8th. World Congress on Computational Mechanics (WCCM8)
5th European Congress on Computational Methods in Applied Sciences and Engineeering (ECCOMAS 2008)
June 30 –July 5, 2008
Venice, Italy

Grid Computations of a Parallel Finite Volume Method for the Simulation of Free Surface Shallow Water Flows

A. I. Delis[1] and * E. N. Mathioudakis[2]

[1,2] Department of Sciences
Division of Mathematics
Technical University of Crete
Chania 73132, Crete, Greece
[1] adelis@science.tuc.gr
http://www.science.tuc.gr/adelis

[1,2] Applied Mathematics and
Computers Laboratory
Technical University of Crete
Chania 73132, Crete, Greece
[2] manolis@science.tuc.gr
http://www.science.tuc.gr/enm

Key Words: *Shallow Water Equations, Finite Volume, Domain Decomposition, Grid Computing.*

ABSTRACT

Computational fluids dynamics (CFD) is a fast developing field in science and engineering where the use of parallel computations is becoming increasingly essential due to the requirement of large memory size, computer code run time and other factors. In this work, we construct a parallel algorithm, suitable for distributed memory architectures, of an explicit shock-capturing finite volume method for solving the two-dimensional shallow water equations. The finite volume method [1] is based on the very popular approximate Riemann solver of Roe and is extended to second order spatial accuracy by an appropriate TVD technique [2]. The parallel code is applied to distributed memory architectures using domain decomposition techniques and we investigate its performance on a grid computer using the most common ethernet network of 100Mbps and 1Gbps speed interconnection.

Free-surface flow over a variable bottom topography under the influence of gravity can be modeled by the nonlinear shallow-water (or Saint-Venant) system of equations. Based on the conservation of mass and momentum principles the equations are given as

$$\frac{\partial \mathbf{q}}{\partial t} + \nabla \cdot \mathcal{H}(\mathbf{q}) = \mathbf{R} \quad \text{on} \quad \Omega \times [0,t] \subset \mathbb{R}^2 \times \mathbb{R}^+,$$

where $\Omega = [0,a] \times [0,b]$ and $\Omega \times [0,t]$ is the space-time domain over which solutions are sought, and the vector of conserved variables and fluxes are given by

$$\mathbf{q} = \begin{bmatrix} h \\ hu \\ hv \end{bmatrix}, \quad \mathcal{H}(\mathbf{q}) = [\mathbf{F} \quad \mathbf{G}] = \begin{bmatrix} hu & hv \\ hu^2 + \frac{1}{2}gh^2 & huv \\ huv & hv^2 + \frac{1}{2}gh^2 \end{bmatrix},$$

with $\mathbf{u} = [u,v]^{\mathrm{T}}$ being the vector velocity field, $h(x,y,t)$ the flow depth (distance from the bottom to the free surface) and g the gravitational acceleration. The source term **R** models the effects of the shape of the bottom topography on the flow. Solving the flow problem on a symmetric multiprocessor computing environment, the computations must be carried out along all available processors. Ideally, the workload should be evenly balanced and concurrency maximized so that all of the processors are kept busy doing useful work as much as possible while at the same time the communication overhead is kept at a minimum [3]. Therefore, given the serial finite volume code, its parallel framework can

be broadly written as: (1.) Divide the total computational domain into sub-domains. (2.) Assign each sub-domain as the local domain of a processor. (3.) Let each processor execute the serial code for all the computational cells lying in its local domain. (4.) Each processor will have to communicate with its adjacent processors in order to obtain the flow data required for solving the equations on its local boundary cells, before marching to the next time step. (5.) One of the processors is assigned as *master* in order, (a) to distribute and collect initial and final data and (b) make the decision of advancing or not the solution in the next time level (master-slave communication model).

Performance results for three classical benchmark problems are presented bellow (Fig. 1) across an eight processor and four node of SUN V240z grid system. The applications were developed using the Message Passing Interface standard. The Speedup investigation (Fig. 2-3) illustrates the scalability and efficiency of the resulting implementation. The TVD scheme used requires enough computation to

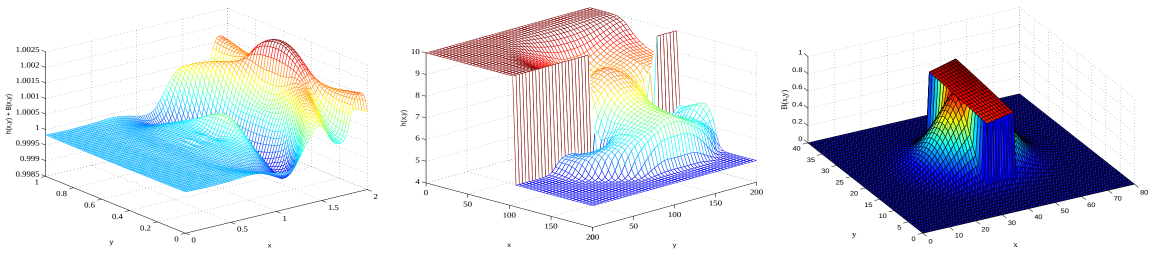

Figure 1: Water depth for Problem 1 (left) - 2 (center) and for 3 the non-smooth bed topography (right)

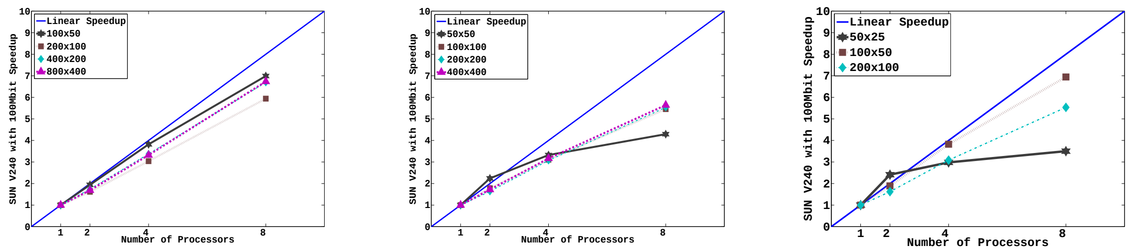

Figure 2: Grid 100Mbps Speedup measurements for Problem 1 (left) - 2 (center) and 3 (right)

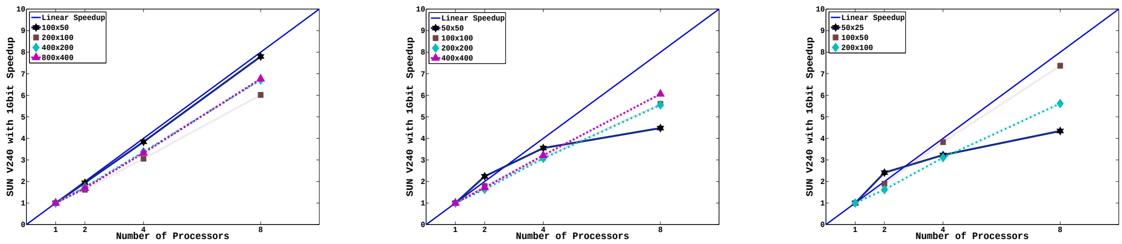

Figure 3: Grid 1Gbps Speedup measurements for Problem 1 (left) - 2 (center) and 3 (right)

provide us with a good ratio between communication and computation. This feature can be directly exploited for efficient implementations on grid computing systems. All though in these cases the network type connection affects directly the speedup performance and savings in computational time can be substantial.

REFERENCES

[1] R. J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*, Cambridge University Press , 2002.

[2] A. I. Delis and C. P. Skeels. "TVD Schemes for Open Channel Flow". *Int. J. Numer. Meth. Fluids*, Vol. **26**, 791–809, 1998.

[3] E. Mathioudakis, E. Papadopoulou and Y. Saridakis. "Iterative solution of elliptic Collocation systems on a Cognitive Parallel Computer". *Comput. & Maths with Appl.*, Vol. **48**, 951–970, 2004.