BULETINUL Universităţii Petrol – Gaze din Ploieşti	Vol. LXI No. 3/2009	255 - 260	Seria Tehnică

Experimenting with Visualization on Desktop Grids

Zoran Constantinescu*, Monica Vlădoiu**

* ZealSoft SRL, 11 Tg. Neamţ Street, Bucureşti, ROMÂNIA
e-mail: zoran@unde.ro

** Petroleum – Gas University of Ploieşti, 39 Bucureşti Blvd., Ploieşti, ROMÂNIA
e-mail: mvladoiu@upg-ploiesti.ro

Abstract

Scientists and engineers develop software systems that implement the models of the systems being studied and run these programs with various sets of input parameters. Typically, these models require massive amounts of calculations (usually floating-point) and are often executed on supercomputers or distributed computing platforms. Visualization could help overcome the dilemma of having information, but not the right interpretation for it. To prove the usability of desktop grid platforms in scientific computing and visualization we present here a visualization experiment that uses QADPZ, a desktop grid computing platform, to model a real world problem: geophysical circulation modeling within the Trondheim fjord.

Key words: *distributed computing, desktop grid computing, scientific computing and visualization*.

Scientific Computing, Visualization and Desktop Grids

Scientific Computing is concerned with constructing mathematical models and numerical solution techniques, and with using computers to analyze and solve scientific and engineering problems. In practical use, it is typically the application of computer simulation and other forms of computation to problems in various scientific and engineering disciplines [11]. The scientific computing approach is to gain understanding, mainly through the analysis of mathematical models implemented on computers. As Richard Hamming has observed many years ago, “the purpose of scientific computing is insight, not numbers” [14].

Scientists and engineers develop software systems that implement the models of the systems being studied and run these programs with various sets of input parameters. Typically, these models require massive amounts of calculations (usually floating-point) and are often executed on supercomputers or distributed computing platforms. Visualization could help overcome the dilemma of having information, but not the right interpretation for it [3, 9, 10]. Interactive computing and visualization would be an invaluable aid during the scientific discovery process, as well as a useful tool for gaining insight into scientific anomalies or computational errors. Scientist needs an alternative to numbers. A cognitive possibility and technical reality is the use of images. The ability of scientists to visualize complex computations and simulations is absolutely essential to ensure the integrity of the analysis, to provoke insights, and to communicate about them with others [1, 4].

With the rapid and simultaneous advances in software and computer technology, especially commodity computing, supercomputing and grid computing, every scientist and engineer will have on his or her desk an advanced simulation kit of tools that will make analysis, product development, and design more optimal and cost effective. Through the availability of increasingly powerful computers with increasing amounts of internal and external memory, it is possible to investigate incredibly complex dynamics by means of ever more realistic simulations. However, this brings with it vast amounts of data [8]. To analyze these data it is imperative to have software tools which can visualize these multi-dimensional data sets. Comparing this with experiment and theory it becomes clear that visualization of scientific data is useful yet difficult. For complicated, time-dependent simulations, the running of the simulation may involve the calculation of many time steps, which requires a substantial amount of CPU time, and memory resources are still limited, one cannot save the results of every time step [2, 4]. Hence, it will be necessary to visualize and store the results selectively in real time so that we do not have to recompute the dynamics if we want to see the same scene again. Real time means that the selected time step will be visualized as soon as it has been calculated.

Due to the huge number of PCs in the world, desktop grid and volunteer computing can (and do) supply more computing power to science than does any other type of computing. This power enables scientific research that could not be done otherwise. This advantage will increase over time, because the laws of economics dictate that consumer electronics (PCs and game consoles) will advance faster than more specialized products, and that there will simply be more of them. To prove the usability of desktop grid platforms in scientific computing and visualization we present here a visualization experiment that uses QADPZ, a desktop grid computing platform, we have developed [6] to model a real world problem: geophysical circulation modeling within the Trondheim fjord.

Real world problem – Trondheim fjord

Geophysical circulation modeling is an increasingly important area for several reasons. One is the growing concern for environmental and ecological issues. This relates to problems of different scales, from global issues to more local questions about water pollution in coastal areas, estuaries, fjords, lakes, etc. In order to analyze such problems, there is a need to predict the flow circulation and transport of different materials, either suspended in water or moving along the free surface or bottom. The numerical model is based on a finite element formulation. It is believed that the finite element flexibility is advantageous for applications in restricted waters, where the topography is usually complex. The basic mathematical formulation is given by the Navier-Stokes equations [21].

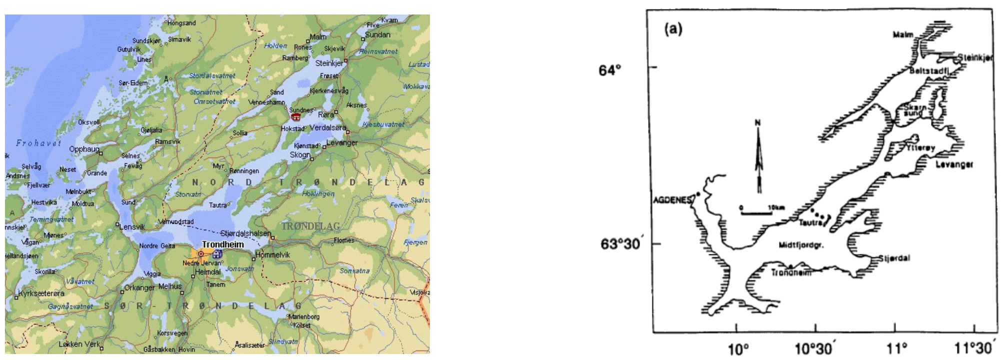

Fig. 1. Maps of the Trondheim fjord.

The figures above show a map of Trondheimsfjorden, a typical Norwegian fjord that is located on the coast of central Norway (Figure 1). Detailed topographical data are used to interpolate the depth data to the element mesh. Figure 1 illustrates the topography of the actual domain. The horizontal element mesh is shown in Figure 2. It consists of 813 bi-quadratic elements with 3683 nodes, and there are 17 levels in the vertical direction with fine grading close to the bottom boundary. This grid is assumed to be detailed enough to describe the main flow field of the Trondheim fjord. Shading the discretized cells according to the value of the scalar data field does color coding. For better appreciation of continuum data the color allocation is linearly graduated. Using directed arrows also represents the velocity vector field (Figure 3 to Figure 6).

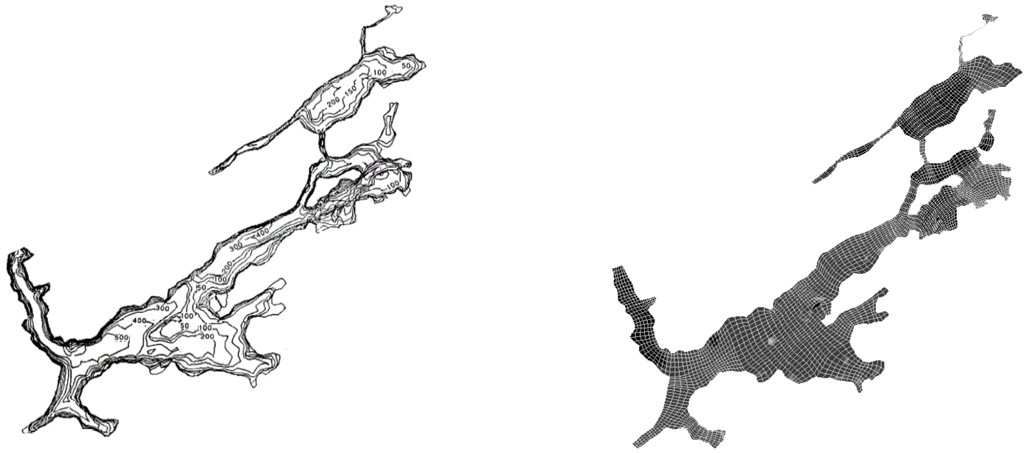

Fig. 2. Trondheim fjord (left: topography, right: grid).

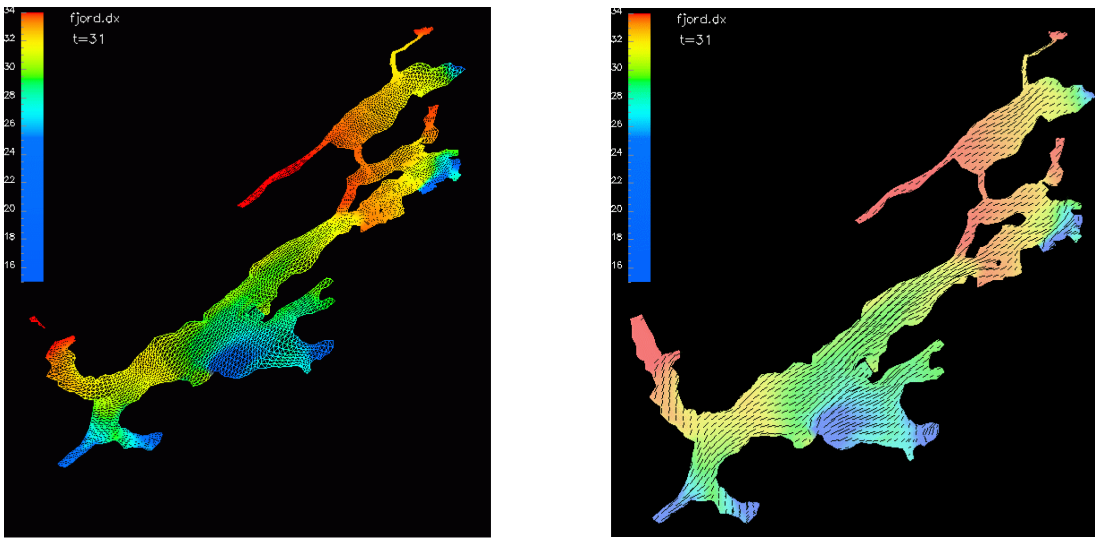

Fig. 3. Grid colored by salinity concentration (max 34 ppt - parts per thousand - mg/l).

Large vector fields, vector fields with wide dynamic ranges in magnitude, and vector fields representing turbulent flows can be difficult to visualize effectively using common techniques such as drawing arrows or other icons at each data point or drawing streamlines. Drawing arrows of length proportional to vector magnitude at every data point can produce cluttered and confusing images. In areas of turbulence, arrows and streamlines can be difficult to interpret [13, 18]. Line Integral Convolution (LIC) is a powerful technique for imaging and animating vector fields [5, 7]. The image is created beginning with a white noise that is then convoluted along integral lines of the given vector field. That creates a visible correlation between image pixels that lie on the same integral line. The local nature of the LIC algorithm suggests a parallel implementation. Such an implementation could, in principle, compute all pixels simultaneously. This would allow for interactive generation of periodic motion animations and special effects.

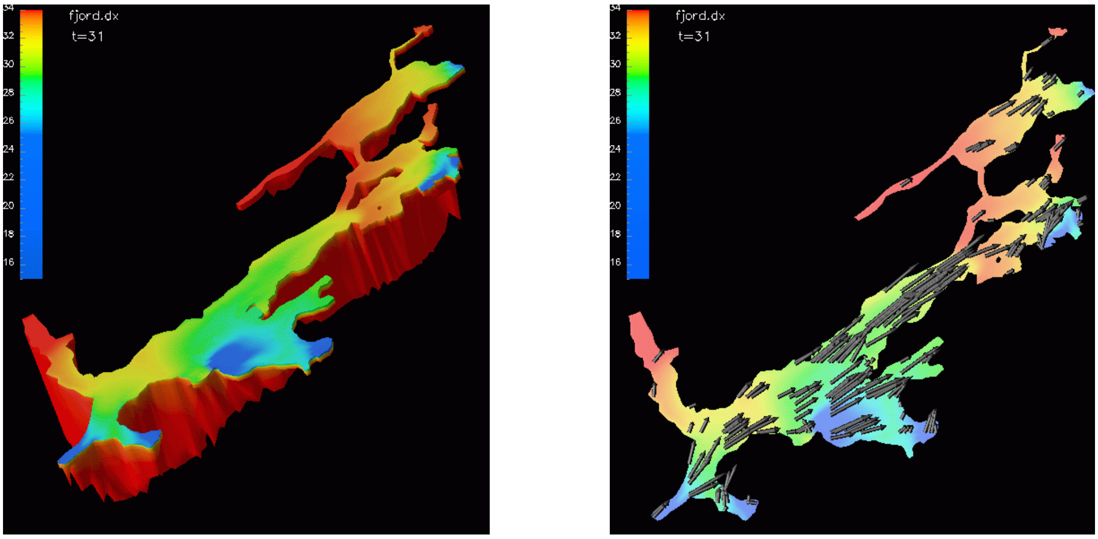

Fig. 4. Trondheim fjord model.

Note. Color by salinity concentration - 3D model representation with isosurface representation.

Fig. 5. Velocity vector field – LIC representation.

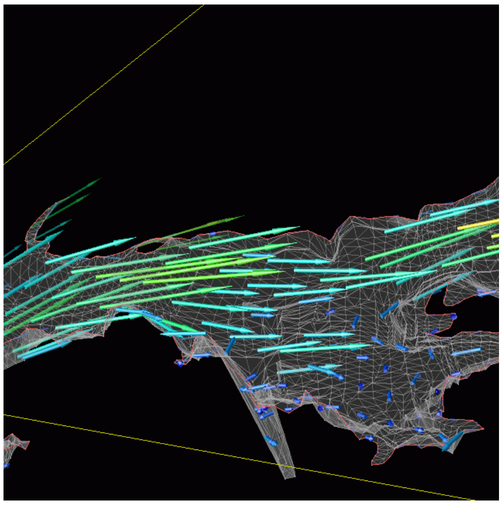

Fig. 6. 3D representation surface – vector field top layer.

Conclusions

Scientific computing programs often model real-world changing conditions, such as weather, air flow around a plane, automobile body distortions in a crash, the motion of stars in a galaxy, an explosive device, etc. Such programs might create a 'logical mesh' in computer memory where each item corresponds to an area in space and contains information about that space relevant to the model [11]. For example in weather models, each item might be a square kilometer; with land elevation, current wind direction, humidity, temperature, pressure, etc. The program would calculate the likely next state based on the current state, in simulated time steps, solving equations that describe how the system operates, and then repeat the process to calculate the next state.

Scientific breakthroughs depend on insight. In our collective experience, better visualization of a problem leads to a better understanding of the underlying science, and often to an appreciation of something profoundly new and unexpected [16, 17]. Advanced capabilities for visualization may prove to be as critical as the existence of the supercomputers themselves for scientists and engineers, and also for specialists in other domains. Better visualization tools would enhance human productivity and improve efficiency in several areas of science, industry, business, medicine and government. The most exciting potential of widespread availability of visualization is the insight gained and the mistakes caught by spotting visual anomalies while computing. Visualization will put the scientist into the computing loop and change the way the science is done [14].

Usually, complex computational and visualization algorithms require large amounts of computational power. The computing power of a single desktop computer is insufficient for running such complex algorithms, and, traditionally, large parallel supercomputers or dedicated clusters were used for this job. However, very high initial investments and maintenance costs limit the availability of such systems. A more convenient solution, which is becoming more and more popular is based on the use of non-dedicated desktop PCs in a desktop grid computing environment [12, 15]. This is done by harnessing idle CPU cycles, storage space and other resources of networked computers to work together on a particularly computational intensive application. Increasing power and communication bandwidth of desktop computers provides for this solution. In a desktop grid system, the execution of an application is orchestrated by a central scheduler node which distributes the tasks amongst the worker nodes and awaits workers' results. It is important to note that an application only finishes when all tasks have been completed. The attractiveness of exploiting desktop grid systems is further reinforced by the fact that costs are highly distributed: every volunteer supports her resources (hardware, power costs and internet connections) while the benefited entity provides management infrastructures, namely network bandwidth, servers and management services, receiving in exchange a massive and otherwise unaffordable computing power. The usefulness of desktop grid computing is not limited to major high throughput public computing projects. Many institutions, ranging from academics to enterprises, hold vast number of desktop machines and could benefit from exploiting the idle cycles of their local machines [6].

The core idea of the work presented in this paper has been to provide a desktop grid computing framework and to prove its viability by testing it in some scientific computing and visualization experiments. We presented here an experiment of using QADPZ to model a real world problem: geophysical circulation modeling within the Trondheim fjord. QADPZ is an open source platform for desktop grid computing, which enables users from a local network or even Internet to share their resources. It is a multi-platform, heterogeneous system, where different computing resources from inside an organization can be used. It can also be used for volunteer computing, where the communication infrastructure is the Internet [6].

References

1. Baker, C. W. – *Scientific visualization: the new eyes of science*. Brookfield, Connecticut, Millbrook Press, 2000.
2. Bederson, B., Shneiderman, B. – *The craft of information visualization : readings and reflections*. Amsterdam, Boston, Morgan Kaufmann, 2003.
3. Bonneau, G.-P., Ertl, T., Nielson, G. M. – *Scientific visualization: the visual extraction of knowledge from data*. Berlin, Springer-Verlag, 2006.

4. Card, S. K., Mackinlay, J. D., Shneiderman, B. – *Readings in Information Visualization: using vision to think*. Morgan Kaufmann Publishers, 1999.
5. Cabral, B., Leedom, L. C. – *Imaging vector fields using line integral convolution.* Proceedings of the 20th annual conference on Computer graphics and interactive techniques, New York, ACM, 1993.
6. Constantinescu, Z. – *A Desktop Grid Computing Approach for Scientific Computing and Visualization*. PhD Thesis, Norwegian Univ. of Science and Technology, Trondheim, Norway, 2008.
7. Forssell, L. K., Cohen, S. D. – *Using line integral convolution for flow visualization: curvilinear grids, variable-speed animation, and unsteady flows.* Transactions on Visualization and Computer Graphics, No. 1, pp. 133-141, 1995.
8. Foster, I., Kesselman, C. – *The grid: blueprint for a new computing infrastructure*. Boston, Morgan Kaufmann Publishers, 2004.
9. Friedhoff, R. M., Peercy, M. S. – *Visual Computing*, Scientific American Library, New York, 2000.
10. Hansen, C. D. – Johnson, C. R. – *The visualization handbook*. Amsterdam, Boston, Elsevier-Butterworth Heinemann, 2005.
11. Heath, M. T. – *Scientific computing: an introductory survey*. Boston, McGraw-Hill, 2002.
12. Kondo, D. et al. – *Characterizing and evaluating desktop grids - An empirical study*. In: Proc. of 18th IEEE/ACM International Parallel & Distributed Processing Symposium (IPDPS2004), SanteFe, New Mexico, 2004.
13. Merzkirch, W. – *Flow visualization*. Orlando, Academic Press, 1987.
14. Mccormick, B. H. – *Visualization in scientific computing.* SIGBIO Newsletter, No. 10, pp. 15-21, 1988.
15. Mustafee, N., Taylor, S. J. E. – *Using a desktop grid to support simulation modelling*. In: Taylor, S. J. E., Ed., Proc. of 28th International Conference on Information Technology Interfaces (ITI 2006), Dubrovnik, Croatia, 2006.
16. Nielson, G. M., Hagen, H. Müller, H. – *Scientific visualization: overviews, methodologies, and techniques*. Los Alamitos, Calif., IEEE Computer Society, 1997.
17. Patrikalakis, N. M. – *Scientific visualization of physical phenomena*. Tokyo, New York, Springer-Verlag, 1991.
18. Smits, A. J. & Lim, T. T. – *Flow visualization: techniques and examples*. River Edge, NJ, Imperial College Press, 2000.
19. Tufte, E. R. – *Visual explanations images and quantities, evidence and narrative*. Cheshire, Connecticut, Graphics Press, 1997.
20. Tufte, E. R. – *The visual display of quantitative information*. Cheshire, Connecticut, Graphics Press, 2001.
21. Utnes, T. & Brors, B. – *Numerical modelling of 3-D circulation in restricted waters.* Appl. Math. Model., No. 17, pp. 522-535, 1993.

Experimente de vizualizare pe sisteme Desktop Grid

Rezumat

Cercetătorii şi programatorii dezvoltă sisteme software care implementează modele ale sistemelor studiate şi rulează aceste programe cu diverse seturi de parametri de intrare. În general, aceste modele necesită volume mari de calcule (de obicei în virgulă mobilă) şi sunt adesea executate pe supercalculatoare sau pe platforme distribuite de calcul. Vizualizarea poate ajuta depăşirea dilemei de a avea informaţii, dar nu şi interpretarea potrivită pentru acestea. Pentru a demonstra viabilitatea utilizării platformelor desktop grid în calculul ştiinţific şi în vizualizare, prezentăm aici un experiment de vizualizare care foloseşte QADPZ, o platformă desktop grid, pentru a modela o problemă din lumea reală: modelarea circulaţiei geofizice în fiordul Trondheim.