Computers & Fluids 45 (2011) 241–248

Contents lists available at ScienceDirect

Computers & Fluids

journal homepage: www.elsevier.com/locate/compfluid

Development of a parallelized 2D/2D-axisymmetric Navier–Stokes equation solver for all-speed gas flows

Meng-Hua Hu [a], Jong-Shinn Wu [a,*], Yen-Sen Chen [b]

[a] *Department of Mechanical Engineering, National Chiao Tung University, Hsinchu, Taiwan*
[b] *National Space Organization, National Applied Research Laboratories, Hsinchu, Taiwan*

ARTICLE INFO

Article history:
Received 23 September 2010
Received in revised form 12 December 2010
Accepted 15 December 2010
Available online 24 December 2010

Keywords:
Parallelized
Finite-volume
Extended SIMPLE
Navier–Stokes equation
Conjugate heat transfer

ABSTRACT

A parallelized 2D/2D-axisymmetric pressure-based, extended SIMPLE finite-volume Navier–Stokes equation solver using Cartesians grids has been developed for simulating compressible, viscous, heat conductive and rarefied gas flows at all speeds with conjugate heat transfer. The discretized equations are solved by the parallel Krylov–Schwarz (KS) algorithm, in which the ILU and BiCGStab or GMRES scheme are used as the preconditioner and linear matrix equation solver, respectively. Developed code was validated by comparing previous published simulations wherever available for both low- and high-speed gas flows. Parallel performance for a typical 2D driven cavity problem is tested on the IBM-1350 at NCHC of Taiwan up to 32 processors. Future applications of this code are discussed briefly at the end.

1. Introduction

Many applications of numerical flow modeling [1–6] use density as a main dependent variable and extract the static pressure from the equation of state. However, in incompressible or low Mach number compressible flows without special treatment, such methods cannot even lead to converged solutions. Because in low compressibility limit, the density changes are very small and the pressure–density coupling becomes very weak. Some methods, such as pressure correction methods, use pressure as the primary variable [7] for solving the continuity equation, are mostly utilized for incompressible flow. However, there are several physical problems require the consideration of a compressible flow at low Mach number. Examples may include a mixed convection problem with the large buoyancy effect [8] and neutral thermal flow problem in low-temperature plasma jet [9].

There are several popular methods for solving pressure–velocity coupled flows: SIMPLE algorithm by Partankar and Spalding [7], SIMPLER by Patankar [10], SIMPLEC by Van Doormaal and Raithby [11], SIMPLEX by Van Doormaal and Raithby [12], and SIMPLEST by Sha [13]. Based on a pressure–density coupled correction scheme, the SIMPLE methods can be further extended to calculate the compressible flow problems at all speed. It is thus the major objective of this paper to present the development of a parallelized Navier–Stokes equation solver at all speeds.

The work described in this paper represents a parallel, compressible, pressure-based, collocated cell-centered finite volume method applicable at all speeds of flow, in which the primary variables are the Cartesian velocity components, pressure, and total enthalpy [14].

2. Numerical methods

2.1. Governing equations

The general form of mass conservation, Navier–Stokes equation, and energy conservation equations can be recast in the Cartesian tensor form as follows:

$$\frac{\partial(\rho\phi)}{\partial t}+\frac{\partial}{\partial x_j}(\rho V_j\phi)=\frac{\partial}{\partial x_j}(\mu_\phi\frac{\partial\phi}{\partial x_j})+S_\phi \tag{1}$$

where t is the time, x is the coordinate, V is the velocity, and the subscript j can take the value 1, 2, 3, denoting the three space coordinates. μ_ϕ is an effective diffusion coefficient, S_ϕ is the source term, ρ is the fluid density and $\phi = (1, u, v, h_t)$ stands for the variables for the mass, momentum, and energy equations, respectively. h_t $\left(h_t = CpT + \frac{V^2}{2}\right)$ is the total enthalpy, where Cp is the specific heat capacity at constant pressure and T is the temperature.

* Corresponding author. Tel.: +886 3 573 1693.
E-mail address: chongsin@faculty.nctu.edu.tw (J.-S. Wu).

0045-7930/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compfluid.2010.12.017

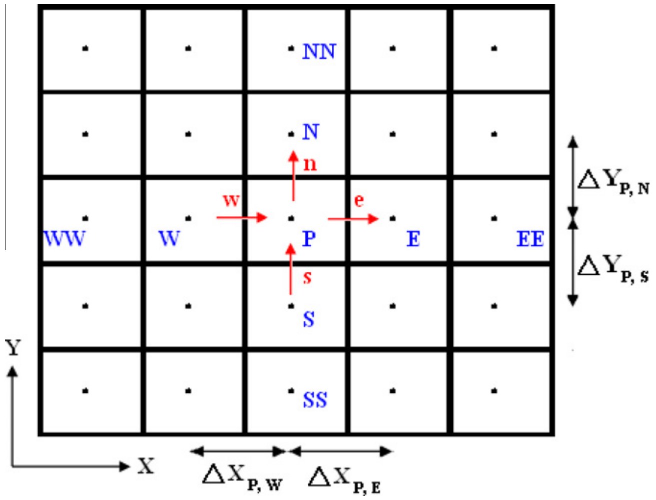

Fig. 1. Two-dimensional control volume.

2.2. Numerical scheme

2.2.1. Spatial discretization

The transport equations using the cell-centered finite-volume scheme can be written generally in integral form as

$$\frac{\partial}{\partial t}\int_{\Omega}\rho\phi\, d\Omega+\int_{\Gamma}\vec{F}\cdot\vec{n}\, d\Gamma=\int_{\Omega}S_{\phi}\, d\Omega \tag{2}$$

where Ω is the domain of interest, Γ is the surrounding surface, and $\vec{n}$ is the unit normal in outward direction. The time derivative is calculated using the first-order forward difference scheme, and the source term is treated using last time step value. The flux function $\vec{F}$ consists of the inviscid and the viscous parts:

$$\vec{F}=\rho\vec{V}\phi-\mu_{\phi}\nabla\phi \tag{3}$$

The finite volume formulation of flux integral can be evaluated by the summation of the flux vectors over each face,

$$\int_{\Gamma}\vec{F}\cdot\vec{n}\, d\Gamma=\sum_{j=k(i)}F_{i,j}\Delta\Gamma_{j} \tag{4}$$

where $k(i)$ is a list of faces of cell i, $F_{i,j}$ represents convection and diffusion fluxes through the interface between cell i and j, $\Delta\Gamma_j$ is the cell-face area. The viscous flux for the face e between control volumes P an E as shown in Fig. 1 can be approximated as:

$$\nabla\phi_{e}=\frac{\phi_{E}-\phi_{P}}{|\vec{x}_{E}-\vec{x}_{P}|}=\frac{\phi_{E}-\phi_{P}}{\Delta x_{P,E}} \tag{5}$$

2.2.2. Upwind scheme

The inviscid flux is evaluated through the values at the upwind cell and a linear reconstruction procedure to achieve second order accuracy as

$$\phi_{e}=\phi_{u}+\Psi_{e}\nabla\phi_{u}\cdot(\vec{x}_{e}-\vec{x}_{u}) \tag{6}$$

where the subscript e and u represents interface and the upwind cell, respectively, and Ψ_e is a flux limiter used to prevent from local extrema introduced by the data reconstruction. Defining $\phi_{\max}$ = max(ϕ_u, ϕ_j) and $\phi_{\min}$ = min(ϕ_u, ϕ_j), where ϕ_j is the neighbor cell of upwind cell, the Ψ_e associated with the gradient at cell u due to edge e is

$$\Psi_{e}=\begin{cases}\min\left(1,\frac{\phi_{\max}-\phi_{u}}{\phi_{e}^{0}-\phi_{u}}\right), & \text{if } \phi_{e}^{0}-\phi_{u}>0\\ \min\left(1,\frac{\phi_{\min}-\phi_{u}}{\phi_{e}^{0}-\phi_{u}}\right), & \text{if } \phi_{e}^{0}-\phi_{u}<0\\ 1, & \text{if } \phi_{e}^{0}-\phi_{u}=0\end{cases} \tag{7}$$

where ϕ_e^0 is computed without the limiting condition (i.e. $\Psi_e = 1$).

2.2.3. Pressure smoothing

The cell face velocity u_e is usually obtained by linear interpolation as

$$u_{e}=\frac{1}{2}(u_{E}+u_{P}) \tag{8}$$

To avoid the pressure oscillations due to simulation on a collocation grid, the face velocity can be modified as [15]

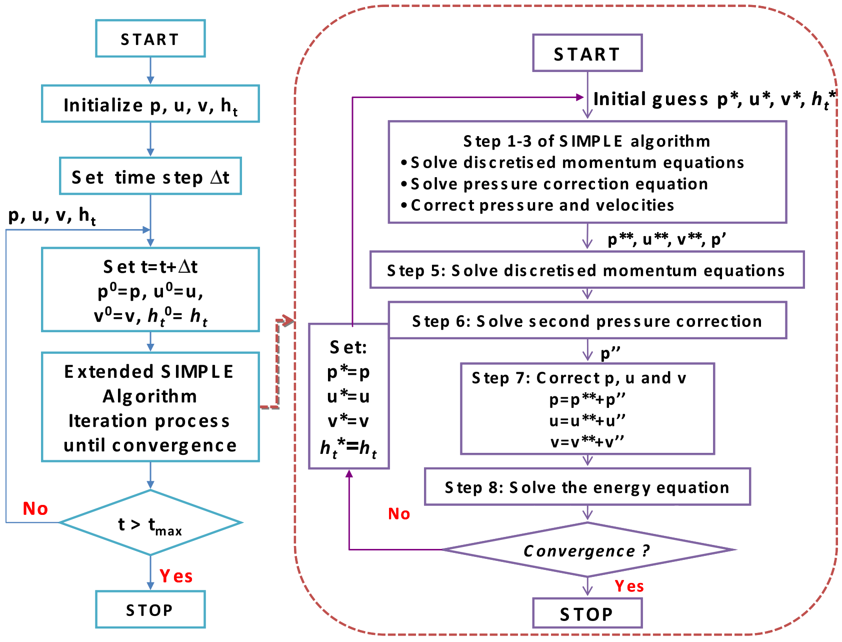
START
Initialize p, u, v, h_t
Set time step Δt
p, u, v, h_t
Set t=t+Δt
p^0=p, u^0=u,
v^0=v, h_t^0= h_t
Extended SIMPLE
Algorithm
Iteration process
until convergence
No
t > t_{max}
Yes
STOP
START
Initial guess p*, u*, v*, h_t*
Step 1-3 of SIMPLE algorithm
•Solve discretised momentum equations
•Solve pressure correction equation
•Correct pressure and velocities
p**, u**, v**, p'
Step 5: Solve discretised momentum equations
Step 6: Solve second pressure correction
p''
Step 7: Correct p, u and v
p=p**+p''
u=u**+u''
v=v**+v''
Step 8: Solve the energy equation
Set:
p*=p
u*=u
v*=v
h_t*=h_t
No
Convergence ?
Yes
STOP

Fig. 2. Flowchart of the extended SIMPLE algorithm.

$$u_e = \frac{1}{2}(u_E + u_P) + \left(\frac{\partial P}{\partial x}\frac{\delta\Omega}{A}\right)_{e1} - \left(\frac{\partial P}{\partial x}\frac{\delta\Omega}{A}\right)_{e2} \tag{9}$$

where A is the coefficient in the discretized momentum equation.

The first pressure gradient term is calculated as the mean value of cell P and E,

$$\begin{aligned}\left(\frac{\partial P}{\partial x}\frac{\delta\Omega}{A}\right)_{e1} &= \frac{1}{2}\left(\left(\frac{\partial P}{\partial x}\right)_E + \left(\frac{\partial P}{\partial x}\right)_P\right)\left(\frac{\delta\Omega}{A}\right)_e \\ &= \frac{1}{2}\left(\frac{P_{EE} - P_P}{\delta x_{P,EE}} + \frac{P_E - P_W}{\delta x_{W,E}}\right)\left(\frac{\delta\Omega}{A}\right)_e\end{aligned} \tag{10}$$

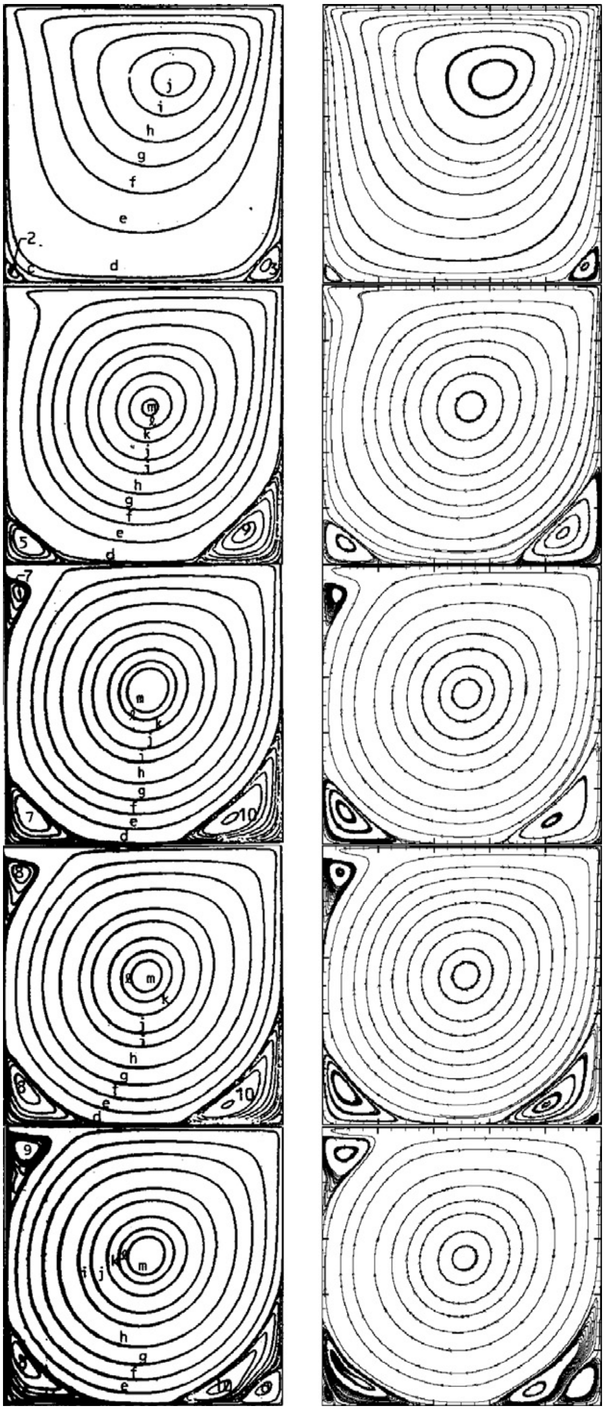

Fig. 3. Streamlines for driven cavity flow with Reynolds numbers of 100, 1000, 3200, 5000 and 10,000 (top to bottom). *Note*: Ghia et al. [19] (left); present (right).

The second one is calculated on the edge,

$$-\left(\frac{\partial P}{\partial x}\frac{\delta\Omega}{A}\right)_{e2} = -\left(\frac{P_E - P_P}{\delta x_{P,E}}\right)\left(\frac{\delta\Omega}{A}\right)_e \tag{11}$$

Set $\delta x_{P,EE} = \delta x_{W,E} = 2\delta x_{P,E}$ and

$$u_e = \frac{1}{2}(u_E + u_P) + \frac{1}{4\delta x_{P,E}}\left(\frac{\delta\Omega}{A}\right)_e [P_{EE} - 3P_E + 3P_P - P_W] \tag{12}$$

which is used to calculate the convection flux through the control volume faces. The first term is treated as a weighted average, and the second one is kept as it is to deal with non-equidistant grids.

2.2.4. Velocity-slip and temperature-jump boundary conditions

The velocity-slip boundary condition is given as:

$$v_s - v_w = \zeta \frac{\partial v}{\partial n}\bigg|_s \tag{13}$$

where v_s is the velocity of gas at the solid wall surface, v_w is the velocity of wall, $\zeta = 1.1466 \cdot Kn_{local} = 1.1466 \cdot Kn/\rho_{local}$, Kn_{local} is the local Knudsen number, ρ_{local} in the local density, and $\frac{\partial v}{\partial n}\big|_s$ is the derivative of velocity normal to the wall surface [16]. The temperature-jump is treated in a similar way:

$$T_s - T_w = \tau \frac{\partial T}{\partial n}\bigg|_s \tag{14}$$

where T_s is the temperature of gas at the solid wall surface, T_w is the temperature of wall, $\tau = 2.1904 \cdot Kn_{local} = 2.1904 \cdot Kn/\rho_{local}$, and $\frac{\partial T}{\partial n}\big|_s$ is the derivative of temperature normal to the wall surface [16].

2.2.5. Solution procedure

A general implicit discretized time-marching scheme for the transport equations is employed to solve the discretized equations. It can be written as:

$$\left(\frac{\rho^n}{\Delta t} + A_p\right)\phi_p^{n+1} = \sum A_{NB}\phi_{NB}^{n+1} + \frac{(\rho\phi_p)^n}{\Delta t} + S_\phi^n \tag{15}$$

where the superscripts n and $n + 1$ mean old value (at time t) and new value (at time $t + dt$) of the variables, respectively. The high order differencing terms and cross diffusion terms are treated using

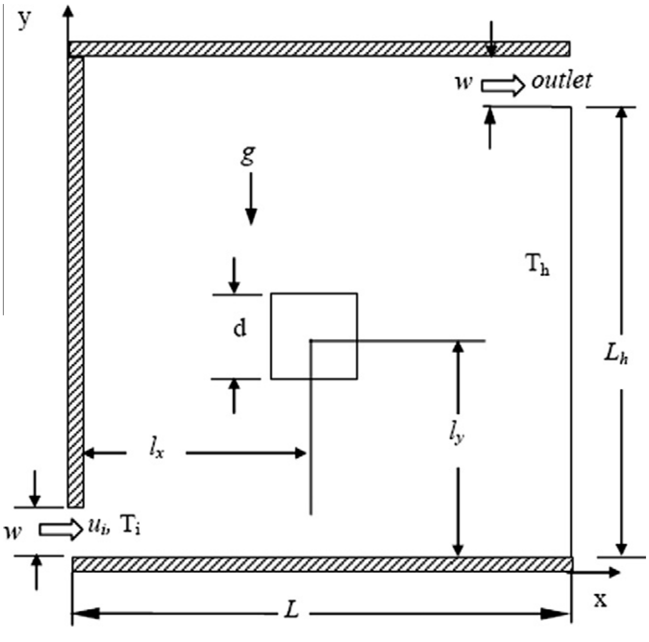

Fig. 4. Schematic of the flow in a square cavity with a square block.

known quantities and retained in the source term and updated explicitly.

In an extended SIMPLE [17,18] family pressure-correction algorithm, the pressure-correction equation for all-speed flows is formulated using the perturbed equation of state, momentum and continuity equations. The simplified formulations can be written as

$$\rho' = \frac{p'}{RT} \tag{16a}$$

$$u'_m = -D_u \nabla p' \tag{16b}$$

$$u^{k+1} = u^k + u' \tag{16c}$$

$$p^{k+1} = p^k + p' \tag{16d}$$

$$\frac{\partial \rho}{\partial t} + \nabla(u_m \rho') + \nabla(\rho u'_m) = -\nabla(\rho u_m)^k \tag{17}$$

where R is the ideal gas constant, u_m is the mth Cartesian component of the velocity, and D_u is the pressure–velocity coupling coefficient. Substituting Eq. (16) into Eq. (17), and considering $\Delta\rho = \rho^{k+1} - \rho^n = (\rho^{k+1} - \rho^k) + (\rho^k - \rho^n) = \rho' + (\rho^k - \rho^n)$, the following all-speed pressure-correction equation is obtained,

$$\frac{1}{RT}\frac{p'}{\Delta t} + \nabla\left(\frac{u_m}{RT}p'\right) - \nabla(\rho D_u \nabla p') = -\left(\frac{\rho^k - \rho^n}{\Delta t}\right) - \nabla(\rho u_m)^k \tag{18}$$

where the superscript k represents the last iterative value.

For the cell-centered scheme, the flux integration is conducted along each face and its contribution is sent to the two cells on either side of the interface. Once the integration loop is performed along the face index, the discretization of the governing equations is completed. First, the momentum equation is solved implicitly at the predictor step. Once the solution of pressure-correction equation is obtained, the velocity, pressure and density fields are updated. The predictor–corrector step is repeated two and three times so that the mass conservation is enforced. Then, the solution procedure marches to the next time level for transient calculations or global iteration for steady-state calculations. A basic description of the simulation processes is available in Fig. 2. In addition, parallel computing is implemented and tested on distributed-memory machines using spatial domain decomposition.

3. Results and discussion

3.1. Lid-driven cavity flows

The weakly compressible flow in a square lid-driven cavity is used as the benchmark problem for validation. Solutions are obtained for configurations with meshes consisting of 128 × 128 grids. The streamlines for the cavity flow with increasing Reynolds number (Re) from 100 to 10,000 along with those by Ghia et al. [19] are shown in Fig. 3. As is well known, the center of primary vortex is offset near the top right corner at Re = 100. It moves towards the geometric center of the cavity with increasing Re. It is clearly that the current NS equation solver is capable of reproducing the flow fields as Ghia et al. [19] at near-incompressible flow limit in the wide range Reynolds numbers.

3.2. Low speed flow with conjugate heat transfer

To demonstrate the capability of the solver to simulate gas flow with conjugate heat transfer, we have chosen the conjugate heat

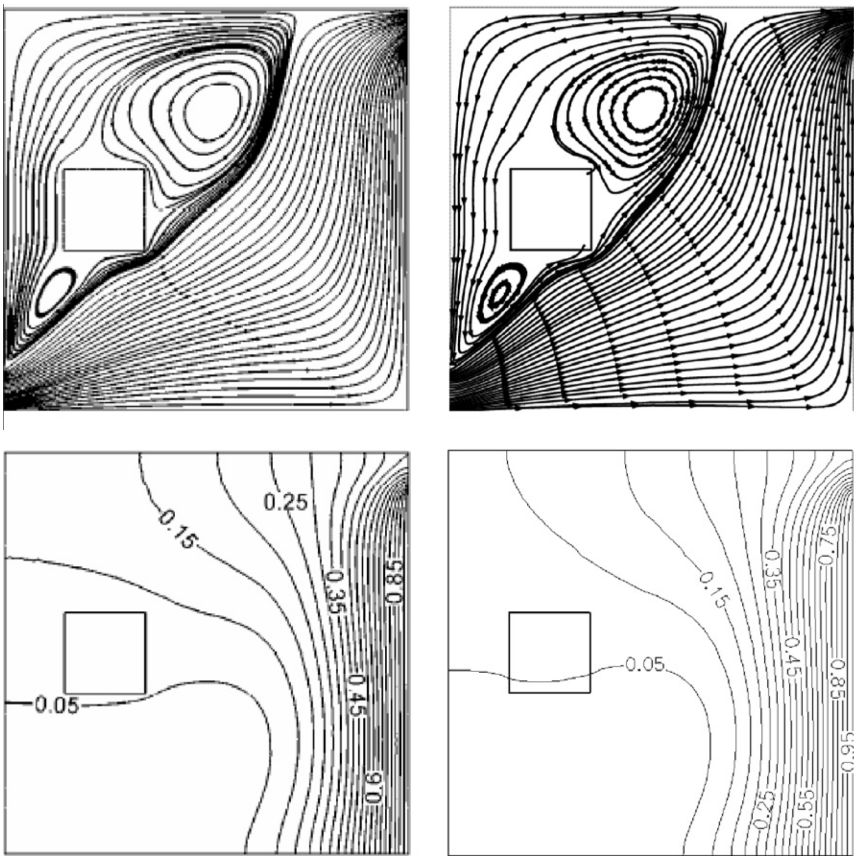

Fig. 5. Streamlines (top) and isotherms (bottom) at Ri = 1.0. *Note*: Rahman et al. [8] (left); present (right).

transfer problem simulated by Rahman et al. [8], as shown in Fig. 4. The Richardson number ($Ri = g\beta(T_h - T_i)L/u_i^2$) for this investigation is set as 0 to 5, where g is the gravitational acceleration, β is the thermal expansion coefficient, L is the length of the cavity, u_i is the inlet velocity, and T_h and T_i are the temperature of heated wall and inlet, respectively. Simulation conditions in the case are keeping $Re = 100$, inlet width $w = 1/10L$, square block width $d = 0.2L$ and solid fluid thermal conductivity ratio $K = ks/k = 5.0$, where ks and k are the heat conductivities of the solid and gas, respectively. 100 × 100 uniform grid points are used for simulations throughout the study, which is the same as Rahman et al. Conjugate heat transfer is considered by solving a steady-state heat conduction equation within the square block and by enforcing the heat flux continuity at the interfaces between gas and solid. Results of predicted streamlines and isotherms at $Ri = 1$ and $Ri = 5$ along with the data of Rahman et al. [8] are shown in Figs. 5 and 6. At smaller Ri ($Ri = 1$), the streamlines and isotherms are almost the same as those obtained by Rahman et al.; however, at higher Ri ($Ri = 5$), the flow and thermal patterns deviate greatly from those by Rahman et al. [8]. The results indicate that the Boussinesq approximation as often assumed by most of the simulations for mixed convection problems; especially at high Richardson number is highly questionable. For this type of flow, a compressible viscous NS equation solver is necessary.

3.3. Microscale high speed flow with slip boundary conditions

A 2D compressible laminar flow around a square cylinder with size a ($a = 1.4$ μm) confined in a micro-channel (height H_{ch}, length L_{ch}) is simulated to demonstrate the capability of handling supersonic flow with slip boundary conditions as shown in Fig. 7. The blockage ratio is $B = a/H_{ch} = 0.1$ and the inflow length is L_a. The reference parameters for this problem are: $P^0 = P_{in}$, $T^0 = T_{in}$, $\rho^0 = \frac{P^0}{RT^0}$, $V^0 = \sqrt{2RT^0}$, where the subscript *in* represents the inlet state. The simulation conditions in the supersonic case are Mach number $Ma = 2.4261$, Knudsen number $Kn = 0.05$, $L_{ch} = 50a$ and $L_a = 5.5a$.

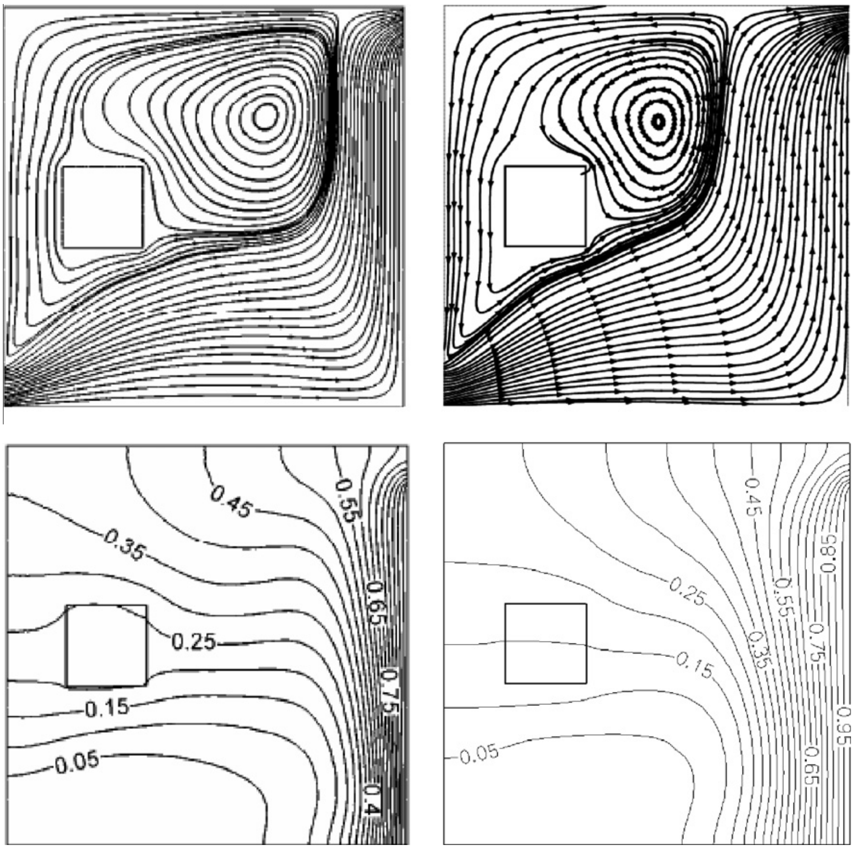

Fig. 6. Streamlines (top) and isotherms (bottom) at $Ri = 5.0$. *Note*: Rahman et al. [8] (left); present (right).

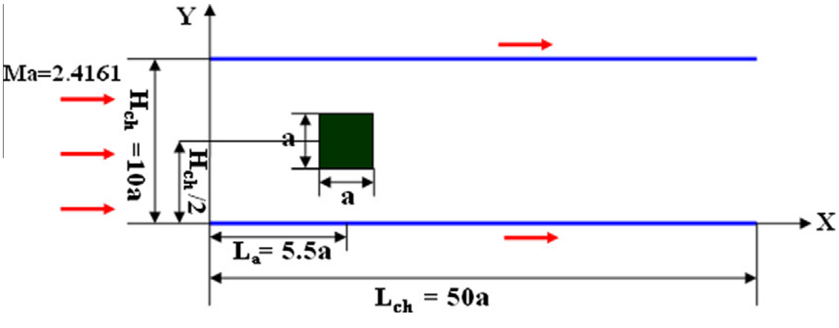

Fig. 7. Schematic of the microscale supersonic flow.

Velocity-slip and temperature-jump boundary conditions are implemented in the solver following the standard approach. 500 × 100 grid points and 16 processors are used for the simulation. The distribution of flow properties in the channel are shown in Fig. 8. Figs. 9 and 10 show typical comparisons between the present results and those by Shterev and Stefanov using SIMPLE-TS [16] using much higher resolution (1600 × 400–8000 × 1600 grids). Results show that our simulation with much lower resolution is compatible with that obtained by much higher resolution, which can reduce the simulation greatly.

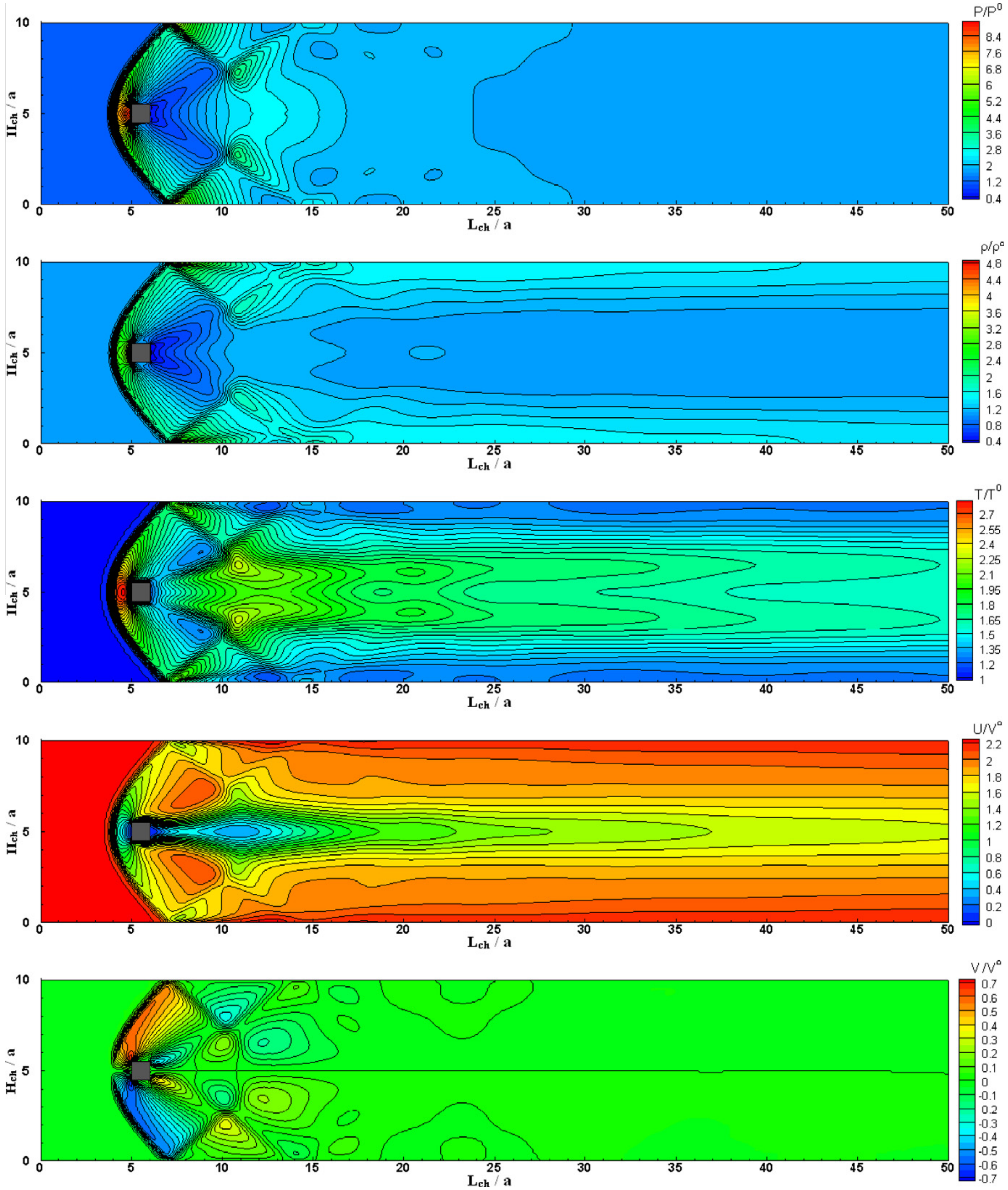

Fig. 8. Normalization distribution of flow properties in the channel at $Ma = 2.4261$ and $Kn = 0.05$.

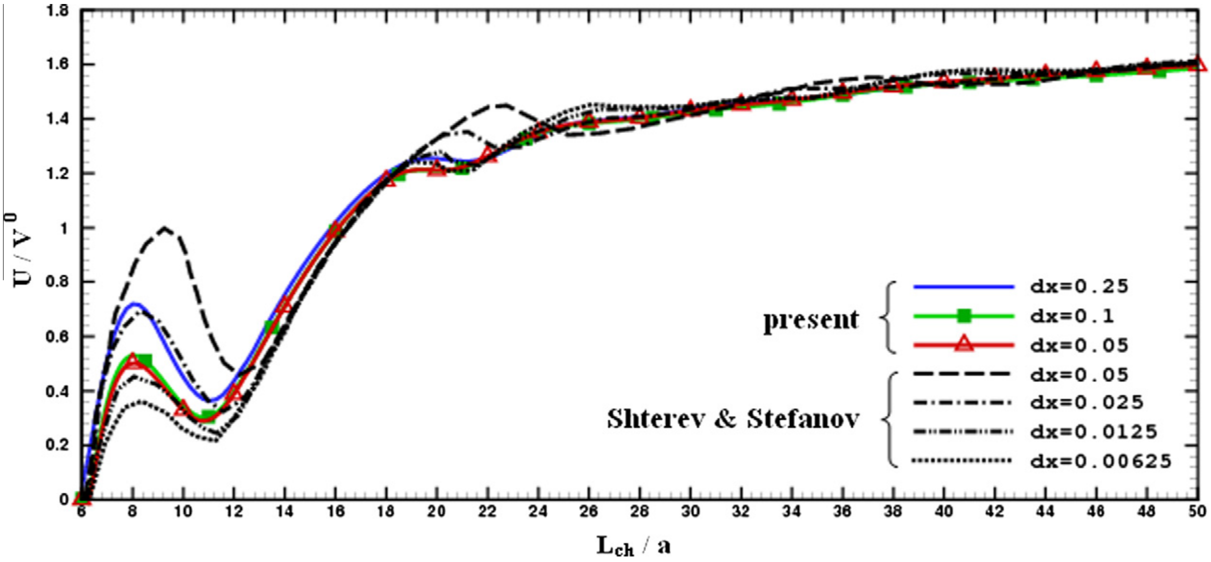

Fig. 9. Profiles of the horizontal velocity along the centre line of the channel ($y = H_{ch}/2$) for different spatial steps behind of the square.

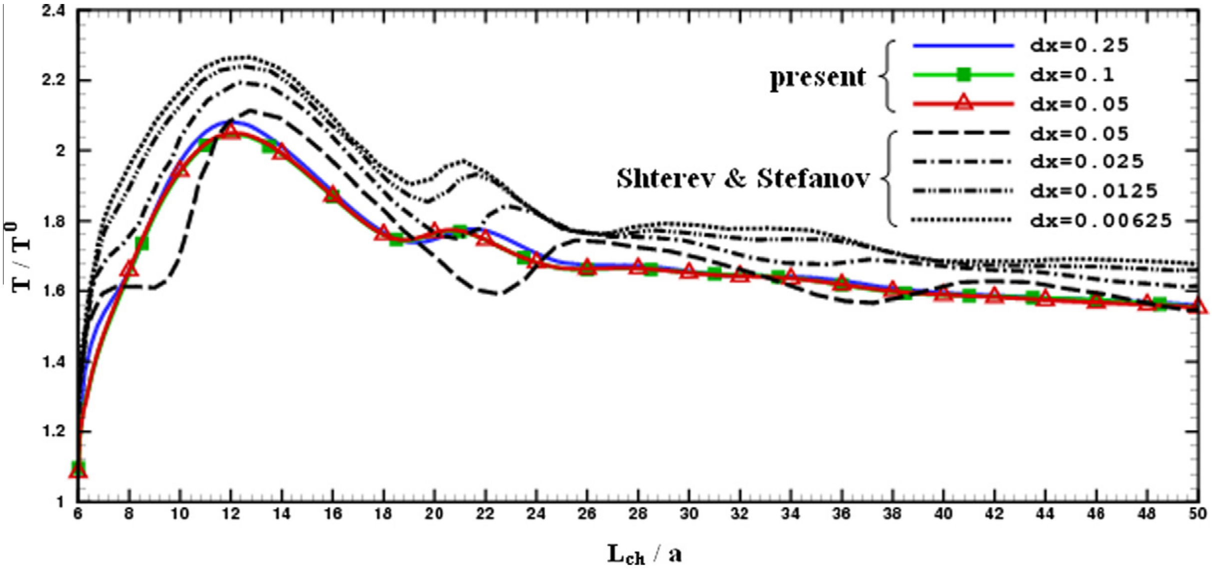

Fig. 10. Temperature profiles along the centre line of the channel ($y = H_{ch}/2$) for different spatial steps behind of the square.

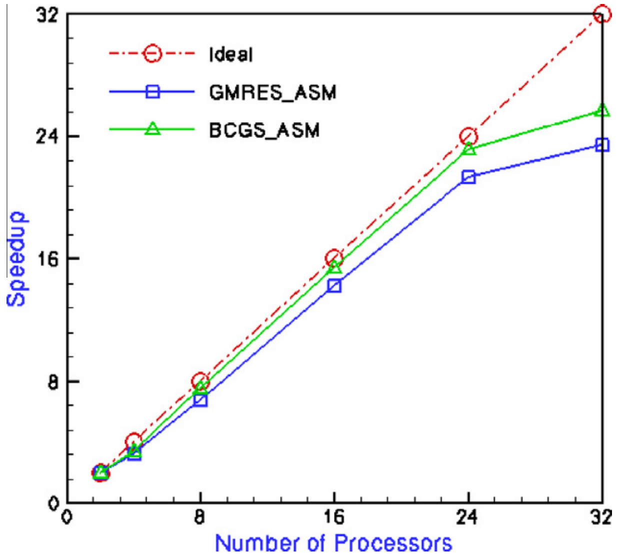

Fig. 11. Parallel performance of 2D square driven cavity with 512 × 512 grids.

3.4. Parallel performance study

The same test problem, the lid-driven cavity flow as shown earlier, is used for parallel performance study. We have used 512 × 512 grids on a PC cluster system (IBM-1350 at National Center for High-performance Computing of Taiwan) up to 32 processors using different combination of linear equation solvers (GMRES and BiCGStab). Fig. 11 shows the parallel speedups by using ILU–GMRES and ILU–BiCGStab is nearly linear up to 24 processors and begins to level off at 32 processors. We attribute this to the small grain size of the test grid size. Although the parallel speedup using ILU–BiCGStab is slightly better than using ILU–GMRES, the absolute runtime using ILU–BiCGStab is slightly longer.

4. Conclusions

In this study, the development of a parallelized 2D/2D-axisymmetric NS equation solver is presented and validated by excellent agreement with several benchmark problems, including a lid-driven cavity flow, a square block within a cavity with conjugate heat transfer and a supersonic microchannel gas flow. Results of parallel performance study shows that the developed code is scaled almost linearly up to 24 processors using 512 × 512 grid points. Combination of this parallelized NS equation solver with a parallelized fluid modeling for gas discharge is currently in progress and will be reported elsewhere in the near future.

References

[1] Jameson A, Schmidt W, Turkel E. Numerical solutions to the Euler equations by finite volume methods using Runge–Kutta time stepping. AIAA 1981:81–1259.

[2] Beam RM, Warming RF. An implicit scheme for the compressible Navier–Stokes equations. AIAA J 1978;16:393–402.

[3] MacCormack RW, Paullay AJ. Computational efficiency achieved by time splitting of finite difference operators. AIAA paper 72-154; 1972.

[4] Ni RH. A multiple grid scheme for solving the Euler equations. AIAA 1982;28:1565–71.

[5] Rizzi A, Eriksson LE. Transfinite mesh generation and damped Euler equation algorithm for transonic flow around wing-body configuration. AIAA 1981:81–0999.

[6] Sahu J, Nietubitcz CJ, Steger JL. Navier–Stokes computations of projectile base flow with and without base injection. AIAA 1985;23:1348–55.

[7] Patankar SV, Spalding DB. A calculation procedure for heat, mass, and momentum transfer in three-dimensional parabolic flows. Int J Heat Mass Transfer 1972;15:1787–806.

[8] Rahman MdM, Alim MA, Saha S, Chowdhury MK. A numerical study of mixed convection in a square cavity with a heat conducting square cylinder at different locations. J Mech Eng 2008;39:78.

[9] Chiang MH et al. Effects of oxygen addition and treating distance on surface cleaning of ITO glass by a non-equilibrium nitrogen atmospheric-pressure plasma jet. Plasma Chem Plasma Process 2010;30:553–63.

[10] Patankar SV. Numerical heat transfer and fluid flow. New York: Hemisphere; 1980.

[11] Van Doormaal JP, Raithby GD. Enhancements of the SIMPLE method for predicting incompressible fluid flows. Numer Heat Transfer 1984;7.

[12] Van Doormaal JP, Raithby GD. An evaluation of segregated approach for predicting incompressible fluid flows. Presented at national heat transfer conference, Denver, Colorado; 1985.

[13] Sha WT. Commix-1B: a three dimensional transient single-phase computer program for thermal hydraulic analysis of single and multi-component systems. Argonne national laboratory report, NASA contractor report; 1985.

[14] Chen YS. An unstructured finite volume method for viscous flow computations. In: 7th International conference on finite element methods in flow problems; February 3–7, 1989. University of Alabama in Huntsville, Huntsville, Alabama.

[15] Nilsson H, Davidson L. CALC-PVM: a parallel SIMPLEC multiblock solver for turbulent flow in complex domain. Int. rep. 98/12, Dep. of Thermo and Fluid Dynamics, Chamers University of Technology, Goteborg.

[16] Shterev KS, Stefanov SK. Pressure based finite volume method for calculation of compressible viscous gas flows. J Comput Phys 2010;229:461–80.

[17] Karki KC, Patankar SV. Pressure based calculation procedure for viscous flows at all speeds in arbitrary configurations. AIAA J 1989;27:1167–74.

[18] Shang HM, Chen YS, Liaw P, Shih MS, Wang TS. Numerical modeling of spray combustion with an unstructured grid method. AIAA paper 95-2781; 1995.

[19] Ghia U, Ghia KN, Shin TC. High resolutions for incompressible flow using the Navier–Stokes equations and a multigrid method. J Comput Phys 1982;48:387–411.