Faster than Real-Time Coastal Wave Visualization with a Phase-Resolving Boussinesq-type Model

Patrick Joseph Lynett
Sonny Astani Department of Civil and Environmental Engineering
University of Southern California
Los Angeles, CA
phone: (213) 740-3133 fax: (213) 744-1426 email: plynett@usc.edu

Award Number: N00014-14-1-0307

LONG-TERM GOALS

This project is driven by the desire to develop a phase-resolving, nonlinear and dispersive wave model for immediate and interactive prediction of coastal wave heights, breaker locations, and wave-induced currents - that can run on laptop and tablet hardware. Such a tool would allow for rapid simulation of complex coastal conditions, either in the office or in the field, and might further our ability to perform realtime coastal wave predictions with forward-looking windows on the order of minutes, if combined, for example, with a coastal radar system.

OBJECTIVES

The main objectives of this effort include:

- Creation of a "fast-Boussinesq" code for use on multi-core CPU's using modern Finite-Volume techniques
- Porting of the CPU code for use on GPU's, where early work has indicated model speeds in exceed of 100 times faster than real time
- Inter-comparison of the fast CPU model, the GPU model, and previously developed high-order Boussinesq type models using data from the New River Inlet field study.

The anticipated outcome of this proposed effort will be the development of two, fast-Boussinesq solvers, one that uses CPU's and a another that uses GPU's. The models will be structured to use standard input and output formats, such that they might be incorporated into established user interfaces (e.g. Delft Dashboard). Through this interface, the user can chose which version of the code to use depending on their local hardware.

APPROACH

Boussinesq-type models found in many applications are high-order variations requiring similarly high-order numerical schemes. While these approaches provide the research-level accuracy needed for certain studies, they do not provide a means for rapid computation; real-time simulations of a modest coastal region can only currently be achieved with parallel processing on dozens to hundreds of cores.

We will utilize a modified form of one of the earlier versions of the Boussinesq-type model (circa early 1990's), but combined with recent numerical solution techniques to maximize computational solution speed.

The model to be solved will be the weakly nonlinear Boussinesq-type model, without the Nwogu-type modifications. While disregarding the Nwogu-type modifications yields a model with decreased dispersion accuracy for intermediate water waves, it also yields a model without third-order spatial derivatives. It is these derivatives that induce huge computational cost. The numerical solution scheme – spatial derivatives and time integration – must be fourth-order accurate to prevent numerical errors that appear similar to these "real" third-order derivatives.

The to-be-employed Boussinesq equations are re-cast in conservative form as shown below:

$$\frac{\partial H}{\partial t} + \nabla \cdot H\vec{U} = 0$$

$$\frac{\partial H\vec{U}}{\partial t} + \nabla\left(H\vec{U} \cdot \vec{U}\right) + gH\nabla\eta + H\left[\frac{d^2}{2}\frac{\partial\nabla\left(\nabla \cdot \vec{U}\right)}{\partial t} + \frac{d}{6}\frac{\partial\nabla\left(\nabla \cdot \left(d\vec{U}\right)\right)}{\partial t}\right] = 0$$

Where H is the total water depth = η +d, η is the free surface elevation, d is the still water depth, U is the depth-averaged velocity vector, and g is gravity. To solve this equation set, we will use modern Finite-Volume (FV) techniques, such as those used in the PI's previous modeling efforts (e.g. Kim et al, 2009). The advantage of using the FV method for the work proposed here is that they can be extremely robust when implementing certain flux reconstruction and limiters. Here, "robust" implies that numerical dissipation keeps the solution stable and smooth. In addition, FV limiters can be used as a proxy for coastal wave breaking. In conventional Boussinesq-type modeling, breaking is approximated with ad-hoc dissipation submodels (e.g. Kennedy et al, 2000). While these submodels are shown to be accurate, they also add a substantial computational and memory cost to the numerical solution. While modeling wave breaking through FV limiters (i.e. numerical dissipation) is not preferred from a physical accuracy perspective, such an approach can be tuned for reasonable breaking prediction for a defined range of grid sizes and wave periods in shallow water.

Initially the model will be coded for use on multi-core CPU's, as nearly all laptops and desktops now contain. This code is considered the "general-use" code, and can be easily executed on all platforms. Efficiency and speed of this code will be tested, using a domain on the scale of New River Inlet. Comparisons between this proposed model, Lynett's high-order models, and the RIVET I field data will provide benchmarks on model accuracy and applicability. After this effort, the fast-Boussinesq code will be ported to the Graphics Processing Unit (GPU). Lynett's research group has already developed and tested a Nonlinear Shallow Water (NLSW) wave equation model on the GPU. An example of the NLSW GPU solution is online here: http://www.youtube.com/watch?v=nM21JeUHzys&feature=youtu.be. The shallow water model runs at ~250 times real time on a high-end GPU card for a domain on the scale of New River Inlet (~10 km^2).

WORK COMPLETED

A hybrid finite volume – finite difference numerical scheme is used to solve Boussinesq equations. The advective part of the equations are solved using finite-volume method (FVM), while dispersive and source terms are discretized using the finite difference formulation. This hybrid discretization enables the software to benefit from robustness of FVM, shock-capturing features, and flux limiters, while retaining the higher accuracy of the model. The model is also positivity preserving and therefore there is no need to keep track of dry/wet cells. Time integration is done with the third order Adams–Bashforth scheme as prediction step and the fourth-order Adams–Moulton algorithm as an optional correction step. The model is validated for well-known test cases such as solitary wave runup on a slope, and the results are promising.

The high performance computing capabilities of the software comes from GPU. The simulations are done on GPU using DirectX API. In order to use this API, the computational problem needs to be reformulated in terms of graphics primitives and according to the underlying graphical concepts. To solve the equations on the GPU, the simulation grid is assumed as a graphical texture and copied on the GPU. Then, each step of the numerical scheme (e.g. reconstruction, flux calculation, etc.) is done by passing a shader over the texture. Shaders are small chunks of codes that perform computations on each element of the texture (texel). This means that hundreds of processing units of GPU are simultaneously working on grid cells.

The most challenging part of the work is the parallelization of the tridiagonal matrix solution included in the numerical scheme. The classic algorithm to solve such a system is the Thomas algorithm consisting of a forward elimination and backward substitution. However the algorithm is inherently serial. Employing such an algorithm will generally need to copy data from GPU to the main memory, run the serial solver and copy the results back on the GPU. Such a process will significantly increase the running time of the software. Here, parallelization is accomplished using the cyclic reduction (CR) algorithm. CR, also consists of two phases, forward reduction and backward substitution. In the forward reduction phase, the system is successively reduced to a smaller system with half the number of unknowns, until a system of 2 unknowns is reached which can be solved trivially. In the backward substitution phase the other half of the unknowns are determined step by step using the previously solved values. CR is computationally twice as expensive as the Thomas algorithm, but it can be implemented in parallel. A snapshot of the GPU-based visualization model is shown in Figure 1.

As a related project to the development of the interactive rapid Boussinesq tool, we have created an augmented reality sandbox. Details of the sandbox can be found here: http://idav.ucdavis.edu/~okreylos/ResDev/SARndbox/, and a photograph of the sandbox is given in Figure 2. The current version of the sandbox uses a crude shallow water solver. We expect to replace this shallow water solver with our Boussinesq solver, such that wind waves and coastal problems may also be examined within the sandbox.

RESULTS

The model runs ~50 faster than real-time for the modest coastal region with $\sim 10^4$ grid points, on an ordinary laptop. The model provides interactivity such that users can change numerical and physical parameters on the fly. They can also raise/lower the water surface or terrain without interrupting the simulation. The users can also opt out visualization to achieve an even higher simulation speed.

IMPACT APPLICATIONS

The GPU-Boussinesq model will be on the order of ~100 faster than real time on workstation hardware, and 10-50 faster than real time with low-end mobile hardware. A primary advantage of the GPU solution process is that one can perform the visualization as part of the model solution. If visualization is the primary objective, and the user platform (tablets or computers) are expected to have a GPU as nearly all do currently, then the GPU-FV solution is a promising option.

RELATED PROJECTS

This modeling ability to be developed in this project will be useful for simulating the data recorded during the RIVET I and II field surveys, or any project that requires a rapid prediction of coastal wave heights.

REFERENCES

Kennedy, A. B., Chen, Q., Kirby, J.T. and Dalrymple, R. A. 2000. "Boussinesq modeling of wave transformation, breaking, and runup. 1: 1D". *J. Waterway, Port, Coastal, and Ocean Engrg*., ASCE, 126, 39-47.

Kim, D.-H., Patrick J. Lynett, Scott A. Socolofsky, 2009. A depth-integrated model for weakly dispersive, turbulent, and rotational fluid flows, Ocean Modelling, Volume 27, Issues 3-4, Pages 198-214

Kim, D.-H. and Lynett, P., "Turbulent Mixing and Scalar Transport in Shallow and Wavy Flows," Physics of Fluids, v. 23 (1), doi:10.1063/1.3531716, 2011.

Lynett, P. (2006). "Wave Breaking Velocity Effects in Depth-Integrated Models." Coastal Engineering, v. 53, p. 325-333.

Lynett, P., Wu, T.-R., and Liu, P. L.-F., 2002. "Modeling Wave Runup with Depth-Integrated Equations," *Coastal Engineering*, v. 46(2), p. 89-107.

Nwogu, O. 1993 Alternative form of Boussinesq equations for nearshore wave propagation. *Journal of Waterway, Port, Coastal and Ocean Engng.* 119(6), 618-638.

Peregrine, H., 1697. Long waves on a beach, *J. Fluid Mech.* 27 (4), pp. 815–827.

Sitanggang, K. and Lynett, P. 2005. "Parallel Implementation of a Boussinesq Wave Model," *International Journal for Numerical Methods in Fluids*, doi: 10.1002/fld.985.

Son, S., Lynett, P., and Kim, D.-H., "Nested and Multi-Physics Modeling of Tsunami Evolution from Generation to Inundation," Ocean Modelling, v. 38 (1-2), p. 96-113, doi: 10.1016/j.ocemod.2011.02.007, 2011.

Son, S. and Lynett, P., "Interaction of Shallow Water Waves with Weakly Sheared Currents of Arbitrary Profile," in preparation for Coastal Engineering, 2013.

PUBLICATIONS

None yet to date

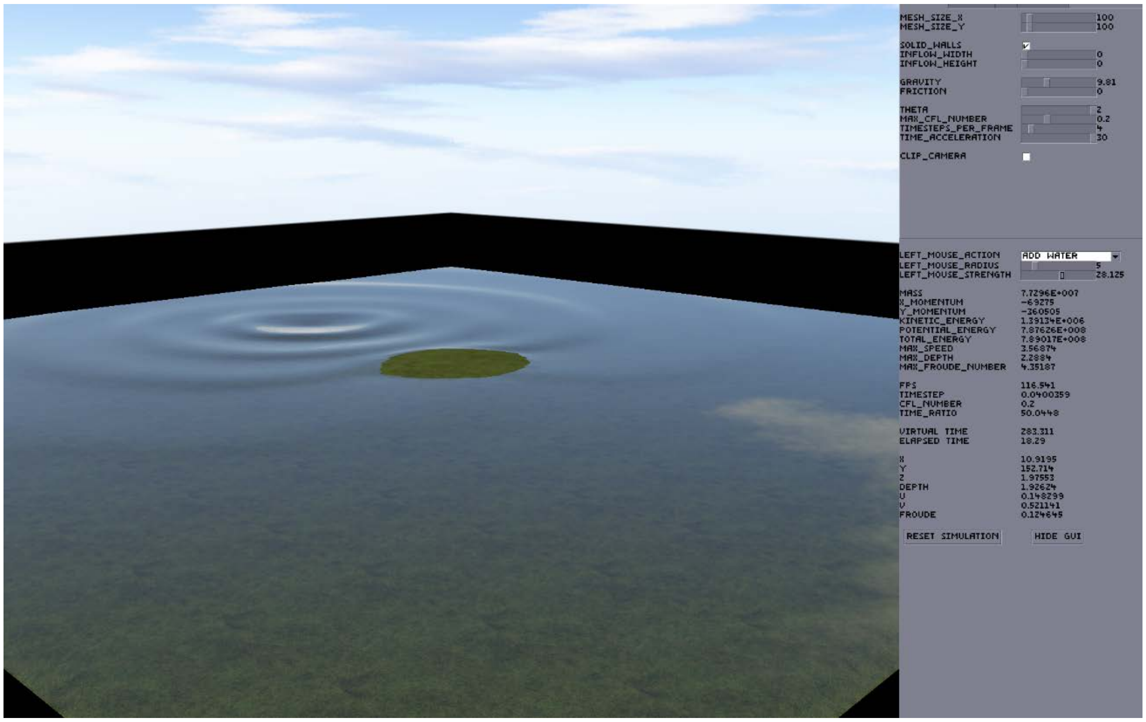

Figure 1. Example output of a wave field from the GPU-based Boussinesq model

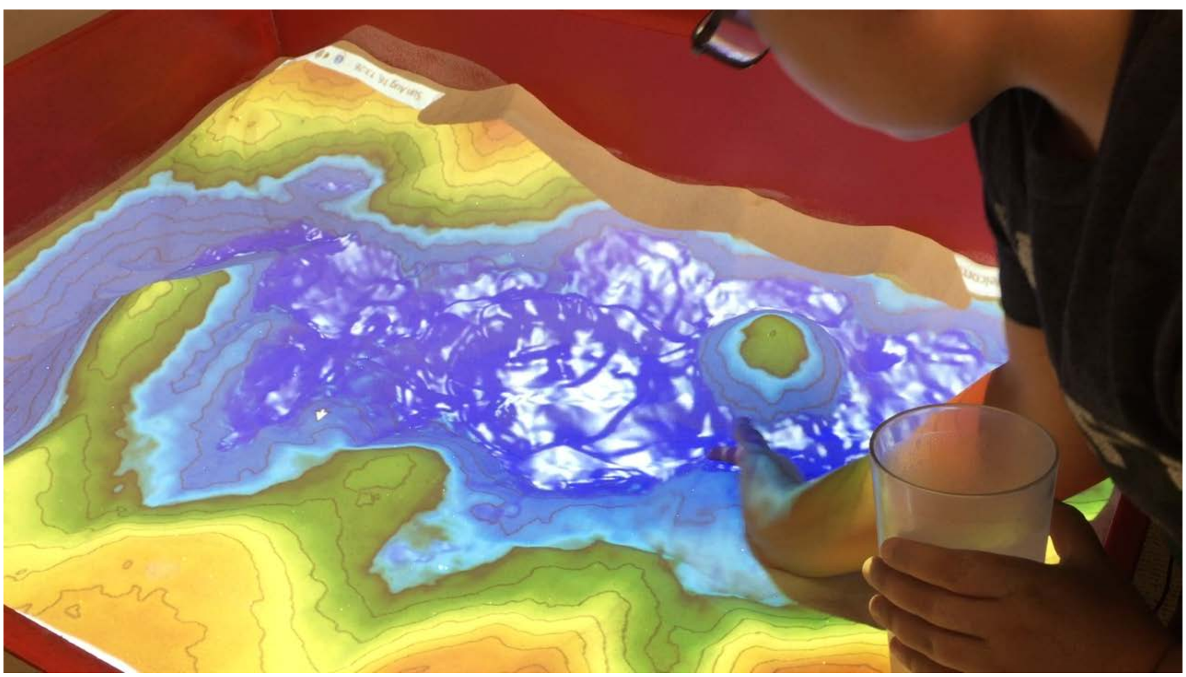

Figure 2. Elementary student interacting with the USC Augmented Reality Sandbox.
The "water" is seen as the highly reflective blue surface, and the projected contours follow the elevation of the sand.