Integrated Intelligent Research (IIR)

International Journal of Computing Algorithm
Volume: 04 Issue: 02 December 2015 Pages:81-84
ISSN: 2278-2397

Developing and Porting Multi-Grid Solver for CFD Application on Intel-MIC Platform

Hitesh P. Kahale[1] , Rekha Kulkarni[1], Vikas Kumar[2]
[1] Department of Computer Engineering, PICT, Pune-411043, India
[2]CAE Group, CDAC, Pune-411007, India

Abstract— This paper presents an implementation of one dimensional Burgers equation using implicit method with Intel Xeon Phi Coprocessor. In particular, we used MAGMA MIC library which is an open source high performance library for solving a systems of non-linear equations. Further for high performance computation we consider offload mode as the primary mode of operation for Intel Xeon phi coprocessor. The result obtained from implicit scheme is then compared with the exact values and it's seen that the results obtained are approximate and reliable. The result table showed that the proposed scheme achieved higher performance on Intel MIC platform.

Keywords- High Performance Computing (HPC), implicit method, LU solver, Burgers equation, Computational Fluid Dynamics (CFD) and Intel Xeon Phi-coprocessor

I. INTRODUCTION

The Burgers equations have the wide range of applicability in industrial and non-industrial applications. It is widely used in a field of computational fluid dynamics (CFD) for studying the aerodynamics of aircraft and vehicle, hydrodynamics of ships, chemical engineering process, biomedical and marine engineering problems. It is found that Burgers equation is used to describe various kinds of problem model such as mathematical model of turbulence and the approximate theory of flow through a shock wave travelling in a viscous fluid [1]. The one dimensional viscous Burgers equation consists of a set of partial differential equations involving convection and diffusion part. For solving the Burgers equations for large range and smaller spatial and time steps, it requires large computational power and capacity. For solving such a complex problem rapidly and economically we are using Intel Xeon Phi coprocessor. The one dimensional viscous Burgers equation can be solved using different schemes like finite difference method (FDM), finite element method (FEM), finite volume method (FVM) or boundary element method (BEM).

The related study work is shown in Section 2. With the discretization of the Burgers equation it leads to the system of non-linear equations. Solving these non-linear equations is fundamental to scientific computing. For this the popular LAPACK library [2], Intel's MKL [3] and AMD's ACML [4] have been the software of choice to provide the sparse matrix solver. In this paper we have presented our approach in solving Burgers equation using MAGMA MIC library [5]. The MAGMA MIC library offers the hybridization methodology that deal with the extreme level of parallelism and heterogeneity. In hybridization methodology we will split our algorithm of interest into chunks of code and only offload the chunk of code that requires high computation to Intel Xeon Phi coprocessor. The architecture of the Intel Xeon phi coprocessors leads to accelerate the application process. However Intel Xeon Phi coprocessor-based clusters present certain challenges to achieving good communication performance [6].

II. RELATED WORK

In CFD domain the Burgers equation serves as basis for studying and describing the shock in the fluid. For approximating the one dimensional viscous Burgers equation there has been different numerical approaches. For solving these partial differential equations right selection method leads to greater convergent solution so proper numerical scheme must be chosen based on their applicability. Prior [7], Kutluay et al. have presented explicit and exact-explicit finite difference methods for solving these equations. They have discretized the diffusion equation arisen from Hopf-Cole transformation and gave an explicit expression for the exact solution of it. But they used severe stability condition which requires small size of time steps. In [8] author applied the Crank-Nicolson method to the linearized equation to obtain unconditionally stable implicit scheme for solving Burgers equation. To this scheme, they do not require any restriction over mesh sizes but the solution obtained from this scheme is not precise. The linear finite difference scheme for the initial-boundary problem of Rosenau-Burgers equation was proposed in [9]. It is flexible and efficient to apply finite difference method (FDM) to a variety of transient problems with fixed and free boundary domains [10, 11]. Its simplicity alone cannot provide the sufficient convergence rate. Using FDM we cannot easily include the topological nature of the problems and also it cannot handle higher dimensional geometric objects. In [12], K. Pandey et al. applied the Douglas finite difference to the linearized equation to obtain unconditionally stable fourth-order accuracy in space and second-order accuracy in time implicit scheme for solving Burgers equation. But this paper only gets fourth-order accuracy numerically for the diffusion equation. Authors in [13] presents the various solution methods like Fourier method, Chebyshev-Tau method, Crank-Nicolson (ABCN), ABCN collocation method and finite difference (FD) solution using a second-order approximation and a stretching transformation. But all this method is used to solve the Burgers equation with small viscosity parameter and it also leads to the geometric inflexibility of the problem. The other methods like exponential finite difference method have been used by Bhattacharya [14] and Handschuh and Keith [15].

In this paper we have used a new scheme where we first smooth the equation using LU decomposition and later applying the implicit exponential finite difference method and fully implicit exponential finite difference method [16]. For solving the sets of non-linear equations on Intel Xeon Phi coprocessor we are using MAGMA MIC library. As a solution to the high performance computing (HPC) we are using Intel

International Journal of Computing Algorithm
Volume: 04 Issue: 02 December 2015 Pages: 81-84
ISSN: 2278-2397

Xeon Phi coprocessor. The reason behind selecting this accelerator is its simplicity and reliability. It may be consider as an alternative to NVIDIA's GPU cards that require understanding of CUDA programming [17] or AMD's GPU cards that are programmed with OpenCL [18] and uses AMD's GPU libraries.

III. FORMULATION

A. *Burger's Equation*

The one dimensional viscous Burgers equation was first treated by Johannes Martinus Burgers in 1895-1981 as a mathematical model for turbulence and after whom such an equation is widely referred to as Burgers equation. It is used to model gas dynamics and traffic flow problem. The Burgers equations consist of non-linear convection term, a diffusion term and a time dependent term. Therefore it can be treated as a qualitatively correct approximation of the Navier-Stokes equations. That means it can be considered as a good model for the numerical solution of the complicated Navier-Stokes equations. Among the few non-linear partial differential equations Burgers equation can be solved exactly for an arbitrary initial condition using boundary value and condition.

The following is the one dimensional viscous Burgers equation:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial t} = v\frac{\partial^2 u}{\partial t^2} \quad (1)$$

With the initial condition:

$$u(x,0) = \emptyset(x), \quad a < x < b \quad (2)$$

And the boundary condition:

$$u(a,t) = f(t) \text{ and } u(b,t) = g(t), t > 0 \quad (3)$$

In the above partial differential Burgers equation, the equation ∂u/∂t is the time dependent equation. And the equation u.∂u/∂t and v(∂^2 u)/(∂t^2) are the non-linear convection and diffusion term. The two independent variables t and x stands for time and space coordinates. x is divided into the equal sub-intervals which is given by x_i = ih where h = Δx = (x_{max}– x_{min})/M, the spatial mesh size and i=0,1,..,M. Similarly, time t is equally divided into n intervals, i.e. t_n = nk where k = Δt = (t_{max}-t_{min})/N is the time step and n=0,1,..,N. For the present case x_{min}= 0, x_{max}= 2, ∂x=Δx, and ∂t=Δt. The dependent variables u is the velocity term and v stands for the viscosity term. If we drop the viscosity term from the above equation, i.e. v=0, then it will be a non-viscous Burgers equation.

B. *Implicit Method*

It is observed from [16] that applying the implicit exponential finite difference method (FDM) and fully implicit exponential finite difference method on Burgers equation produces a good convergent solution. Our current scheme is based on this method where the high-frequency residuals error are eliminated by the LU solver. The remaining residuals components are then projected to implicit and fully implicit FDM. Now for solving the non-linear system of equations using LU method on Intel MIC we first discretized the Burgers equation (1) using the following finite difference schemes:

Backward difference method in space direction for 1st order derivative,

$$\frac{\partial u}{\partial t} = \frac{u_i^n - u_{i-1}^n}{k} \quad (4)$$

Backward difference method in space direction for n+1 time instance

$$\frac{\partial u}{\partial x} = \frac{u_i^{n+1} - u_{i-1}^{n+1}}{h} \quad (5)$$

Central difference method in space direction for 2nd order derivative:

$$\frac{\partial^2 u}{\partial x^2} = \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{h^2} \quad (6)$$

Here we arrange the equations (4), (5) and (6) in equation (3). We obtain the coefficient matrix A, x is a vector of unknown variables and B is also a vector containing constant term. Here we solve Ax = B using MAGMA MIC library routine magma_dgesv_mic() that computes an LU factorization of a general M × N matrix A.
After smoothing the equation we apply the implicit exponential finite element difference method (I-EFDM):

$$U_i^{n+1} = U_i^n exp\left[p\left(-\frac{\Delta x U_i^n}{2v}a + b\right)\right] \quad (7)$$

And Fully Implicit exponential finite element difference method (FI-EFDM):

$$U_i^{n+1} = U_i^n exp\left[p\left(-\frac{\Delta x U_i^{n+1}}{2v}a + b\right)\right] \quad (8)$$

where,

$$p = \frac{v\Delta t}{(\Delta x)^2} \quad a = \frac{(U_{i+1}^{n+1} - U_{i-1}^{n+1})}{U_i^n}$$

$$b = \frac{(U_{i-1}^{n+1} - 2U_i^{n+1} + U_{i+1}^{n+1})}{U_i^n}$$

which is valid for values of i lying in the interval $1 \le i \le N-1$. The numerical solution obtained from this two method yields a faster and precise result as compared with the traditional explicit method.

International Journal of Computing Algorithm
Volume: 04 Issue: 02 December 2015 Pages: 81-84
ISSN: 2278-2397

```
1: Initialization
2:  nt ← number of time instance
3:  nx← number of space instance
4:  for i = 0: nt do
5:    for i = 0: nx do
6.:     magma_dsetmatrix(N,N,h_A,0,lda,d_A,0,
ldda, queue)
7:      magma_dsetmatrix(N, nrhs, h_B, 0, lda, d_B,
0, ldda, queue)
8:      magma_dgesv_mic(N, nrhs, d_A, 0, ldda, piv,
d_B, 0, ldda, &info, queue)
9:      magma_dgetmatrix(N, nrhs, d_B, 0, ldda, h_X,
0 ,lda,queue)
10:  Solve U_i^{n+1} using I-EFDM or FI-EFDM
11:  end for

12:  end for
```

C. Algorithm for solving Burgers equation on Intel Xeon Phi

The uses of accelerator device for solving the CFD problem are currently in widespread use. The algorithm introduces two variables nx and nt that are the number of space and time instances which we set at line 2 and line 3. The function magma_dsetmatrix () copies the coefficient matrix h_A and constant matrix h_B to Intel Xeon Phi coprocessor at line 6 and line 7. The control is passed to Xeon Phi where it is performing LU decomposition using magma_dgesv_mic() operation. The result from Xeon Phi is copied back to host at line 9 using magma_dgetmatrix() where we further refine and solve U_i^{n+1} using I-EFDM or FI-EFDM method. The Burgers equations are a complex set of partial differential equation that requires high computational power for obtaining solution and hence we are using accelerator device like Intel Xeon Phi. Other researchers have also focused on optimizing applications on Intel Xeon Phi coprocessor-based clusters [19].

IV. RESULT

This section presents the results obtained by our scheme on Xeon Phi coprocessor using MAGMA MIC library routine and implicit method.

A. Experimental Environment

Our experiment was performed on Intel Xeon Phi processor which is our host system and Intel Xeon Phi coprocessor which is our device. The host part consists of dual socket, 8 core Intel Xeon E5 processor with 2.6GHz frequency rate. It has 20 MB shared L3 cache in each socket and each core has 256 KB L2 and 32 KB LI and LD cache. The system is equipped with Intel Xeon Phi coprocessor via PCI Express card. It comprise of 60 Intel Architecture (IA) cores, each with four way HyperThreading, to produce the total of over 240 logical cores. Additionally, Xeon Phi has 8 GB GDDR5 RAM for faster data access, it runs at 1.05 GHz with each core having 512 KB L3 cache.

B. Experimental Result

In order to study and assess the performance of the implicit scheme in our algorithm we have compared the numerical result with that of analytical result. The test problem in our case is Problem 3 from [16]. The problem is solved for a range 2 i.e. a=0 and b=2. The given domain is equally discretized into 20 steps with h=0.1. And time step is discretized into 10 time instance with time value being 1 and k=0.1. Next the problem is solved for the given geometry with initial and boundary conditions given:

The initial condition of the problem is:

$$u(x,1) = \frac{x}{1 + exp\left[1/4v\left(x^2 - \frac{1}{4}\right)\right]} \qquad (9)$$

Where, a < x < b

And boundary condition

$$u(a,t) = u(b,t) = 0$$

With the analytical solution

$$u(x,1) = \frac{x/t}{1 + [t/\exp(1/8v)]^{1/2}\exp(x^2/4vt)} \qquad (10)$$

The result for the above problem are displayed in Table-I and in Fig.1 here for each spatial coordinates x, numerical result u_i and analytical result ua_i is calculated for values of i lying in the interval $0 \le i \le M$.

As shown from the figure we have illustrated the solution for the problem for a particular time instance t, v=1, h =0.1, k = 0.1 and $0 \le x \le 2$. The obtained numerical result is then compared with the analytical solution in the Table-I. It shows that the numerical solution obtained is precise to that of the analytical solution with our given approach.

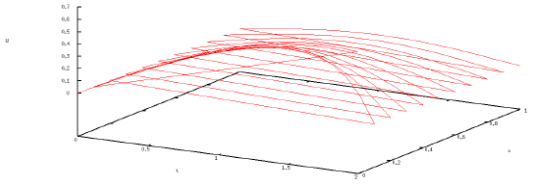

Fig. 1. Solution with Implicit method at different times instance for $a = 0, b = 2, \nu = 1, h = 0.1$ and $k = 0.1$.

V. CONCLUSION

In this paper we have shown how to solve Burgers equation using new hardware platform i.e, Intel Xeon Phi coprocessor. The challenge of porting the code effort stemmed from the fact that the new coprocessor from Intel and the integration of MAGMA MIC library that were markedly different. Further the solution of Burgers equation using MAGMA port to Xeon Phi can be considered as a good model for the numerical solution of the complicated Navier-Stokes equations. It will help to solve 3D Navier-Stokes problem where a non-linear terms grows in the equation. The ultimate goal of our research is that it will help to tremendously reduce the execution time using high-performance.

ACKNOWLEDGMENT

I am indeed thankful to my guide Prof. Rekha Kulkarni and Dr. Vikas Kumar for their able guidance and assistance to complete this paper. I am grateful for their valued support and faith on

International Journal of Computing Algorithm
Volume: 04 Issue: 02 December 2015 Pages: 81-84
ISSN: 2278-2397

me. I extend my special thanks to Srisai Meher and Samrit Kumar Maity for their helpful discussions.

TABLE I. COMPARISON OF THE NUMERICAL SOLUTION WITH THE ANALYTICAL SOLUTION FOR V= 1, H = 0.1, K= 0.1 AT SPECIFIC TIME INSTANCE
TABLE II.

At time instance t = 0		
Number of spatial steps	*Numerical solution(u_i)*	*Analytical solution(ua_i)*
0.0	0.000000	0.000000
0.1	0.035925	0.035911
0.2	0.071342	0.072298
0.3	0.105699	0.106059
0.4	0.138404	0.139540
0.5	0.168818	0.168517
0.6	0.196265	0.19725
0.7	0.220039	0.220129
0.8	0.239414	0.238504
0.9	0.253671	0.253801
1.0	0.262129	0.262125
1.1	0.264183	0.260603
1.2	0.259365	0.260301
1.3	0.247399	0.247401
1.4	0.228262	0.228280
1.5	0.202236	0.202208
1.6	0.169935	0.169938
1.7	0.132289	0.132301
1.8	0.090497	0.090503
1.9	0.045927	0.045912
2.0	0.000000	0.000000

REFERENCES

[1] A R Bahadır, Int. J. Appl. Math. 1, 897 (1999)
[2] Edward Anderson, Zhaojun Bai, Christian Bischof, Suzan L. Blackford, James W. Demmel, Jack J. Dongarra, Jeremy Du Croz, Anne Greenbaum, Sven J. Hammarling, Alan McKenney, and Danny C. Sorensen. LAPACK User's Guide. SIAM, Philadelphia, Third edition,1999.
[3] Intel. Math Kernel Library. Available at http://software. intel.com/en-us/articles/intel-mkl/.
[4] AMD. AMD Core Math Library (ACML). Available at http://developer.amd.com/tools/.
[5] Software distribution of MAGMA MIC version 1.3.1. http: //icl.cs.utk.edu/magma/software/, January 30 2015.
[6] M. Chuvelev, A. Davis, D. Durnov, S. Kazakov, A. Supalov, S. Sur, A. Tananakin, and J. Xiong. Multiple DAPL* Provider Support for Intel Xeon Phi Coprocessor with Intel MPI Library. In Proceedings of International Conference on Parallel Computing, 2013.
[7] Kutluay, S., A. R. Bahadir, and A. zde. "Numerical solution of onedimensional Burgers equation: explicit and exact-explicit finite difference methods." Journal of Computational and Applied Mathematics 103.2, 251-261.
[8] M.K. Kadalbajoo, A. Awasthi, A numerical method based on Crank Nicolson scheme for Burgers equation, Appl. Math. Comput. 182 (2006) 14301442.
[9] Li, Desheng, et al. "A finite difference simulation for Rosenau-Burgers equation." Information Engineering and Computer Science, 2009. ICIECS 2009. International Conference on. IEEE, 2009.
[10] Bell, John B., Phillip Colella, and Harland M. Glaz. "A second order projection method for the incompressible Navier-Stokes equations." Journal of Computational Physics 85.2 (1989): 257-283.
[11] Hirt, C. W., B. D. Nicholas, and N. C. Romero. "(1975), SOLAA numerical solution algorithm for transient fluid flows, Los Alamos Scientific Laboratory, Report LA-5852."
[12] K. Pandey Lajja Verma Amit K. Verma, On a finite difference scheme for Burgers equation Applied Mathematics and Computation 215 (2009) 22062214.
[13] Basdevant, Cea, et al. "Spectral and finite difference solutions of the Burgers equation." Computers and Fluids 14.1 (1986): 23-41.
[14] M C Bhattacharya, Commun. Appl. Numer. Methods 6, 173 (1990).
[15] R F Handschuh and T G Keith, Numer. Heat Transfer 22, 363 (1992)
[16] Inan, Bilge, and Ahmet Refik Bahadir. "Numerical solution of the onedimensional Burgers equation: Implicit and fully implicit exponential finite difference methods." Pramana 81.4 (2013): 547-556.
[17] VIDIA Corporation. NVIDIA CUDA C Programming Guide, February 13 2014. Retrieved on May 1, 2014 from http://docs.nvidia.com/cuda/ cuda-c-programming-guide/index.html.
[18] Khronos OpenCLWorking Group. The OpenCL specification, version: 1.0 document revision: 48, 2009.
[19] Hadri Application Experiences on a Cluster Supercomputer Equipped with Intel Xeon Phi Coprocessors.In Supercomputing Lab, 2013.