A parallel algorithm with improved performance of Finite Volume Method (SIMPLE-TS)

Kiril S. Shterev[1], Stefan K. Stefanov[1], and Emanouil I. Atanassov[2]

[1] Institute of Mechanics, Bulgarian Academy of Sciences,
Acad. G. Bonchev Str., Block 4, Sofia 1113, Bulgaria
{kshterev,stefanov}@imbm.bas.bg
http://www.imbm.bas.bg
[2] Institute of Information and Communication Technologies,
Bulgarian Academy of Sciences,
Acad. G. Bonchev Str., Block 2, Sofia 1113, Bulgaria
emanouil@parallel.bas.bg
http://www.iict.bas.bg

Abstract. In this paper a parallel version of the finite volume method SIMPLE-TS for calculation of two-dimensional compressible, viscous gas flows with improved performance is presented. It is demonstrated on a problem regarded to micro gas flows, taking place in Micro-Electro-Mechanical Systems (MEMS). The reorganisation of the parallel algorithm improve the algorithm performance, when more cores are used for calculations on computational grids with relatively small number of nodes or cells. The reorganisation is two-fold: first to reduce the number of communications between the processes, and second to reorder the calculation of some variables in such a way that increases the number of calculations during the communications between the processes. The comparison of speed-up between previous and new parallel versions of SIMPLE-TS was performed on two types of clusters with regard to the communication hardware: the first uses specialised cards with low latency for the interconnections between the computers and the other uses conventional cards for the interconnections. The clusters are a part of the GRID-infrastructure of the European Research Area (ERA).

Key words: finite volume method, SIMPLE-TS, gas microflows, parallel algorithms, GRID

1 Introduction

The computational analysis of fluid dynamics problems depends strongly on the computational resources [8]. The computational demands are related mainly to: the CPU performance and the memory size. In this paper we consider the problem of calculation of a two-dimensional unsteady state gaseous flow past a particle moving with supersonic speed in a planar microchannel, which is a typical example of problem requiring very big amount of computational resources. We consider a unsteady supersonic flow with Mach number equal to 2.43. The

shock wave formed in front of the particle reflects from the channel walls and interacts with the Karman vortex street (see Fig. 1) behind it. The shock wave have significant gradients of velocities, pressure and temperature. Thus, an accurate calculation of the flow requires the use of a very fine or adaptive grid. The steady state calculations have been carried out for a set of gradually refined meshes. Finally, a mesh with 8000x1600 cells was found to give stable and accurate enough results [6]. In this paper speedups of the parallel version of SIMPLE-TS published in [5] and proposed parallel version of SIMPLE-TS are compared. Hereafter the parallel version of SIMPLE-TS, published in paper [5], is referred as SIMPLE-TS [5], while the proposed in this paper is referred as SIMPLE-TS.

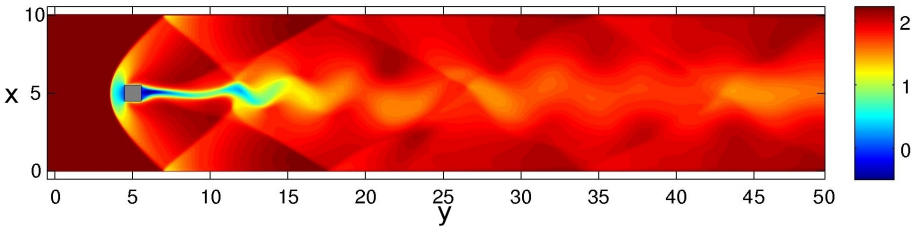

Fig. 1. Horizontal velocity field calculated by parallel version of SIMPLE-TS.

2 Continuum model equations

A two dimensional system of equations describing the unsteady flow of viscous, compressible, heat conductive fluid can be expressed in a general form as follows:

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u)}{\partial x} + \frac{\partial (\rho v)}{\partial y} = 0 \tag{1}$$

$$\begin{aligned}\frac{\partial (\rho u)}{\partial t} + \frac{\partial (\rho u u)}{\partial x} + \frac{\partial (\rho v u)}{\partial y} &= \rho g_x - A\frac{\partial p}{\partial x} + B\left[\frac{\partial}{\partial x}\left(\Gamma \frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\Gamma \frac{\partial u}{\partial y}\right)\right] \\ &+ B\left\{\frac{\partial}{\partial x}\left(\Gamma \frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\Gamma \frac{\partial v}{\partial x}\right) - \frac{2}{3}\frac{\partial}{\partial x}\left[\Gamma\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)\right]\right\}\end{aligned} \tag{2}$$

$$\begin{aligned}\frac{\partial (\rho v)}{\partial t} + \frac{\partial (\rho u v)}{\partial x} + \frac{\partial (\rho v v)}{\partial y} &= \rho g_y - A\frac{\partial p}{\partial y} + B\left[\frac{\partial}{\partial x}\left(\Gamma \frac{\partial v}{\partial x}\right) + \frac{\partial}{\partial y}\left(\Gamma \frac{\partial v}{\partial y}\right)\right] \\ &+ B\left\{\frac{\partial}{\partial y}\left(\Gamma \frac{\partial v}{\partial y}\right) + \frac{\partial}{\partial x}\left(\Gamma \frac{\partial u}{\partial y}\right) - \frac{2}{3}\frac{\partial}{\partial y}\left[\Gamma\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)\right]\right\}\end{aligned} \tag{3}$$

$$\begin{aligned}\frac{\partial (\rho T)}{\partial t} + \frac{\partial (\rho u T)}{\partial x} + \frac{\partial (\rho v T)}{\partial y} \\ = C^{T1}\left[\frac{\partial}{\partial x}\left(\Gamma^{\lambda}\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(\Gamma^{\lambda}\frac{\partial T}{\partial y}\right)\right] + C^{T2}.\Gamma.\Phi + C^{T3}\frac{Dp}{Dt}\end{aligned} \tag{4}$$

$$p = \rho T \tag{5}$$

where:

$$\Phi = 2\left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2\right] + \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right)^2 - \frac{2}{3}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)^2 \tag{6}$$

u is the horizontal component of velocity, v is the vertical component of velocity, p is pressure, T is temperature, ρ is density, t is time, x and y are coordinates of a Cartesian coordinate system. Parameters A, B, g_x, g_y, C^{T1}, C^{T2}, C^{T3} and diffusion coefficients Γ and Γ^λ, given in Eqs. (1)-(5), depend on the gas model and the equation non-dimensional form. A first order upwind scheme is used for the approximation of the convective terms, and a second order central difference scheme is employed for the approximation of the diffusion terms.

The Navier-Stokes-Fourier equations (1) - (5) are given in general form. For gaseous microflow description we use the model of a compressible, viscous hard sphere gas with diffusion coefficients determined by the first approximation of the Chapman-Enskog theory for low Knudsen numbers [7]. The Knudsen number (Kn), a nondimensional parameter, determines the degree of appropriateness of the continuum model. It is defined as the ratio of mean free path ℓ_0 to macroscopic length scale of the physical system L ($Kn = \ell_0/L$). For the calculated case the Knudsen number is equal to $Kn = 0.001$. For a hard-sphere gas, the viscosity coefficient μ and the heat conduction coefficient λ (first approximations are sufficient for our considerations) read as follows:

$$\mu = \mu_h\sqrt{T},\ \mu_h = (5/16)\rho_0\ell_0 V_{th}\sqrt{\pi} \tag{7}$$

$$\lambda = \lambda_h\sqrt{T},\ \lambda_h = (15/32)c_p\rho_0\ell_0 V_{th}\sqrt{\pi} \tag{8}$$

The Prandtl number is given by $Pr = 2/3$, $\gamma = c_p/c_v = 5/3$. The dimensionless system of equations (1) - (5) is scaled by the following reference quantities, as given in [7]: molecular thermal velocity $V_0 = V_{th} = \sqrt{2RT_0}$ for velocity, for length - square size a (Fig. 1), for time - $t_0 = a/V_0$, the reference pressure (p_0) is pressure at the inflow of the channel, the reference temperature (T_0) is equal to the channel walls, reference density (ρ_0) is calculated using equation of state (5). The corresponding non-dimensional parameters in the equation system (1) - (5) are camputed by using the following formulas:

$$\begin{aligned} &A = 0.5,\ B = \frac{5\sqrt{\pi}}{16}Kn,\ \Gamma = \Gamma^\lambda = \sqrt{T} \\ &C^{T1} = Kn\sqrt{\pi\frac{225}{1024}},\ C^{T2} = \frac{\sqrt{\pi}}{4}Kn,\ C^{T3} = \frac{2}{5} \end{aligned} \tag{9}$$

3 Parallel organization

Parallel organizations of FVM algorithms are discussed in many papers (see, e.g., [9], [4]). In this paper we present a modification of the parallel organization of

algorithm SIMPLE-TS. A conventional domain decomposition (data partitioning) approach, which details are given in [5]. The realization was accomplished by using standard MPI (Message Passing Interface) [3] instructions. There are a lot of communications in one iteration of SIMPLE-TS. To reach high parallel efficiency non-blocking communications are used and communications are overlapped with calculations.

The corresponding serial algorithm, coefficients of numerical equations and base ideas of the algorithm SIMPLE-TS are presented in details in [6]. Here is placed briefly the parallel algorithm SIMPLE-TS [5], which follows the steps of the serial algorithm SIMPLE-TS:

The SIMPLE-TS [5] algorithm
Initialize variables.
Start loop 1:

Set the initial condition for the calculated time step.
Start loop 2 (calculating a state for a new time step):
Calculate convective and diffusion fluxes.
Calculate pseudo velocities (velocities, without pressure term), coefficients for pressure equation and start send/receive messages between the processes.
Wait to complete communications between the processes.
Start loop 3:
Solve the coupled equations for energy and pressure and start send/receive messages between processes (there is exchange between data of boundary conditions and ***maximum residuals for pressure and temperature of sub-domains in order to determine maximum residuals in the computational domain***).
Wait to complete communications between processes.
Stop loop 3. In most cases two iterations are sufficient.
Calculate velocities using pseudo velocities and pressure (calculated within loop 3) and start send/receive messages between processes (here are exchanged data of boundary conditions and maximum residuals for velocities of sub-domains to determine maximum residuals in computational domain).
Compute density, using pressure and temperature calculated within loop 3.
Wait to complete communications between processes.
Convergence of loop 2: Check for convergence of the iteration process for the current time step.
Convergence of loop 1: If the final time is not reached continue.

The idea of parallel algorithm SIMPLE-TS with improved performance, is to improve the parallel scalability, when hundredths of cores are used. The number of iterations of loop 3 is fixed. This reduces the number of communications needed to determine the maximum residual for pressure and energy equations within the loops and gives the possibility to make a lot of useful calculations during the communication process. The main bottleneck of the parallel algorithm SIMPLE-TS [5] is avoided with this approach. Another new part of the presented

algorithm are the calculations of convective and diffusion fluxes at the end of the loop 2, increasing the number of calculations, which mask communications time between processes. The underlined, bold and italic text in the presented algorithms (SIMPLE-TS [5] and SIMPLE-TS) denotes the differences between them.

The SIMPLE-TS algorithm with improved performance
Initialize variables.
Start loop 1:

Set the initial condition for the calculated time step.
Start loop 2 (calculating a state for a new time step):
Calculate pseudo velocities (velocities, without pressure term), coefficients for pressure equation and start send/receive messages between processes.
Wait to complete communications between processes.
Start loop 3:
Solve the coupled equations for energy and pressure and start send/receive messages between processes (here are exchanged data of boundary conditions).
Wait to complete communications between processes.
Stop loop 3. Make fixed number of iterations.
Start send/receive maximum residuals for pressure and temperature of sub-domains in order to determine maximum residuals in the computational domain.
Calculate velocities using pseudo velocities and pressure (calculated within loop 3) and start send/receive maximum residuals for velocities of sub-domains to determine maximum residuals in computational domain.
Compute density, using pressure and temperature calculated within loop 3.
Calculate convective and diffusion fluxes.
Wait to complete communications between processes.
Convergence of loop 2: Check for convergence of the iteration process for the current time step.
Convergence of loop 1: If the final time is not reached continue.

4 Speedup analysis

The comparison of speedup of both parallel algorithms SIMPLE-TS and SIMPLE-TS [5] for calculations of gas microflows, was performed on two clusters. The first cluster uses DDR InfiniBand [1] interconnection (BG01-IPP). InfiniBand is a switched fabric communications link used in high-performance computing and enterprise data centres [2]. High throughput and low latency of InfiniBand are the most important features for achieving good speedup of the algorithm SIMPLE-TS. The second cluster, uses conventional Gigabit Ethernet cards for interconnection (BG03-NGCC). The worker nodes of both clusters contain two quad core processors. The characteristics of the clusters are shown in Table 1. The first cluster is part of the HP-SEE project infrastructure, while the latter is a GRID cluster, accessible via X509 certificate authentication.

Table 1. The characteristics of the clusters BG01-IPP and BG03-NGCC.

Cluster:	**BG01-IPP**	**BG03-NGCC**
Numbers of cores:	300 (600 Hyper-Threading enabled)	200
Numbers of cores per worker node:	8 (16 Hyper-Threading enabled)	8
CPU model:	Intel(R) Xeon(R) CPU X5560	Intel(R) Xeon(R) CPU E5430
CPU GHz:	2.80 GHz	2.66 GHz
Cache size:	8 MB	6 MB
RAM per core:	6 GB (3 Hyper-Threading enabled)	2 GB
Interconnection:	InfiniBand	Conventional cards

The speedup is calculated as $S_n = T_s/T_{par}$, where n is the number of cores (CPUs), S_n is the speedup, when n-cores are used, T_s is the time of a run on a single core, T_{par} is the time of a run performed on n-cores.

The test calculations were carried out on two meshes: 500x100 and 1000x200 cells. The final state is a periodic unstedy state flow. Determine of unsteady state regime of gas microfow for the case 500x100 cells, takes an hour on 24 cores on BG01-IPP. The same calculation on one core would needs about 23 hours Fig. 2 a).

The results concerning the speedup of both parallel algorithms are shown in

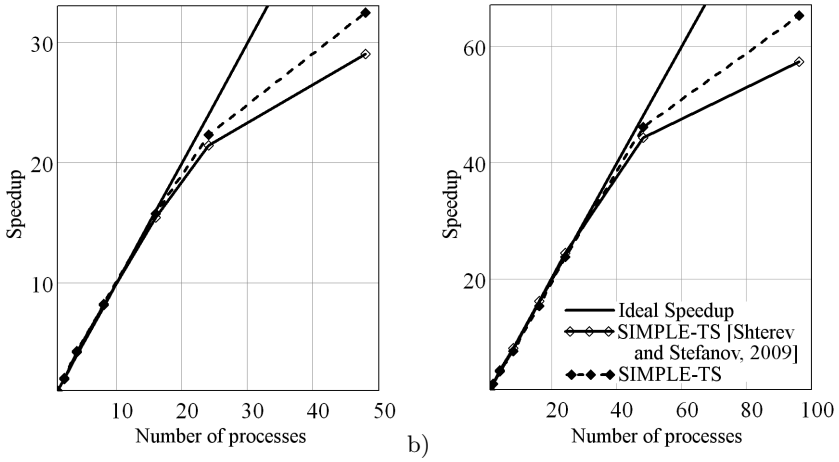

Fig. 2. The speedup of the parallel algorithm SIMPLE-TS [5] and parallel algorithm SIMPLE-TS presented in this paper on cluster BG01-IPP, for meshes: a) 500x100 cells and b) 1000x200 cells.

Fig. 2 and Fig. 3. In the proposed parallel version of SIMPLE-TS all processes are synchronised at the end of loop 2 or in other two places: at the end of loop 3 and loop 2. The parallel algorithm becomes unstable, when no synchronisation

is used. The command `MPI_Barrier()` is used for synchronisation. SIMPLE-TS [5] synchronises all processes in two places: at the end of loop 3 and loop 2. SIMPLE-TS synchronises all processes at the end of loop 3 and loop 2, when it was run on cluster BG01-IPP. A linear speedup was reached on the cluster BG01-IPP for the case with mesh 500x100 cells for less then 24 processes, Fig. 2. a). The speedup for 48 processes of SIMPLE-TS is higher then speedup of SIMPLE-TS [5]. For a finer mesh (1000x200 cells) the speedup, Fig. 2. b), is very close to ideal. Here the speedup of SIMPLE-TS is also higher then speedup of SIMPLE-TS [5] for more then 48 processes. The results show that scalability of the parallel algorithm SIMPLE-TS with improved performance is better then the algorithm SIMPLE-TS [5].

Three speedups are compared on cluster BG03-NGCC. First speedup is

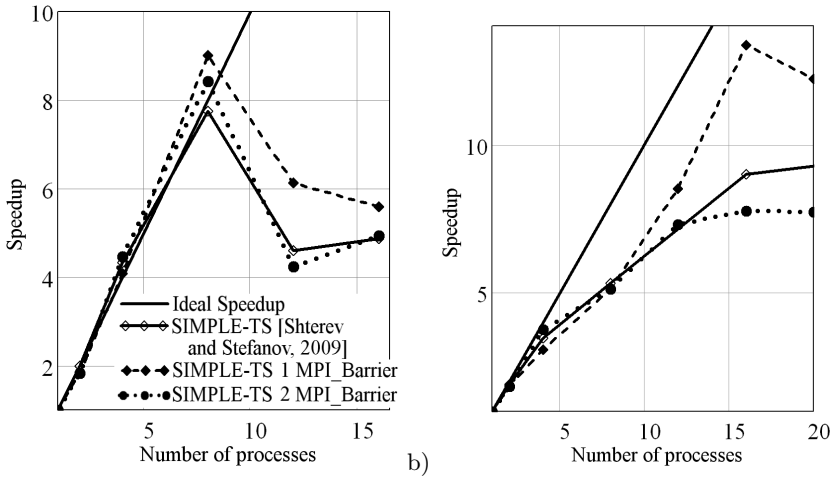

Fig. 3. The speedup of the parallel algorithm SIMPLE-TS [5] and parallel algorithm SIMPLE-TS presented in this paper with one synchronisation and two synchronisations of pal processes on cluster BG03-NGCC, for meshes: a) 500x100 cells and b) 1000x200 cells.

of SIMPLE-TS [5]. The second is of SIMPLE-TS, which make one synchronisation of all processes at the end of loop 2. The third is of SIMPLE-TS, which makes synchronisations of all processes at the end of loop 3 and loop 2. A linear speedup was reached on the cluster BG03-NGCC for the case with mesh 500x100 cells and for less then 8 processes (one worker node), Fig. 3. a), and decrease, when more then 8 processes (more then one worker node) are used. The reason for the speedup decreasing is regarded to the communications between two worker nodes. The speedup of SIMPLE-TS with one synchronisation point is better then the others. The speedups of SIMPLE-TS [5] and SIMPLE-TS with two synchronisation points are close. For a finer mesh (1000x200 cells) the speedup of SIMPLE-TS, which makes one synchronisation, Fig. 3. b), is very

good. The speedup of SIMPLE-TS [5] is better then speedup of SIMPLE-TS, with two synchronisation points.

5 Conclusions

The non-blocking communications give the possibility to reach excellent speedup. The speedup of SIMPLE-TS is improved on a cluster with InfiniBand interconnection (BG01-IPP) compared to speedup of SIMPLE-TS [5]. The speedup of SIMPLE-TS with one synchronisation point is better than the speedup reached on the cluster with conventional cards (BG03-NGCC). In other case, the algorithm SIMPLE-TS [5] shows better results, when in algorithm SIMPLE-TS are made two synchronisations of all processes. The algorithm with one synchronisation point is stable, which is important for reaching an improved parallel efficiency on cluster with conventional cards (BG03-NGCC). These results show that the good parallel organization makes the proposed algorithm efficient even, when it is run on clusters with conventional cards, used for interconnections.

Acknowledgments.
The authors appreciate the financial support by the NSF of Bulgaria under Grant (SuperCA++)- 2009 No DCVP 02/1 and the European Comission 7FP HP-SEE - 2010, grant No 261499.

References

1. InfiniBand trade association, `www.infinibandta.org`
2. Introduction and hardware charachteristics of InfiniBand, `http://en.wikipedia.org/wiki/InfiniBand`
3. The official MPI (Message Passing Interface) standards documents, errata, and archives of the mpi forum, `www.mpi-forum.org`
4. Bui, T.T.: A parallel, finite-volume algorithm for large-eddy simulation of turbulent flows. Computers & Fluids 29(8), 877–915 (2000), `http://dx.doi.org/10.1016/S0045-7930(99)00040-7`
5. Shterev, K.S., Stefanov, S.K.: A parallelization of finite volume method for calculation of gas microflows by domain decomposition methods. In: Large-Scale Scientific Computing. Large-ScaleScientific Computations, vol. 5910, pp. 523–530. Springer-Verlag Berlin Heidelberg (June 04-08 2009), `http://www.springeronline.com/978-3-642-12534-8`
6. Shterev, K.S., Stefanov, S.K.: Pressure based finite volume method for calculation of compressible viscous gas flows. Journal of Computational Physics 229(2), 461–480 (January 2010), `http://dx.doi.org/10.1016/j.jcp.2009.09.042`
7. Stefanov, S., Roussinov, V., Cercignani, C.: Rayleigh-bénard flow of a rarefied gas and its attractors. i convection regime. Physics of Fluids 14(7), 2255–2269 (2002)
8. Versteeg, H.K., Malalasekra, W.: An Introduction to Computational Fluid Dynamics: The Finite Volume Method. Prentice Hall, Pearson, 2nd edn. (2007)
9. Wang, P., Ferraro, R.D.: Parallel multigrid finite volume computation of three-dimensional thermal convection. Computers & Mathematics with Applications 37(9), 49–60 (1999), `http://dx.doi.org/10.1016/S0898-1221(99)00113-3`