Code development of finite element based MIG for flow applications

Chin-Cheng Wang

Department of Mechanical Engineering,
Yuan Ze University, Taoyuan 32003, Taiwan
ccwang@saturn.yzu.edu.tw

Abstract

Multi-scale ionized gas (MIG) flow code is mainly designed for simulation of plasma and flow interaction. MIG is based on the Galerkin and the discontinuous Galerkin finite element methods and written in FORTRAN language. The Newton-Raphson method is implemented for solving a large sparse nonlinear system at each time step. In order to improve the code performance, a fast assembly of the global stiffness matrix is employed. In addition, a single program multiple data (SPMD) model and a message passing interface (MPI) library are applied to MIG for parallel computation. Also, a parallel high performance preconditioners (Hypre) based solver is implemented. Hypre library incorporates several preconditioners to reduce the condition number of the Jacobian matrix. The MIG simulation results of compressible Euler equations and incompressible Navier-Stokes equations are verified with analytical solutions. For applications, a three-dimensional plasma actuated fluid flow is presented and validated with experimental data. Based on the performance evaluation, the reasonable speed-up and parallel efficiency are achieved using the parallel MIG flow code.

Key Words: *Discontinuous Galerkin, MPI, Hypre library.*

MIG is an acronym for multiscale ionized gas. This study describes the use of MIG flow code for the numerical solution of partial differential equations. MIG is originally developed by Subrata Roy at the Computational Plasma Dynamics Laboratory at Kettering University. MIG is a family of finite element based modules that can study macroscopic multiscale collisional plasmas. The source code is written in Fortran 77 and use Cray-style Fortran pointers for UNIX-type environment. MIG provides enormous flexibility for users. The library is organized hierarchically, enabling users to employ the level of abstraction that is most appropriate for a particular problem. MIG has an open framework and allows for general physics. Problems aren't hard coded and have been utilized to model low pressure Hall (SPT) and MPD thruster applications [1], microthrusters, microchannels and nanopore flows [2]. Also, it has been verified with 1-D, 2-D and 3-D problems including fluid dynamics and heat transfer related problems, specifically to modeling DC/RF induced dielectric barrier discharges [3], and designing electromagnetic propulsion thrusters. MIG solutions show details of the distribution of charged and neutral particles and their effects on the flow dynamics for the above applications.

Benchmark calculations have been carried out to determine the performance of the parallelized program on high performance computing clusters. The system used in this study is computer clusters running CentOS 6.5 operating system. This system consists of 32 cores at 2 GHz with a total memory of 88 GB. For the current scaling tests, we use the BiCGStab algorithm as the iterative Krylov subspace solver. Two important parameters to characterize the performance of a parallel computing are the speed-up $S(P)$ and the parallel efficiency $Eff(P)$, where P denotes number of processors. In order to verify a parallel version of the MIG flow code, the comparison has been made using a sequential code to evaluate the execution time on a single processor. The first scaling case is to show the speed-up ratio for 3-D Navier-Stokes equations of 20×20×10 elements seen in Fig. 1. This case shows the scaling as the number of processors varied from 1 to 32. The speed-up is based on the measured time of the master CPU in the first time-step and iteration. As one can see, the parallel performance is reasonably good when the number of processors is less than 16. The scaling begins to break down when the number of processors is over 16. At 32 processors, the parallel efficiency becomes much lower due to communication costs which are beginning to be important for this case. In general, the speed-up ratio can not be expected to increase linearly with growing number of processors due to the data partition communication and the parallelized algorithm. For the second case, we consider 3-D RF plasma simulation of 88×48×40 elements. Fig. 2 shows the comparison of computational time of a single processor for a sequential code and 32 processors for a parallel code. We can see the speed-up of 32 processors is over tenfold faster than that of a single processor at the first time step. At the tenth time step, the case of 32 processors is around eightfold faster than the case of a single processor. In general, a parallel version of the MIG flow code is successfully reduced the computational cost for a 3-D plasma problem.

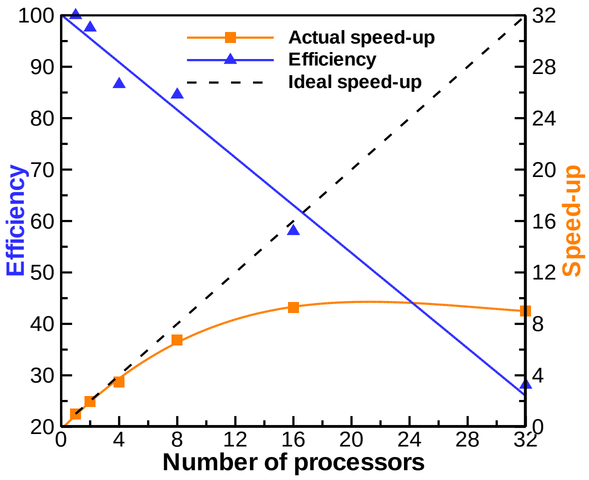

Fig. 1. Parallel efficiency and speed-up ratio against the number of processors for 3-D fluid flow problem.

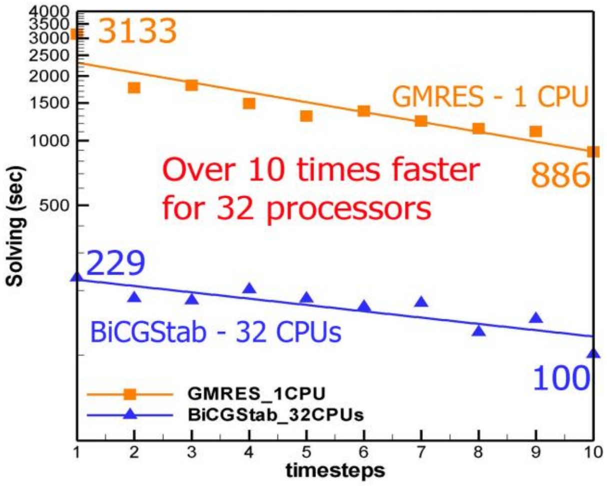

Fig. 2. Comparison of computational time with different time steps on a CPU and 32 CPUs for 3-D plasma problem.

REFERENCES

[1] Roy S and Pandey BP (2003) Development of a finite element based Hall thruster model, *J Propul Power* **19**(5):964-71.

[2] Roy S, Raju R, Chuang H, Kruden B and Meyyappan M (2003) Modeling gas flow through microchannels and nanopores, *J Appl Phys* **93**(8):4870-79.

[3] Wang CC, Durscher R and Roy S (2011) Three-dimensional effects of curved plasma actuators in quiescent air, *J Appl Phys* **109**:083305.