

Patra, A. and J. T. Oden, 1995. "Problem decomposition for adaptive *hp* finite element methods," *Computing Systems in Engineering 6*: 97–109.

Sagan, H, 1994. *Space-Filling Curves.* Springer-Verlag, New York.

Walters, R. A., 1983. "Numerically induced oscillations in finite element approximations to the shallow water equations," *International Journal for Numerical Methods in Fluids 3*: 591–604.

Westerink, J. J, R. A. Luettich and J. C. Muccino, 1994. "Modeling tides in the western North Atlantic using unstructured graded grids," *Tellus 46A*: 178-199.