

Amouzgar R, Liang Q, Clarke PJ, Yasuda T, Mase H. Computationally efficient tsunami modelling on Graphics Processing Units (GPUs). *International Journal of Offshore and Polar Engineering* 2016, 26(2), 154-160.

Copyright:

© International Society of Offshore and Polar Engineers 2016

Link to article:

<http://www.isopec.org/publications/journals/ijope-26-2/abst-26-2-p154-ak10-Amouzgar.pdf>

Date deposited:

27/09/2016



This work is licensed under a [Creative Commons Attribution-NonCommercial 3.0 Unported License](https://creativecommons.org/licenses/by-nc/3.0/)

Computationally Efficient Tsunami Modeling on Graphics Processing Units (GPUs)

Reza Amouzgar, Qihua Liang* and Peter J. Clarke
School of Civil Engineering and Geosciences, Newcastle University
Newcastle upon Tyne, United Kingdom

Tomohiro Yasuda and Hajime Mase
Kyoto University, Kyoto, Japan

Tsunamis generated by earthquakes commonly propagate as long waves in the deep ocean and develop into sharp-fronted surges moving rapidly toward the coast in shallow water, which may be effectively simulated by hydrodynamic models solving the nonlinear shallow water equations (SWEs). However, most of the existing tsunami models suffer from long simulation time for large-scale real-world applications. In this work, a graphics processing unit (GPU)-accelerated finite volume shock-capturing hydrodynamic model is presented for computationally efficient tsunami simulations. The improved performance of the GPU-accelerated tsunami model is demonstrated through a laboratory benchmark test and a field-scale simulation.

INTRODUCTION

Tsunamis are among the most dangerous natural disasters and are reported to potentially pose medium to high risk to most coastlines worldwide. Numerical modeling of tsunami propagation and run-up is essential for evacuation planning, risk assessment, and sometimes real-time forecasting. Numerical models based on the shallow water equations (SWEs) are commonly accepted for simulation of tsunami wave propagation from deep ocean to near shore including inundation.

To solve the SWEs for tsunami modeling, different approaches have been used, including the finite difference method, finite volume method, finite element method, and smoothed particle hydrodynamics (SPH). Most of the conventional tsunami models are based on finite difference leapfrog schemes, e.g., TUNAMI by Goto et al. (1997), MOST by Titov and Synolakis (1995), and COMCOT by Wang and Liu (2006). In recent years, finite volume Godunov-type schemes have also been implemented to solve the SWEs for tsunami modeling and have gradually gained popularity (Popinet, 2011; Leveque et al., 2011). These models boast automatic shock-capturing capability, superior conservation property, and flexibility for implementation on different types of computational grids for better boundary fitting. Because of these advantages, a second-order finite volume Godunov-type hydrodynamic model incorporated with an HLLC Riemann solver for interface flux calculation is used in this work for tsunami simulations. However, these sophisticated fully 2-D hydrodynamic models are normally computationally demanding for high-resolution simulations over large domains, restricting their wider applications.

Different approaches have been explored to improve the computational efficiency for the hydrodynamic tsunami models to enable

multiscale tsunami simulations. For example, Leveque et al. (2011) employed adaptive block meshes to accelerate their finite volume Godunov-type tsunami model. Popinet (2011) reported a finite volume tsunami model on dynamically adaptive quadtree grids. Liang et al. (2015) presented another finite volume shock-capturing tsunami model developed on a simplified adaptive grid system that is free of data structure. Depending on applications, these adaptive mesh refinement (AMR) techniques may speed up a model several times (Liang et al., 2015) but have difficulty in ensuring full conservation of both mass and surface gradient during grid adaptation.

Adopting a different approach, Pophet et al. (2011) explored the use of multicore parallel computing to improve computational efficiency for their tsunami model solving the Boussinesq equations. A similar parallel algorithm was also used by Delis and Mathioudakis (2009) to develop their shock-capturing tsunami model, which solves the SWEs.

Accessible even on general desktop PCs, a more promising high-performance computing technique involving the use of graphics processing units (GPUs) has started to gain rapid popularity in the last few years. GPUs have been commonly used in the game industry but are only recently available for scientific computing (Brodtkorb, 2010). There are hundreds of processing elements on a single GPU to provide powerful parallel computing capability, in contrast to a central processing unit (CPU). The benefit of using GPUs to provide high-performance computing is evident. In less than one decade, numerous GPU-accelerated models have been developed and used in many areas of scientific computing, e.g., computational fluid dynamics (CFD), magnetohydrodynamics, and gas dynamics (Wang et al., 2010; Kuo et al., 2011; Rossinelli et al., 2011; Schive et al., 2012).

In computational hydraulics that focuses on SWE models, Brodtkorb (2010) implemented Kurganov–Levy and Kurganov–Petrova numerical schemes to solve the SWEs on a GPU and test the Compute Unified Device Architecture (CUDA)-based heterogeneous architectures for improved computational performance. More recently, Smith and Liang (2013) presented a second-order accurate finite volume Godunov-type SWE model on GPUs. Because of the use of the OpenCL programming framework, their model can be run on any modern GPUs and CPUs and therefore

*ISOPE Member.

Received March 8, 2015; updated and further revised manuscript received by the editors December 7, 2015. The original version (prior to the final updated and revised manuscript) was presented at the Twenty-fourth International Ocean and Polar Engineering Conference (ISOPE-2014), Busan, Korea, June 15–20, 2014.

KEY WORDS: Tsunami modeling, shallow water equations, finite volume Godunov-type scheme, graphics processing units (GPUs), heterogeneous computing, CUDA.

offers greater flexibility in model applications. Both of these GPU SWE models were originally developed and tested for pluvial or surface flood modeling; however, their capability needs to be further verified for tsunami modeling, which is numerically more challenging and requires accurate representation of wave propagation, dispersion, and overland surge.

In this work, a GPU-accelerated second-order accurate hydrodynamic model is presented for tsunami simulations, which is an extension of the first-order accurate model previously reported by the authors (Amouzgar et al., 2014) and better suited for practical tsunami simulations. The model solves the 2-D SWEs using a finite volume Godunov-type scheme incorporated with an HLLC approximate Riemann solver. Effective numerical techniques are implemented to ensure a well-balanced solution of the lake-at-rest problem and to accurately track moving wet/dry shorelines (Liang, 2010). Finally, the model is implemented on GPUs using the NVIDIA CUDA framework to allow highly parallelized computation.

GOVERNING EQUATIONS

In a matrix form, the 2-D hyperbolic conservation laws of the SWEs may be written as:

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}}{\partial x} + \frac{\partial \mathbf{g}}{\partial y} = \mathbf{s} \quad (1)$$

where x and y are Cartesian coordinates; t denotes time; and \mathbf{u} , \mathbf{f} , \mathbf{g} , and \mathbf{s} are the vectors containing the conserved variables, fluxes in the x - and y -directions, and source terms, respectively. Without considering the viscous terms, surface stresses, and Coriolis effects, the vector terms may be expressed as (Liang and Borthwick, 2009):

$$\mathbf{u} = \begin{bmatrix} \eta \\ uh \\ vh \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} uh \\ u^2h + 0.5g(\eta^2 - 2\eta z_b) \\ uvh \end{bmatrix},$$

$$\mathbf{g} = \begin{bmatrix} vh \\ uvh \\ v^2h + 0.5g(\eta^2 - 2\eta z_b) \end{bmatrix}, \quad (2)$$

$$\mathbf{s} = \begin{bmatrix} 0 \\ -c_f u \sqrt{u^2 + v^2} - g\eta \frac{\partial z_b}{\partial x} \\ -c_f v \sqrt{u^2 + v^2} - g\eta \frac{\partial z_b}{\partial y} \end{bmatrix}$$

where η is the water level (stage), h is the total water depth, u and v are depth-averaged velocity components in the x - and y -directions, and z_b is the bed level above datum. The bed roughness coefficient is calculated by $C_f = gn^2/h^{1/3}$, with n denoting the Manning coefficient and $g = 9.81 \text{ m/s}^2$ as the acceleration due to gravity. Using the water level η as a flow variable, the above formulation expresses a set of prebalanced SWEs that automatically satisfy the lake-at-rest conditions for applications involving irregular domain topographies (Liang and Borthwick, 2009).

NUMERICAL SCHEME

SWEs 1 and 2 are solved using a shock-capturing finite volume Godunov-type scheme, with a two-step unsplit MUSCL-Hancock method applied to achieve second-order accuracy in both space

and time. In the predictor step, intermediate flow variables are calculated to half of a time step $\Delta t/2$ using the following formula:

$$\mathbf{u}_i^{k+1/2} = \mathbf{u}_i^k - \frac{\Delta t}{2\Delta x}(\mathbf{f}_E - \mathbf{f}_W) - \frac{\Delta t}{2\Delta y}(\mathbf{g}_N - \mathbf{g}_S) + \frac{\Delta t}{2}\mathbf{s}_i^k \quad (3)$$

where superscript k represents the time step; subscripts E , W , N , and S indicate the east, west, north, and south interfaces of the cell under consideration; i is the cell index; Δt is the time step; and Δx and Δy are the size of the cell in the x - and y -directions. The interface fluxes \mathbf{f}_E , \mathbf{f}_W , \mathbf{g}_N , and \mathbf{g}_S are directly computed from the face values of the variables at the middle point of the respective cell face, which are obtained using the MUSCL slope limited linear reconstruction based on cell-center values of the flow variables to prevent spurious oscillations of the solution in the vicinity of discontinuities or steep gradients. The minmod limiter is adopted in this work to guarantee better numerical stability.

In the corrector step, the HLLC Riemann solver is used to calculate the interface fluxes, and the flow variables are updated to a new time step using the following fully conservative time-marching formula:

$$\mathbf{u}_i^{k+1} = \mathbf{u}_i^k - \frac{\Delta t}{\Delta x}(\mathbf{f}_E - \mathbf{f}_W) - \frac{\Delta t}{\Delta y}(\mathbf{g}_N - \mathbf{g}_S) + \Delta t \mathbf{s}_i^{k+1/2} \quad (4)$$

Detailed implementation of this second-order finite volume HLLC Godunov-type scheme can be found in Liang and Borthwick (2009).

To accurately track the moving wet/dry interface and meanwhile ensure nonnegative water depth, a depth-positivity preserving technique introduced by Liang (2010) is adopted for robust simulation of tsunami inundation. Furthermore, the friction source terms are separately discretized using a pointwise implicit scheme, as adopted in Liang (2010), to improve numerical stability of the scheme for applications involving wetting and drying.

The present numerical scheme is overall explicit, and the maximum permissible time step ensuring stable simulations is controlled by the Courant–Friedrichs–Lewy (CFL) condition; i.e.:

$$\Delta t = C \min \left[\frac{\Delta x}{|u| + \sqrt{gh}}, \frac{\Delta y}{|v| + \sqrt{gh}} \right] \quad (5)$$

where $0 < C \leq 1$ is the Courant number. In this work, variable time steps predicted by Eq. 5 with $C = 0.5$ are used in all of the test cases.

Open or closed boundary conditions are imposed during simulations. For open boundaries, the flow information at the ghost points is imposed to allow zero gradients at the boundary or directly prescribed as inflow or outflow conditions. The closed boundary is implemented similarly for water level and tangential velocity/discharge but zero normal velocity/discharge at the boundary under consideration.

CUDA IMPLEMENTATION

Herein the aforementioned finite volume Godunov-type SWE model is implemented for fully parallelized computing on GPUs using the Compute Unified Device Architecture (CUDA) programming framework. Specifically, CUDA/C is adopted to develop the wholly parallelized calculation component that runs on GPUs (NVIDIA, 2012), and C++ is used to write the nonparallelized or sequential codes.

The program starts by allocating memory on the host (CPU) and the device (GPU). Then the required datasets such as topography, bathymetry, and initial conditions are loaded onto the host.

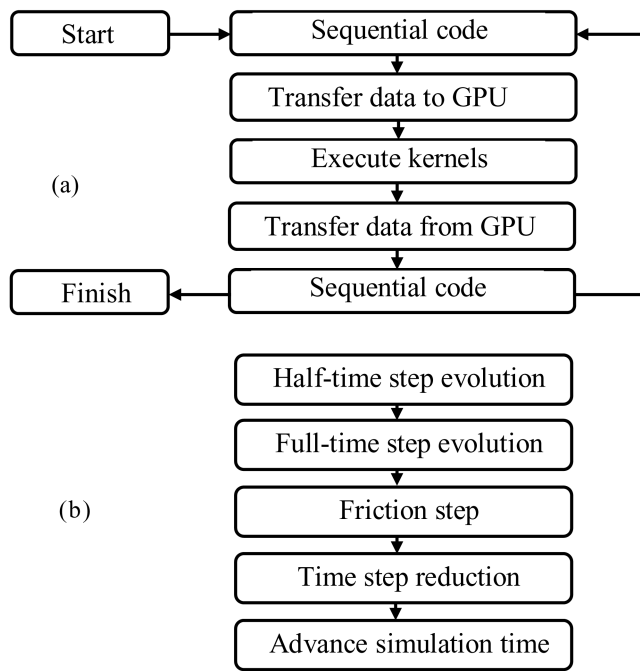


Fig. 1 Flowchart of the GPU heterogeneous parallelized program: (a) executive procedure and (b) GPU kernels

Data allocated to the memory of the host are then copied to the global memory of the GPU. The flow calculation is executed entirely on the GPU by the parallelized parts of the code via the main functions that are known as kernels, written by CUDA/C extensions. Data on the GPU are available for access by the kernels for execution. When required, the simulation results are copied from the device back to the host for postprocessing and visualization. The main executive procedure of the heterogeneous parallel program is illustrated in Fig. 1a.

In the fully parallelized calculation component, four main kernels are defined according to the aforementioned numerical scheme, including MUSCL-Hancock predictor (half time step kernel), MUSCL-Hancock corrector (full time step kernel), friction step, and time step reduction. These kernels are fully executed on the GPU, as shown in Fig. 1b. To calculate the permissible time step for advancing simulation, the reduction algorithm provided by CUDA is used, which is within the thrust library in the CUDA Toolkit (CUDA Toolkit, 2013). Each kernel launches a grid of thread blocks. Each thread has a unique local index in its block, and each block has a unique index in the grid. These blocks can be executed out of order and allow for scalability for a different number of cores in a specific device, whereas the threads in a block are executed together in groups of 32 called “warps.” Threads per block should be launched as a multiple of warp size. The potential performance of these values for a block size is discussed in Sanders and Kandrot (2012). After testing different values in the range of 32–512 threads per block, 64 or 128 threads per block show a better performance and are used in this work.

RESULTS AND DISCUSSION

The current GPU-accelerated tsunami model is first validated against the conical island tsunami benchmark test and then applied to reproduce the 2011 Japan tsunami. GPU simulations are run on a single NVIDIA Tesla M2075 card. The required run times are compared with those resulting from the simulations on a single Intel Core i5-2500 (3.3 GHz) PC using an alternative Fortran

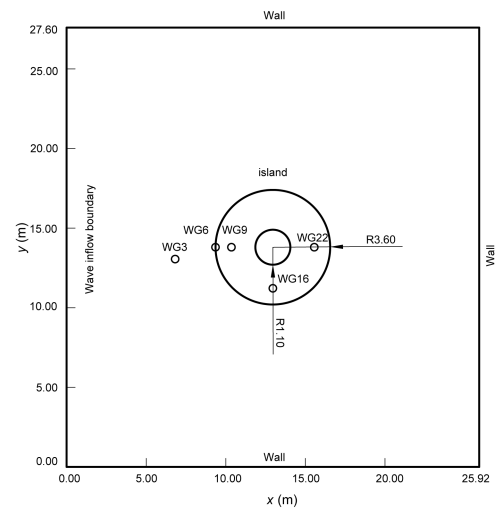


Fig. 2 Experimental layout and gauge locations

code, as reported in Liang (2010). It should be noted that the comparison of run times is only indicative because different computer languages may involve different optimization strategies for simulations, although the numerical schemes are identical for the two models. All of the calculations are carried out using a double-precision (64-bit) floating-point arithmetic.

2-D Run-Up of a Solitary Wave on a Conical Island

This experimental benchmark test of tsunami run-up onto a conical island (Briggs et al., 1995) is simulated to demonstrate the model’s capability for simulating breaking waves and complex flow hydrodynamics with wetting and drying over uneven topography. The experimental setup is illustrated in Fig. 2, where the conical island, with a base diameter of 7.2 m, top diameter of 2.2 m, and height of 0.625 m, is located near the center of a 30 m × 25 m basin. For numerical simulations, the computational domain is set to 25.92 m × 27.6 m with initial water depth of 0.32 m.

The incident wave is imposed from the left boundary at $x = 0$, in order to replicate the solitary wave generated by a wave maker. The varying wave height z and velocity u are specified as follows:

$$z(t) = H \operatorname{sech}^2 \left[\sqrt{\frac{3H}{4D^3}} C(t - T) \right], \quad u(t) = \frac{C\eta(t)}{D + \eta(t)},$$

$$v(t) = 0 \quad (6)$$

where D is the still water depth, H is the wave amplitude, T represents the time when the wave crest reaches the domain, and $C = g(D + H)^{0.5}$ is the wave celerity. The incident wave with an amplitude of $H = 0.064$ m is specifically considered herein to provide a more challenging test involving wave breaking. The corresponding still water depth is $D = 0.32$ m, and $T = 2.45$ s. Bed friction is neglected based on the findings in Liu et al. (1995). The uniform grid resolution is set to 0.04 m in order to be consistent with other works, e.g., Hubbard and Dodd (2002).

Figure 3 presents a series of 3-D water surfaces to show the interaction between the incident solitary wave and the conical island. The incident wave leads to high run-up and inundation at the front side of the island at approximately $t = 9$ s. After reaching the maximum run-up, the wave runs down the inundated region, and the refracted wave propagates around the island toward the lee side, as shown for $t = 11$ s. Then these two waves collide at the lee side, producing the second high run-up at approximately

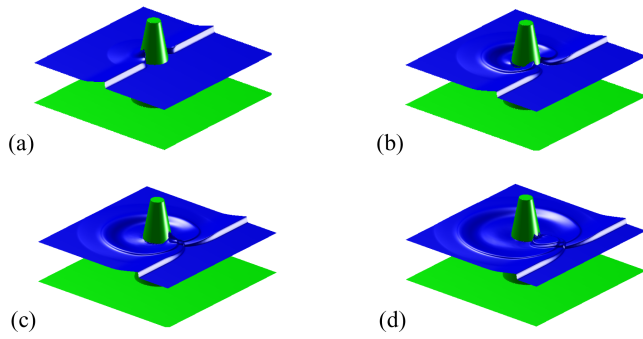


Fig. 3 Sample 3-D water surfaces at (a) $t = 9$ s, (b) $t = 11$ s, (c) $t = 12$ s, and (d) $t = 13$ s

$t = 12$ s. After that, the waves continue to propagate further in different directions around the island, as observed at $t = 13$ s. To further validate the current model, the predicted time histories of water surface elevation at five gauges are compared with experimental measurements in Fig. 4. The numerical results agree satisfactorily with measurements, although a certain level of discrepancy is also predicted. For example, at gauge 3 there is an obvious phase difference between the predicted and recorded leading waves, which is caused by the way the SWEs describe breaking

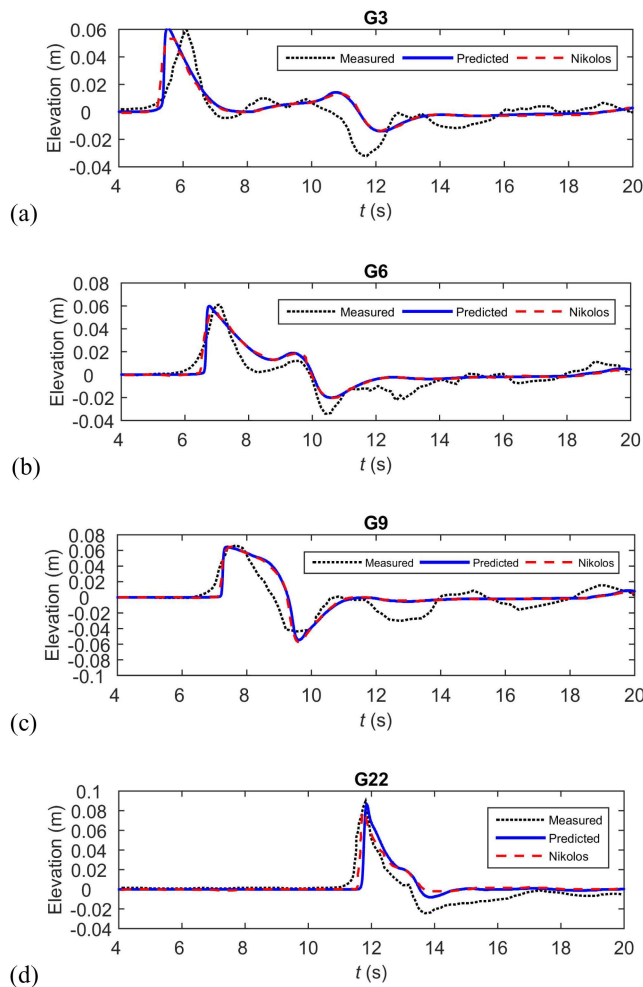


Fig. 4 Comparison of predicted surface water elevation with experimental measurements and alternative numerical solutions at different gauges: (a) gauge 3, (b) gauge 6, (c) gauge 9, and (d) gauge 22

Gauge no.	3	6	9	16	22
RMSE (m)	0.0126	0.00916	0.00702	0.00831	0.0102

Table 1 Conical island simulation: RMSE at different gauges

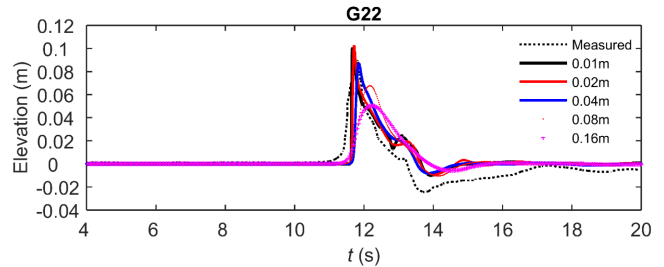


Fig. 5 Effect of grid resolution

CPU (Intel Core i5-2500)	GPU (Tesla M2075)	Speed-up
939.22 s	21.8 s	43.1x

Table 2 Conical island simulation: run times on different devices

waves. In this case, the physical incident wave breaks before arriving at the shoreline, and the SWE model simulates the breaking waves as a propagating bore. The predictions are consistent with numerical predictions reported by other researchers (e.g., Nikolos and Delis, 2009) using an unstructured grid-based finite volume Godunov-type model implemented with a Roe approximate Riemann solver. Nevertheless, the arriving time and magnitude of the leading wave are accurately reproduced, which are the most important aspects for engineering considerations. Table 1 presents the root mean square error (RMSE) calculated at the five different gauges.

To demonstrate the effect of the grid resolution on the numerical results, further simulations are run on uniform grids of finer and coarser resolutions, i.e., 0.01 m, 0.02 m, 0.08 m, and 0.16 m, respectively. The predicted time histories of water surface elevation predicted by the different simulations are shown in Fig. 5 for gauge 22 at the lee side of the island. The simulation results appear to be convergent in capturing the peak with increasing grid resolution.

The performance of the current GPU tsunami model is evaluated by comparing the run times of different simulations (20 s of simulation with 0.04 m grid resolution) on different devices. As indicated in Table 2, the GPU simulation is approximately 43 times more efficient than the run on a single CPU core using the Fortran code of the model (Liang, 2010).

Simulation of the 2011 Japan Tsunami

The Tohoku-Oki $M_w = 9.0$ earthquake triggered a mega-tsunami in East Japan on 11 March 2011, causing over 15,000 casualties and 220 billion U.S. dollars of damage. The present GPU tsunami model is used to reproduce this tsunami event to demonstrate its superior performance for real-world applications. Figure 6 presents the 1,350 km \times 1,822.5 km computational domain. The resolution of the bathymetry/topography data used for simulations is respectively 1,350 m and 450 m. A constant Manning coefficient of 0.025 is used across the whole domain. Initial water surface displacement initiating the tsunami is calculated using the Okada rectangular fault model (Okada, 1985) as provided in Clarke et al. (1997). The fault information and parameters are similar to those reported in Fujii et al. (2011), assuming

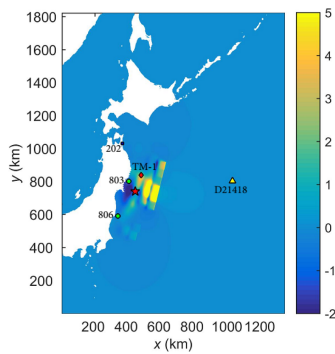


Fig. 6 Initial water surface displacement and location of gauges

instantaneous rupture. The tsunami source is divided into 40 sub-faults, each $50 \text{ km} \times 50 \text{ km}$, covering the whole affected area. The focal mechanisms of the subfaults were taken from the USGS Wphase moment tensor solution and are as follows: strike, 193° ; dip, 14° ; and slip, 81° . The top depth is assumed to be 0 km, 12.1 km, 24.2 km, and 36.3 km for near-trench, shallow, middle, and deep subfaults, respectively.

The tsunami event is simulated for 6 hours, using the current GPU hydrodynamic model on a grid with 12,150,000 cells of 450 m in resolution. Figure 7 presents the tsunami wave propagation in the first 20 minutes. After being initiated by the earthquake, the tsunami wave propagates radially into the deep ocean and toward the east coast of Japan. The first leading high wave reaches the coast in approximately 20 minutes, consistent with records at the wave gauges.

For this event, field records of water surface elevation are available in a number of gauge stations of different types. The measurements from five gauges, as detailed in Table 3, are used in this work to verify the model results. These include one wave gauge close to the coast (202), two near-shore GPS buoys (803 and 806),

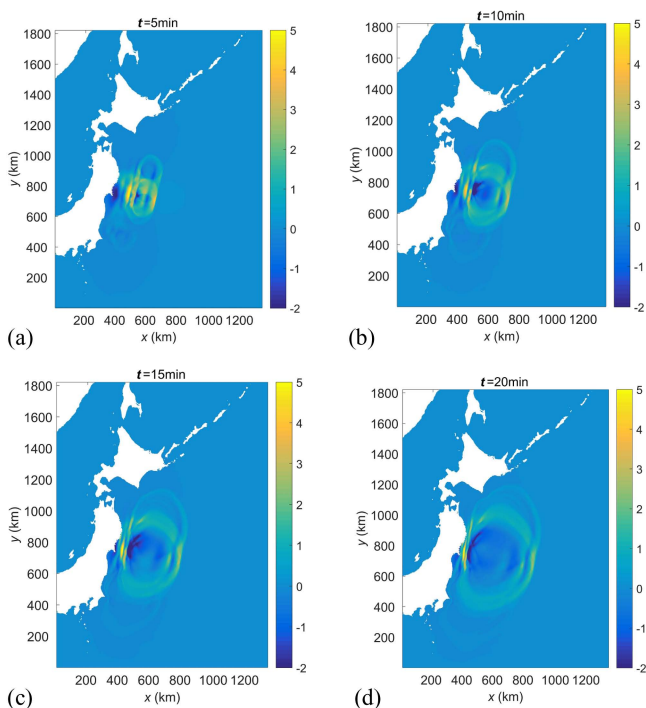


Fig. 7 Propagation of tsunami wave: (a) $t = 5 \text{ min}$, (b) $t = 10 \text{ min}$, (c) $t = 15 \text{ min}$, and (d) $t = 20 \text{ min}$

Gauge	Type	Depth (m)
803 Miyagi North	GPS buoy	160
806 Fukushima	GPS buoy	137
202	NOWPHAS	44
TM-1	Pressure gauge	1,600
D21418	Tsunamieter	5,660

Table 3 Sample gauges where records are available for comparison

one cabled pressure gauge (TM-1), and one DART buoy (D21418) approximately 500 km offshore, away from the epicenter.

Figure 8 shows the comparison between the simulation results and field measurements at these gauges. Specifically, the maxi-

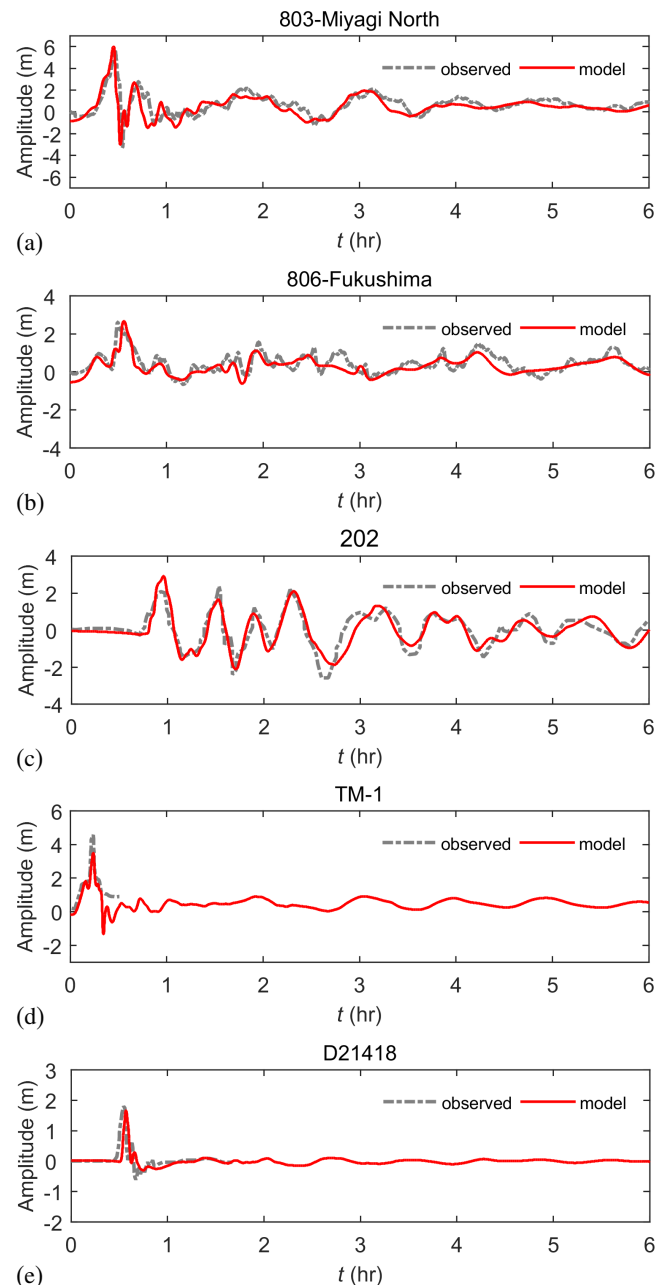


Fig. 8 Comparison of observed and simulated wave time series at five gauges: (a) gauge 803, (b) gauge 806, (c) gauge 202, (d) gauge TM-1, and (e) gauge D21418

Resolution	Number of cells	CPU Intel	GPU Tesla
1,350 m	1,350,000	60 min	1 min (~ 60×)
450 m	12,150,000	2 days	45 min (~ 64×)

Table 4 Japan tsunami simulations: run times on different devices

mum wave amplitude is approximately 6 m at gauge 803 (Miyagi North), which is well captured by the model. Despite a small shift of phase, the model predicts reasonably well the wave series at gauge 806 (Fukushima). At the near-shore gauge, 202, where the water depth is only 44 m, the model prediction shows good agreement with the actual record in both amplitude and phase. The pressure gauge TM-1 located in the medium depth of 1,600 m recorded a wave peak of more than 4 m at 18 min after the earthquake. Although the waveform is successfully reproduced, the peak is slightly underestimated by the current model. The underestimation of wave peak at this gauge was also reported by Fujii et al. (2011) using a different model. At gauge D21418, which is located offshore at a depth of over 5,000 m, a peak of 1.64 m was recorded at approximately 33 min after the earthquake, which is the largest tsunami wave ever recorded by a deep-ocean tsunamieter. Again, the current model successfully predicts the waveform, the arrival time, and the depression. Overall, the model reproduced reasonably well the first dominant wave in all of the gauges, as well as the rest of the wave series in those gauges with data available for comparison. The current numerical results also compare favorably with model predictions presented by other researchers, e.g., Fujii et al. (2011) and Wei et al. (2013).

To further demonstrate the performance of the current GPU model, the run times for simulations on grids of different resolutions (i.e., 1,350 m and 450 m) are compared for different hardware devices. The simulations are carried out for the first 70 min of the tsunami event to allow reasonable run time of the CPU runs. Table 4 details the run time comparison. The coarse-resolution simulation requires only 1 min run time on a single GPU despite 1.35 million cells being involved in the computation. On the other hand, the same simulation on a single CPU using the Fortran code takes 1 hour to complete, showing 60 times speed-up by the GPU model. For the fine-resolution simulations involving 12.15 million cells, the GPU model only needs 45 min of run time whereas the CPU model takes approximately 2 days, giving a speed-up of 64 times.

CONCLUSIONS

In this paper a hydrodynamic model based on GPU parallel computing has been presented for tsunami simulations. The model solves the 2-D nonlinear SWEs using a MUSCL-Hancock second-order finite-volume Godunov-type scheme incorporated with an HLLC approximate Riemann solver. The model is capable of simulating tsunami propagation and run-up involving advancing bores and moving shorelines in the inundation zone over irregular topographies. The model has been applied to reproduce a laboratory-scale tsunami test and the 2011 Japan tsunami. Model predictions compared well with laboratory measurements, field records, and alternative numerical results whenever available.

The improved performance of the current GPU model has been demonstrated by comparing with a Fortran code based on the identical numerical scheme that runs on a single CPU core. When simulating the laboratory-scale tsunami test, the GPU model is over 40 times more efficient than the CPU code. When reproducing the 2011 Japan event, the GPU-accelerated model is more than

60 times faster for both the coarse-resolution simulation involving 1.35 million cells and the fine-resolution simulation involving 12.15 million cells.

REFERENCES

- Amouzgar, R, Liang, Q, and Smith, L (2014). "A GPU-Accelerated Shallow Flow Model for Tsunami Simulations," *Proc ICE Eng Comput Mech*, 167(2), 117–125. <http://dx.doi.org/10.1680/eacm.13.00024>.
- Briggs, M, Synolakis, C, Harkins, G, and Green, D (1995). "Laboratory Experiments of Tsunami Runup on a Circular Island," *Pure Appl Geophys*, 144(3–4), 569–593. <http://dx.doi.org/10.1007/BF00874384>.
- Brodtkorb, AR (2010). *Scientific Computing on Heterogeneous Architectures*, PhD Thesis, University of Oslo, Norway.
- Clarke, PJ, et al. (1997). "Geodetic Investigation of the 13 May 1995 Kozani–Grevena (Greece) Earthquake," *Geophys Res Lett*, 24(4), 707–710. <http://dx.doi.org/10.1029/97GL00430>.
- CUDA Toolkit (2013). *CUDA Toolkit Documentation*. Available at: <http://docs.nvidia.com/cuda/thrust/index.html> (Accessed: 20/3/2014).
- Delis, AI, and Mathioudakis, EN (2009). "A Finite Volume Method Parallelization for the Simulation of Free Surface Shallow Water Flows," *Math Comput Simul*, 79(11), 3339–3359. <http://dx.doi.org/10.1016/j.matcom.2009.05.010>.
- Fujii, Y, Satake, K, Sakai, S, Shinohara, M, and Kanazawa, T (2011). "Tsunami Source of the 2011 Off the Pacific Coast of Tohoku Earthquake," *Earth Planets Space*, 63(7), 815–820. <http://dx.doi.org/10.5047/eps.2011.06.010>.
- Goto, C, Ogawa, Y, Shuto, N, and Imamura, F (1997). *Numerical Method of Tsunami Simulation with the Leapfrog Scheme (IUGG/IOC Time Project)*, IOC Manual, UNESCO, No 35.
- Hubbard, ME, and Dodd, N (2002). "A 2D Numerical Model of Wave Run-Up and Overtopping," *Coastal Eng*, 47(1), 1–26. [http://dx.doi.org/10.1016/S0378-3839\(02\)00094-7](http://dx.doi.org/10.1016/S0378-3839(02)00094-7).
- Kuo, FA, Smith, MR, Hsieh, CW, Chou, CY, and Wu, JS (2011). "GPU Acceleration for General Conservation Equations and Its Application to Several Engineering Problems," *Comput Fluids*, 45(1), 147–154. <http://dx.doi.org/10.1016/j.compfluid.2010.10.007>.
- Leveque, RJ, George, DL, and Berger, MJ (2011). "Tsunami Modelling with Adaptively Refined Finite Volume Methods," *Acta Numerica*, 20, 211–289. <http://dx.doi.org/10.1017/S0962492911000043>.
- Liang, Q (2010). "Flood Simulation Using a Well-Balanced Shallow Flow Model," *J Hydraul Eng*, ASCE, 136(9), 669–675. [http://dx.doi.org/10.1061/\(ASCE\)HY.1943-7900.0000219](http://dx.doi.org/10.1061/(ASCE)HY.1943-7900.0000219).
- Liang, Q, and Borthwick, AGL (2009). "Adaptive Quadtree Simulation of Shallow Flows with Wet-Dry Fronts over Complex Topography," *Comput Fluids*, 38(1), 221–234. <http://dx.doi.org/10.1016/j.compfluid.2008.02.008>.
- Liang, Q, Hou, J, and Amouzgar, R (2015). "Simulation of Tsunami Propagation Using Adaptive Cartesian Grids," *Coastal Eng J*, 57(3), 1550016. <http://dx.doi.org/10.1142/S0578563415500163>.
- Liu, PLF, Cho, Y-S, Briggs, MJ, Kanoglu, U, and Synolakis, CE (1995). "Runup of Solitary Waves on a Circular Island," *J Fluid Mech*, 302, 259–285. <http://dx.doi.org/10.1017/S0022112095004095>.

- NVIDIA (2012). *CUDA C Programming Guide*. Available at: <http://docs.nvidia.com/cuda/cuda-c-programming-guide/> (Accessed: 05/05/2016).
- Nikolos, IK, and Delis, AI (2009). "An Unstructured Node-Centered Finite Volume Scheme for Shallow Water Flows with Wet/Dry Fronts over Complex Topography," *Comput Methods Appl Mech Eng*, 198(47-48), 3723–3750. <http://dx.doi.org/10.1016/j.cma.2009.08.006>.
- Okada, Y (1985). "Surface Deformation due to Shear and Tensile Faults in a Half-Space," *Bull Seismol Soc Am*, 75(3), 1135–1154.
- Pophet, N, Kaewbanjak, N, Asavanant, J, and Ioualalen, M (2011). "High Grid Resolution and Parallelized Tsunami Simulation with Fully Nonlinear Boussinesq Equations," *Comput Fluids*, 40(1), 258–268. <http://dx.doi.org/10.1016/j.compfluid.2010.09.030>.
- Popinet, S (2011). "Quadtree-Adaptive Tsunami Modelling," *Ocean Dyn*, 61(9), 1261–1285. <http://dx.doi.org/10.1007/s10236-011-0438-z>.
- Rossinelli, D, Hejazialhosseini, B, Spampinato, DG, and Koumoutsakos, P (2011). "Multicore/Multi-GPU Accelerated Simulations of Multiphase Compressible Flows Using Wavelet Adapted Grids," *SIAM J Sci Comput*, 33(1), 512–540. <http://dx.doi.org/10.1137/100795930>.
- Sanders, J, and Kandrot, E (2012). *CUDA by Example: An Introduction to General-Purpose GPU Programming*, Addison-Wesley Professional, 312 pp.
- Schive, HY, Zhang, UH, and Chiu, T (2012). "Directionally Unsplit Hydrodynamic Schemes with Hybrid MPI/OpenMP/GPU Parallelization in AMR," *Int J High Perform Comput Appl*, 26(3), 367–377. <http://dx.doi.org/10.1177/1094342011428146>.
- Smith, LS, and Liang, Q (2013). "Towards a Generalised GPU/CPU Shallow-Flow Modelling Tool," *Comput Fluids*, 88(0), 334–343. <http://dx.doi.org/10.1016/j.compfluid.2013.09.018>.
- Titov, VV, and Synolakis, CE (1995). "Modeling of Breaking and Nonbreaking Long-Wave Evolution and Runup Using VTCS-2," *J Waterw Port Coastal Ocean Eng*, ASCE, 121(4), 308–316. [http://dx.doi.org/10.1061/\(ASCE\)0733-950X\(1995\)121:6\(308\)](http://dx.doi.org/10.1061/(ASCE)0733-950X(1995)121:6(308)).
- Wang, P, Abel, T, and Kaehler, R (2010). "Adaptive Mesh Fluid Simulations on GPU," *New Astron*, 15(7), 581–589. <http://dx.doi.org/10.1016/j.newast.2009.10.002>.
- Wang, X, and Liu, PLF (2006). "An Analysis of 2004 Sumatra Earthquake Fault Plane Mechanisms and Indian Ocean Tsunami," *J Hydraul Res*, 44(1), 147–154. <http://dx.doi.org/10.1080/00221686.2006.9521671>.
- Wei, Y, Chamberlin, C, Titov, VV, Tang, L, and Bernard, EN (2013). "Modeling of the 2011 Japan Tsunami: Lessons for Near-Field Forecast," *Pure Appl Geophys*, 170(6–8), 1309–1331. <http://dx.doi.org/10.1007/s00024-012-0519-z>.

Proceedings of the 10th (2013) ISOPE Ocean Mining & Gas Hydrates Symposium

Szczecin, Poland, September 22–26, 2013

DEEP-OCEAN MINERALS AND PROCESSING, EXPLORATION AND ENVIRONMENT, DEEP-OCEAN MINING SYSTEMS AND TECHNOLOGY (Mining Systems, Ship, Pipe, Nodule Lift, Buffer, Link, Oceanfloor Miner, and Miner Control)

GAS HYDRATES (Fundamentals, Properties, Geotechnical and Geochemical Characteristics, Development)

The Proceedings (ISBN 978-1-880653-92-0; ISSN 1946-0066): \$100 (ISOPE Member: \$80) in a single volume (CD-ROM) is available from www.isoape.org or www.deeppoceanmining.org. ISOPE, P.O. Box 189, Cupertino, California 95015-0189, USA (Fax +1-650-254-2038; orders@isoape.org)