RESEARCH ARTICLE OPEN ACCESS

Parallel Multi Grid Solver for Navier Stokes Equation Using Openmpi

Mr.Shekhar Sitaram Shelke [1], Prof.Kalyani Waghmare [2]
P.G. Student [1], Associate Professor [2]
Department of Computer Science and Engineering
Pune Inst.of Comp.Technology,Pune
Maharashtra – India

ABSTRACT
Computational fluid dynamics is a computer aided study of the flow of fluid where the balanced knowledge is required from the three domain namely computer science, mathematics and fluid dynamics. Along with the knowledge of these three domains what needed is the massive computational power which was usually available with the supercomputers and high performing computational platforms but as the advancement of the computer technology is taking place continuously the new architectures and approaches coming into picture the need of the high performance computing platform is no more mandatory. The most effective way to handle the complexity is with the parallel computing approach as the hardware is available but not the way to access that hardware to its fullest.
Keywords:- *CFD*, 1D NS

I. INTRODUCTION

Computer technology is introducing itself into various branches of engineering and science due to its ability to adopt to the requirements and computing power. Among all the fields got benefits from the computer technology there is a field of fluid dynamics namely computational fluid dynamics. Computational fluid dynamics is a branch of fluid dynamics where the computer knowledge is put on usage for the study of the fluid flow with the help of the mathematics.

The flow of fluid follows some equations which are very basic to the field of the physics and fluid flow ; one such equation is Navier Stokes Equation which is a set of equations which governs the fluid flow with various properties like varying viscosity , temperature , velocity , acceleration , density etc. As to get to solve those equations two possible ways are available as analytical solution or the numerical solution.

The analytical solution of each problem is not yet possible; numerical solution is the available alternative. To solve the equation at hand using numerical method the computing power required is significantly large. High performance computers usually had such significant power but now the newer architectures used for GPU's , multicore processors and Coprocessors are powerful enough to be used along side the main HPC facilities to harness the computing power.

II. MOTIVATION

The field of computational fluid dynamics or computer aided engineering are application field of the computer technology which makes a bridge between the computer domain and engineering domain. These equation from the field of CFD(Computational Fluid Dynamics) are of varying order and partial differential equations. To solve these equations various things have to taken into consideration like discritisation,preconditioners[3],type of solvers as there is very rich mathematical background included with those equations. As there is very rich set of possibilities from which the method of descritisation,type of preconditioned and type of solver confusion is inevitable. This paper will help to understand the various methods of discretisation , type of preconditions and solvers.

A. Literature survey

In 2006 paper named "why multigrid methods are so efficient" by Irad Yavneh ,idea of multigrid is discussed briefly. The multi grid is known to be the fastest of all the numerical method to solve the elliptic boundary value problem. This paper is intended for the naïve readers who get the brief insight to the fundamentals of multigrid methods and are interested in practical use of multigrid methods [1]. In 2009 another paper named "fundamentals of multigrid" by Niko Pollner introduced multigrid methods. In this paper numerical method other than multigrid method are described briefly in order to compare these methods with the multigrid method. Some methods like gauss siedel, Jacobi are also discussed and then analyzed for the pros and cons of the cycle. Multigrid method is also discussed to its depth [2]. In 2013 paper named "Numerical solution of the one- dimensional Burgers' equation: Implicit and fully implicit exponential finite difference methods" by Bilge Inan and Ahmet Refik Bahadir two different finite difference scheme to solve a viscid burgers equation were proposed.One of them is implicit while other is explicit. In this paper space dimensions are put under the central difference scheme whereas the time is put under backward difference scheme; and thus desired accuracy is being achieved [3].

III. PROPOSED WORK

Before dealing with the proposed system directly we will get through background of the CFD, Numerical Methods.

B. CFD

As briefly depicted in the introduction Computational Fluid Dynamics is the branch of the fluid dynamics where in we solve the problem at hand by using the numerical methods rather than analytical method with the help from the algorithms and the computing power of the computer (High performance) to get the approximate solution. As its very much well understood that in CFD we are using HPC to get the approximate solution for the problem rather than the exact solution this is because the exact solution requires an analytical answer to the equation which is very tedious and complex generally and most of the time impossible to solve that's where the numerical methods kick in. Numerical methods give an approximate solution but approximate timely solution is better than exact late solution is the no solution. Numerical methods have trait that it usually gets better. and better with the iterations. On the other hand the computer can't find the analytical solution it uses the numerical method and numerical methods also fit perfect to computer domain. Thus CFD tries to find the solution of the fluid flow problems using the numerical methods as well as the computer's computing power.

As seen in section II the CFD is one solving the fluid flow problem using computer. In the fluid dynamics there are various families of fluid. Each family is different from the other by the characteristics it have. Thus these characteristics shape the behaviour of the fluid.

C. Numerical Methods

The numerical methods are the methods which made it possible to analyses and simulate equations from CFD.There are number of numerical methods and interesting thing about these equations is that these methods are having varying complexity. Among available set of all the numerical methods there is method called multigrid method which is based on the mesh approach.

Multigrid method is kind of numerical method where the whole range is broken down into fine sub ranges called as mesh and when range is abundantly divided then the solution at that level is found .This solution at fine level is then used to interpolate the solution at the higher level and thus a solution in derived. There are various approaches available to do so. Some of them are

1. V cycle [1],[2]
2. W cycle [1],[2]
3. Full multigrid Cycle [1],[2]

The main activities while multigridding are going from coarse grid to fine grid i.e. zooming in, finding the solution and going from fine grid to coarse grid i.e. zooming out. Depending on the multi gridding approach the one of the activity is performed in specific manner.

D. PROPOSED SYSTEM

In proposed system input is accepted which include the boundary conditions and initial condition. These conditions are checked for the error in first step then these input are given to the actual solver. In this system it is assumed that the user is full aware of the input given to the system and it is assumed that user is have contemplated output for the initial conditions and boundary conditions. The behaviour of the proposed system will be as follows:

1. Take input
2. Check input
3. If(input is not error free)
   Exit
4. Else
   a. Apply Initial condition to get initial value
   b. Apply implicit method iterations on those values
   c. Plot a graph.

E. Results

The graphs shown here are for inviscid fluid following 1D NS equation. The initial conditions for the fluids flow are Initial Conditions:

$$u(x,0)=x;$$
$$u(0,t)=0$$

With

Total time=1 unit
Step Size=0.1 unit
Time Step Size =0.1 unit

The obtained result statistics are as follows:

TABLE I
STATISTICS OF THE SOLUTION OF THE EQUATION WITH THE HELP OF NO PRECONDITIONER:

	Max	Average	Total
Time	1.635e-01	1.635e-01	
Objects	3.910e+02	3.910e+02	
Flops	2.605e+05	2.605e+05	2.605e+05
Flops/sec	1.593e+06	1.593e+06	1.593e+06
Memory	1.449e+05		1.449e+05
MPI Messages	0.0		0.0
MPI MessageLength	0.0	0.0	0.0
MPI Reduction	0.0		

TABLE III
STATISTICS FOR OPENMPI IMPLEMENTATION WITH MULTIGRID APPROACH WITH 1 PROCESSOR:

	Max	Average	Total
Time	6.876e-02	6.876e-02	
Objects	2.810e+02	2.810e+02	
Flops	2.294e+03	2.294e+03	2.294e+03
Flops/sec	3.336e+04	3.336e+04	3.336e+04
Memory	7.100e+05		7.100e+05
MPI Messages	0.0		0.0
MPI MessageLength	0.0	0.0	0.0
MPI Reduction	0.0		

TABLE IIIII
STATISTICS OF THE SOLUTION OF THE EQUATION WITH THE HELP OF 2 PROCESSORS:

	Max	Average	Total
Time	2.272e-01	2.272e-01	
Objects	4.110e+02	4.110e+02	
Flops	1.806e+03	1.629e+03	3.258e+03
Flops/sec	7.949e+03	7.170e+03	1.434e+04
Memory	9.343e+05		1.862e+06
MPI Messages	1.060e+02	1.060e+02	1.060e+02
MPI MessageLength	8.273e+04	7.805e+02	1.655e+05
MPI Reduction	9.450e+02		

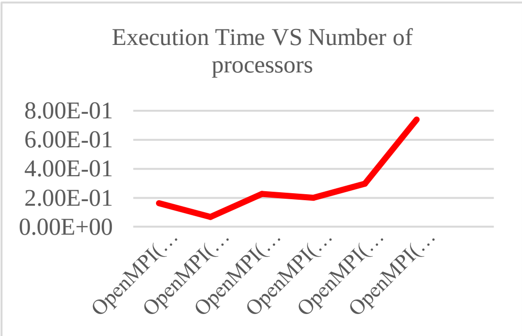
Similarly the summarized results for the serial solution with no preconditioner and Parallel OpenMPI solution with multigrid preconditioner with varying number of processors is summarized as follows:

TABLE IVV

No of Processors.	Time(Avg)	FLOPS (Avg)	Message (Avg)	Message Length (Avg)
1	1.6344e-01	2.6046e+05	0.000e+00	0.000e+00
1(MPI)	6.8718e-02	2.2940e+03	0.000e+00	0.000e+00
2(MPI)	2.2716e-01	3.2580e+03	2.120e+02	7.805e+02
3(MPI)	2.0043e-01	4.4860e+03	4.400e+02	7.533e+02
4(MPI)	2.9699e-01	5.9660e+03	6.840e+02	7.276e+02
5(MPI)	4.0221e-01	7.6740e+03	9.440e+02	7.034e+02
11(MPI)	7.3906e-01	7.2060e+03	2.400e+03	6.923e+02

Based on the table IV graph of execution time vs no of processors can be drawn as follows:

FIG. I



Thus from figure I the trend of proposed work can easily deduced that execution time is least for the parallel processes with two number of processors. Memory foot print is another vital aspect to check efficiency of the program. Memory foot prints means the effective amount of memory program looked for. Thus, by analysis of memory foot prints the programs memory needs are tracked. The parameter of memory footprint could also be included for further analysis of the solver.

IV. CONCLUSIONS

This paper includes brief idea of multigrid methods which when applied to the fluid problems from the fluid dynamics gives the far accurate results than iterative methods. The results shown here are for in viscid 1D NS equation which looks promising for the navier stoke also. Thus the same approach as used here can also be applied to the Navier stokes equations in multigrid fashion. But with accuracy timely performance is also expected which is achieved by the parallel hardware and parallel algorithms. Thus parallel multigrid navier stoke equation could also be deployed on massively parallel platforms like Intel Xeon Phi coprocessor, Nvidia Cuda and parallel technology like CUDA,OpenMP,OpenMPI.

REFERENCES

[1] Why Multigrid Methods Are So Efficient by IRAD YAVNEH 1521-9615/06/$20.00 © 2006 IEEE Copublished by the IEEE CS and the AIP

[2] Fundamentals of Multigrid Methods by Niko Pollner, FAU Erlangen-Nürnberg

[3] Multigrid Methods by Gopalkrishnan University of Florida

[4] Higher order finite element methods and multigrid solvers in a benchmark problem for the 3D Navier–Stokes equations by Volker John INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS Int. J. Numer. Meth. Fluids 2002;40:775–798 (DOI: 10.1002/d.377)

[5] Muitigrid Algorithms on the Hypercube Multiprocessor by TONY F. CHAN AND YOUCEF SAAD IEEE TRANSACTIONS ON COMPUTERS, VOL. C-35, NO. 11, NOVEMBER 1986

[6] GPU Acceleration of Algebraic Multigrid Preconditioners for Discrete Elliptic Field Problems by Christian Richter, Sebastian Schöps, and Markus Clemens IEEE TRANSACTIONS ON MAGNETICS, VOL. 50, NO. 2, FEBRUARY 2014

[7] Numerical solution of the one-dimensional Burgers' equation: Implicit and fully implicit exponential finite difference methods c Indian Academy of Sciences Vol. 81, No. 4 October 2013 pp. 547–556

[8] Discretisation, https://en.wikipedia.org/wiki/Discretization.

[9] Parallelcomputing, https://en.wikipedia.org/wiki/Parallel_computing.