Efficient Parallel Implementation of a Hybrid Finite-Volume/Particle Method for the PDF Equations of Turbulent Reactive Flows

Renfeng Cao, David A. Caughey and Stephen B. Pope
Sibley School of Mechanical & Aerospace Engineering
Cornell University, Ithaca, NY 14853-750 USA

July 30, 2003

1 Introduction

Turbulent reacting flows are of great importance in modern society and occur frequently, for example, in internal combustion engines, gas turbines, furnaces etc. The computational modelling of turbulent reacting flows is widely used to aid understanding and control turbulent flames, to improve efficiency, and to reduce pollutant emissions, reduce design costs and development times. However, turbulent combustion remains one of the most complicated phenomena to describe and simulate using numerical tools.

Several types of turbulence and combustion models are used. Among these, PDF methods have the advantages of representing convection and reaction exactly without modelling assumptions, and they have been demonstrated to be an effective approach for calculating turbulent reactive flows. In PDF methods, a modelled transport equation for the joint Probability Density Function (PDF) is solved numerically by Monte Carlo methods. Monte Carlo methods are better suited than other numerical methods for this purpose since the computational effort rises only linearly with the dimensionality of the PDF[9]. The hybrid finite-volume(FV)/particle method is an attractive implementation which has less statistical and bias error than previous stand-alone particle-mesh algorithms[3, 6, 7].

Parallel computation is an effective way to reduce turnaround time for computations using the Monte Carlo/Particle methods and has been applied in several fields such as molecular simulation[2], plasma particle-in-cell codes[4], and electromagnetic particle-in-cell codes[8] etc. However, relatively few papers can be found about the parallel computation for PDF/Monte Carlo methods for turbulent combustion. Pope[10] gives an overview of previous work on the implementation of parallel computation with Monte Carlo methods. Three strategies: multiple independent simulations, particle partitioning and mesh work partitioning are described in [10]. Raju[11, 12] successfully implemented a node-based Monte Carlo method with structured/unstructured grids and parallel computing. The node-based method has several disadvantages when compared with grid-free methods[9].

In this paper, an efficient parallel implementation of the hybrid finite-volume/particle method for the PDF equations of the turbulent reactive flows is introduced. A method called domain partitioning of particles is implemented.

The code has been tested by using a non-reacting case of the vitiated coflow combustor [1]. Results about the speed-up performance in this test case are shown. Since in PDF/Monte Carlo methods the particle properties are advanced independently in time, the speed-up performance of this code for the

particle part is very close to ideal. Finally, the speed-up performance with detailed chemistry calculation is analyzed.

2 Domain partitioning of particles

The hybrid finite-volume(FV)/particle method is a combination of a finite-volume scheme and a Monte Carlo/particle method. The finite-volume scheme is used to solve the Reynolds averaged Navier-Stokes equations and the Monte Carlo/particle method is used to solve the modelled joint PDF transport equation. The particle part obtains the mean velocity and pressure from the FV code and it in turn supplies the Reynolds stresses, the scalar fluxes, and the reaction terms to the FV code[3, 6, 7].

The FV part and the particle part are relatively independent. When detailed chemistry is used in the calculation, typically more than 95% of the CPU time is used in the particle part. The reason is that it is computationally expensive to compute the change in composition due to reaction. At the same time, the nature of the particle part also makes it easier to be parallelized than the FV part. Based on these two observations, only the particle part is parallelized. The identical FV part is run on each processor while the particles are distributed onto different processors.

There are two kinds of parallel strategies available in the previous grid-free PDF code. They are multiple independent simulations and particle partitioning[10]. A new strategy named domain partitioning of particles is presented here.

We consider a PDF simulation performed in parallel using N processors. The FV grid partitions the solution domain into n_{cell} cells. These cells are partitioned into N sub-domains, each consisting of approximatively $n^*_{cell} \equiv n_{cell}/N$ cells. Each sub-domain is assigned to a different processor. There are a total of n_p particles, with each cell containing approximately $n_{pc} \equiv n_p/n_{cell}$ particles. At the beginning of each time step, all of the particles in a cell are stored on the same processor, namely the processor which is assigned to the sub-domain containing the cell. During the time step (on convection sub-steps) particles can move from cell to cell and some may move to cells in different sub-domains. At the end of the time step, message passing (using MPI) is performed to transfer particles that have moved from their initial sub-domain to the processor corresponding to their current sub-domain.

In this domain partitioning of particles, all of the particles in a given cell are stored on the same processor. Hence it is simple to implement particle interaction models (e.g., Curl's model or EMST) on the full ensemble of particles in a cell. In contrast, with particle partitioning, for a given cell, each processor has a sub-ensemble of approximately n_{pc}/N particles on which particle interaction models are implemented. This is a legitimate approach in that it converges as n_p tends to infinity, but it incurs a bias error that is N times larger than that in a serial implementation or in domain partitioning of particles. The same considerations apply to numerical operations (such as particle number and weight control) applied to the ensemble of particles in a cell.

3 Test Cases and Resuts

The domain partitioning of particles was tested by using a non-reacting vitiated coflow turbulent jet flame, which is developed by Cabra et al. The details about this flame can be found at [1]. Masri et al. [5] have performed a PDF calculation of this flame using FLUENT.

The test was performed on the Velocity Cluster at Cornell Theory Center. A series of tests was performed by changing the number of processors N (N=1, 2, 4, 8, 16) and the number of particles per cell n_{pc} (n_{pc}=24 and 48). The CPU time per processor, T, for each test is shown in Fig. 1.

Fig.1(a) shows the CPU time per processor (T) versus the number of processors (N). The CPU time T tends to an asymptote as N increase. The reason is that only the particle part is parallelized in

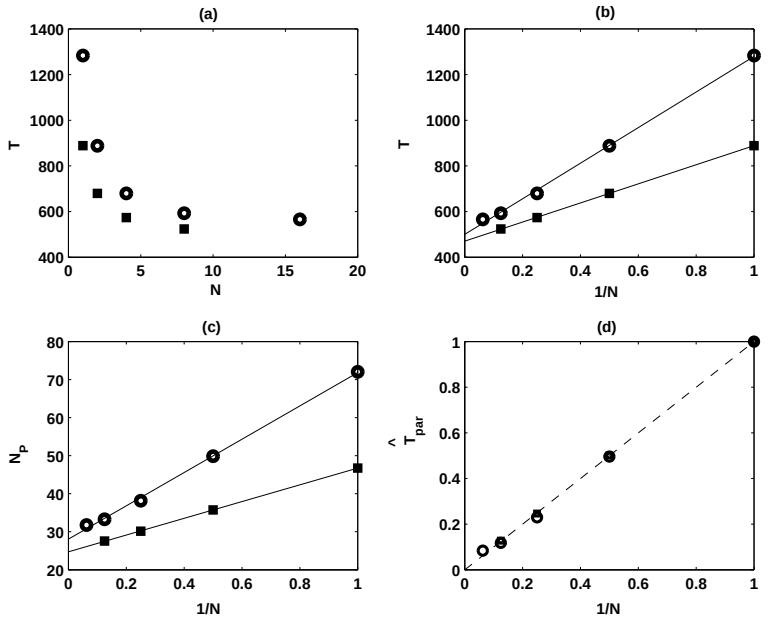

Figure 1: Speed-up performance of the domain partitioning of particles algorithm. Square: n_{pc}=24; circle: n_{pc}=48; solid line: least squares polynomial fit line; dash line: 45 degree line

the FV/particle code. Hence the asymptote corresponds to the time for the non-parallelized part of the computation.

The simplest possible model for the dependence of T on N is:

$$T = T_{non} + T_{par}/N, \tag{1}$$

where T_{non} is the time spent on non-parallelized operations, and T_{par} is the total time (summed over all processors) spent on fully parallelized operations. To investigate this model, we plot T versus $1/N$ in Fig.1(b). As may be seen, for each value of n_{pc}, the data closely conform to the straight-line behavior implied by the model. The values of T_{par} and T_{non} are thus obtained from the slopes of the lines and the intercepts with the coordinate, respectively. We thus obtain $T_{non} = 470\ min.$, $T_{par} = 418\ min.$ and $T_{non} = 490\ min.$, $T_{par} = 793\ min.$ for $n_{pc} = 24$ and $n_{pc} = 48$, respectively.

Based on these observations, we extend the model for T to include the dependence on the number of particles per cell n_{pc}. To a first approximation T_{non} is independent of n_{pc}, whereas T_{par} is linearly proportional to n_{pc}. Hence the model is extended to

$$\begin{aligned} T &= T_{non} + T_p \times n_{pc} \times n_{cell}/N \\ &= T_p \times n_{cell} \times (n_0 + n_{pc}/N), \end{aligned} \tag{2}$$

where $T_p = T_{par}/(n_{pc} \times n_{cell})$ is the CPU time per particle per processor for the parallelized work; and the non-parallel work is expressed as $T_{non} = T_p \times n_{cell} \times n_0$, where $n_0 \equiv T_{non}/T_p$ is the "equivalent number of particles" corresponding to the non-parallel work. From Fig. 1(b), we obtain $n_0 = 25$ and $n_0 = 27$ for $n_{pc} = 24$ and $n_{pc} = 48$, respectively.

Then we define $N_P \equiv n_0 + n_{pc}/N$ as the "total equivalent number of particles" which is shown in Fig.1(c).

Finally, shown in Fig. 1(d) is the normalized CPU time spent on fully parallelized operations $\hat{T}_{par}$, which is defined as $\hat{T}_{par} \equiv T_{par}/T$.

4 Conclusions

The method of domain partitioning of particles has been successfully implemented and tested in a hybrid FV/particle method for solving the PDF equations. Based on the results shown above, we arrive at the following conclusions:

1. The number of particles has a small effect on the CPU time corresponding to the non-parallel part T_{non}.
2. The CPU time corresponding to the parallel part T_{par} is mainly due to the particle calculations.
3. The speed-up performance for the particle part is very close to ideal in this test case.
4. When detailed chemistry is used, more than 95% of the CPU time is used for the particle calculations. So substantial speed-up should be achieved with the present algorithm.

Acknowledgement

This work is supported by Air Force Office of Scientific Research grant F-49620-00-1-0171. The computations were conducted using the resources of the Cornell Theory Center, which receives funding from Cornell University, New York State, federal agencies, foundations, and corporate partners.

References

[1] Cabra, R., T. Myhrvold, J.Y. Chen, R.W. Dibble, A.N. Karpetis and R.S. Barlow, "Simultaneous Laser Raman-Rayleigh-LIF Measurements and Numerical Modeling Results of a Lifted Turbulent H_2/N_2 Jet Flame in a Vitiated Coflow", *Proceedings of the Combustion Institute*, 29, 1881-1888, 2002

[2] Carvalho, A.P., J.A.N.F. Gomes, and M.N.D.D. Cordeiro, "Parallel Implementation Of A Monte Carlo Molecular Simulation Program", *J. Chem. Inf. Comput. Sci.*, 40, 588-592, 2000

[3] Jenny, P., S.B. Pope, M. Muradoglu And D.A. Caughey , " A Hybrid Algorithm For The Joint Pdf Equation Of Turbulent Reactive Flow" *J. Comp. Phys.*, 166 , 281-252, 2001

[4] Lu, Q. and D. Cai,"Implementation Of Parallel Plasma Particle-In-Cell Codes On Pc Cluster", *Comp. Phys. Commun.*, 135, 93-104, 2001

[5] Masri, A.R., R. Cao, S.B. Pope, and G.M. Goldin, "Pdf Calculations Of Turbulent Lifted Flames Of H_2/N_2 Issuing Into A Vitiated Co-Flow ", *Combust. Theory Modelling*, Submitted, 2003

[6] Muradoglu, M., P. Jenny, S.B. Pope And D.A. Caughey , "A Consistent Hybrid Finite-Volume/Particle Method For The Pdf Equations Of Turbulent Reactive Flows", *J. Comp. Phys.*, 154 , 342-371,1999

[7] Muradoglu, M., S.B. Pope And D.A. Caughey, " The Hybrid Method For The Pdf Equations Of Turbulent Reactive Flows: Consistency Conditions And Correction Algorithms", *J. Comp. Phys.*, 172 , 841-878, 2001

[8] Plimpton, S.J., D.B. Seidel, M.F. Pasik, R.S. Coats, and G.R. Montry , "A Load-Balancing Algorithm For A Parallel Electromagnetic Particle-In-Cell Code", *Comp. Phys. Commun.*, 152, 227-241,2003

[9] Pope, S.B., "Pdf Methods For Turbulent Reactive Flows", *Prog. Energy Combust. Sci.*, 11, 119-192, 1985

[10] Pope, S.B., "Pdf/Monte Carlo Methods For Turbulent Combustion And Their Implementation On Parallel Computers", *Turbulence And Molecular Processes In Combustion*, (Ed. T. Takeno), Elsevier., 51-62, 1993

[11] Raju, M.S., "Application Of Scalar Monte Carlo Probability Density Function Method For Turbulent Spray Flames", *Numer. Heat Tr. Part A*, 30, 753-777, 1996

[12] Raju, M.S., "Scalar Monte Carlo Pdf Computations Of Spray Flames Onunstructured Grids With Parallel Computing", *Numer. Heat Tr. Part B*, 35, 185-209, 1999