

Wave Propagation for Next-Generation Supercomputers - Proposal Update

David Ketcheson ^{*} Aron Ahmadi[†]

October 2, 2011

Requested allocation: 12M-15M core-hours

Long-time asymptotics of solitary waves 3M

Interactions of cylindrical solitary waves 3M

Time-reversal of solitary waves 3M

Contingencies in solitary wave studies 3M

Algorithmic testing and preliminary exploration of new applications
0-3M

1 Achievements

This project entered development status in August 2010 and production status in February 2011. Since then, progress on several fronts has proceeded very rapidly.

1.1 Numerical algorithms and code development

The PetClaw code was extended to 2D, including both the dimensionally split and fully multidimensional Clawpack algorithms. The high order algorithms of SharpClaw and PyWENO have also been incorporated, including solution

^{*}Mathematical and Computer Sciences and Engineering Division

[†]KAUST Supercomputing Laboratory

algorithms of up to 17th order. This is, to our knowledge, the first time that any of these algorithms have been parallelized at this scale. The code has been demonstrated to scale up to all of Shaheen with greater than 95% efficiency. We have extensively characterized the code up to the full machine with a comprehensive analysis of both weak and strong scaling and their dependence on problem size, I/O, CFL synchronization, and kernel complexity. A regression testing framework has been incorporated into the code. The code repository was migrated to GitHub and is freely available worldwide. A Google group was established for users and developers, and is host to frequent discussions. The code has attracted new users with no relation to KAUST or any of the developers. The development team has grown to include about a dozen people at KAUST, ETH Zurich, Argonne National Laboratories, U. Chicago, UT Austin, U. Washington (Seattle), CIMEC (Argentina), UC Davis, Texas A&M, U. North Carolina, and American University of Beirut. The team makes extensive use of an issue tracker on Github. Substantial documentation, including tutorials and reference documentation has been written and is available at <http://numerics.kaust.edu.sa/pyclaw>. A GPGPU version of the code has recently been developed and is fully functional, in preparation for targeting hybrid architectures; that development will be featured in a poster at Supercomputing 2011.

1.2 Scholarly works

The scholarly outcomes from this work are equally substantial. This work has been an integral part of two KAUST MS theses [6, 2] (one completed, one nearing completion). Two conference papers have been produced [4, 1], and two journal articles are nearly complete [3, 5]. One of them, planned for submission to the prestigious SIAM Journal on Scientific Computing (SISC), was directly solicited by the journal's chief editor, Hans Petter Langtangen. Talks about this work were presented at ten international conferences and invited colloquia (see Appendix A). In March, an international workshop based around PetClaw development was held at KAUST (co-sponsored by KSL), with 15 international visitors and numerous KAUST participants; this catalyzed much of the development work mentioned above. The workshop and the code were featured in a long article in *SIAM News*, the periodical of the world's leading applied math society.

1.3 Scientific advances

The scientific outcomes of the previous allocation include two main areas. The first is the demonstration of Python as a legitimate supercomputing language. While many other parallel Python codes exist, they typically are only demonstrated to scale up to relatively small numbers of cores. PyClaw is one of only two codes based on Python that is demonstrated to scale to 65K cores, and our approach to parallelization is unique in that we rely extensively on PETSc through its Python bindings. As a result, the total parallel code in our work is less than 300 lines of Python.

The principal scientific outcome of this work has been the discovery and understanding of new classes of two-dimensional solitary waves occurring in periodic non-dispersive media. These waves are remarkable because solitary waves typically result from a balance between nonlinearity and dispersion, yet these arise in the absence of any dispersive terms. Rather, they rely on effective dispersion arising from material inhomogeneities. In particular, we have been studying solutions of the 2D p-system in media with “checkerboard” (piece-wise constant) or sinusoidally-varying material properties. Under suitable conditions, arbitrary initial conditions resolve after a short time into trains of solitary waves. These solitary waves exhibit striking self-similarity properties, similar to those of solutions of the cylindrical KdV equation. Furthermore, the appearance of shocks (which normally would occur after a short time) is strongly suppressed in these periodic media – a result that in a sense overturns a fundamental tenet of our understanding of nonlinear hyperbolic PDEs and demands development of a new mathematical theory.

The study of these waves is computationally demanding. In order to see them begin to appear, the domain must include at least 200 material periods, and each period must be discretized into at least 100 computational cells (in each direction) in order to achieve even qualitatively correct results. Since the system under consideration involves three equations, this means a bare minimum of $3 \times (200)^2 \times (100)^2 = 1.2 \times 10^9$ unknowns. Of course, the memory considerations are insignificant, but the CFL condition implies that such a simulation requires roughly 5×10^5 timesteps (note that implicit methods are useless because these are time-dependent nonlinear waves that must be resolved accurately). Meanwhile, the numerical simulation of nonlinear waves in spatially varying (even discontinuous) domains is already a highly challenging area of numerical research, and the necessary algorithms are much more expensive than those needed for, e.g., linear wave simulations

commonly used in electromagnetics or seismics.

2 Dynamic Loading and Python

One challenge of using Python is the need for dynamic loading of libraries and shared objects at run-time. Under the current setup on Shaheen, this results in a noticeable cost at the beginning of each simulation. Aron Ahmadi of KSL is working with international collaborators to alleviate this. The cost of dynamic loading is less than 5% for large-scale wave propagation simulations in which waves must traverse the whole domain at least once.

3 Resource Needs

Our consumption of core-hours has accelerated over the summer since parallel scaling was demonstrated and scientific runs began in earnest. A few large, carefully planned and targeted simulations are required to complete the current phase of this work. Planning for these runs is the result of days of careful asymptotic analysis and hundreds of experiments on workstations, followed by moderately large runs on Shaheen. Already waiting in the queue (with insufficient core hours) is a 2D solitary wave simulation on a 400×400 domain, large enough to enable us to isolate these waves completely for the first time, in order to study interactions between them. That simulation will require 4 racks for 8 days, or approximately 3M core hours. A similar number will be required for a second run to investigate head-on collision of concentric solitary wave rings, and another 3M to study the backward propagation of a single pulse on the same domain, as a conclusive demonstration of its solitary wave nature. We are requesting a further 3M in anticipation of unforeseen challenges or opportunities. This allocation will allow us to finish and submit the work on cylindrical solitary waves, which will be the first publication coming from this work that is focused entirely on the science rather than on computing aspects (a significant milestone). We expect to consume this allocation within the next 1-2 months.

Based on the numerous, innovative, and substantial achievements listed above, and if the RAC sees fit, we would appreciate an additional allocation (up to 3M hours, for a total of 15M) for further contingencies and perhaps for very early exploratory work on some of the topics listed in the next

section, in order to assess their feasibility. Any left-over hours will be used in ongoing work to develop, test, and scale PyClaw and its kernels, including more visualization and I/O-efficient techniques for grid coarsening and gauge extraction.

4 Future Work

Whereas this project began as an attempt to extend the widely-used Clawpack software, it is now safe to say that this project is a principal driver of Clawpack development. In February 2012, another international workshop will be held at KAUST, focused on continued development of our code. The main Clawpack developers will attend, as well as a range of experts in HPC and hyperbolic PDEs. This workshop will also aim to work with KAUST researchers in the areas of seismic modeling and flooding. Newly funded AEA projects in these areas (both of which include Prof. Ketcheson as a co-I) will require use of the code.

New production proposals will be submitted as soon as possible to request allocations for the following topics: high-resolution simulations of Rossby waves on the sphere; testing of scaling of the 3D code; investigation of 3D solitary waves in periodic media; and initial forays into modeling of shallow water and seismic wave problems as part of AEA-funded projects.

A Conference presentations

Presentations of work based on our previous allocation were given at the following:

1. SpringSim 2011 (Boston, Jan. 2011)
2. [HPC]³ Workshop at KAUST (March 2011)
3. ASIM Workshop (Technische Universitt Mnchen, March 2011)
4. Workshop on simulation and optimization (invited, Hungary, May 2011)
5. KAUST-IAMCS Workshop on Multiscale Modeling, Advanced Discretization Techniques, and Simulation of Wave Propagation (invited, May 2011)

6. The Closest Point Method Workshop (Canada, July 2011)
7. ICIAM 2011 (Canada, July 2011)
8. WAVES 2011 (Canada, July 2011)
9. Brigham Young University invited mathematics colloquium (Sept. 2011)
10. SciPy 2011 (Austin, July 2011)

References

- [1] A. Alghamdi, Aron Ahmadi, D.I. Ketcheson, M.G. Knepley, K.T. Mandli, and L. Dalcin. PetClaw: A Scalable Parallel Nonlinear Wave Propagation Solver for Python. In *Proceedings of SpringSim '11*. ACM, 2011.
- [2] Amal Alghamdi. *PetClaw: Parallelization and Performance Optimization of a Pythonic Nonlinear Wave Propagation Solver Using PETSc*. Ms thesis, forthcoming, KAUST, 2011.
- [3] David I. Ketcheson, Kyle T Mandli, Aron Ahmadi, Amal Alghamdi, Manuel Quezada, Matteo Parsani, Matthew G. Knepley, and Matthew Emmett. An Accessible Extensible Parallel Wave Propagation Solver for General Hyperbolic PDEs. 2011.
- [4] K. Mandli, A. Alghamdi, Aron Ahmadi, D.I. Ketcheson, and W. Scullin. Using Python to Construct a Scalable Parallel Nonlinear Wave Solver. In *SciPy 2011*, 2011.
- [5] Manuel Quezada and David I Ketcheson. Radial solitary waves in two-dimensional periodic media. 2011.
- [6] Manuel Quezada de Luna. *Nonlinear Wave Propagation and Solitary Wave Formation in Two-Dimensional Heterogeneous Media*. Masters, King Abdullah University of Science and Technology (KAUST), 2011.