Journal of Computational Physics 228 (2009) 3182–3188

Contents lists available at ScienceDirect

Journal of Computational Physics

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

Short Note

ADI-SGS scheme on ideal magnetohydrodynamics

Hiroyuki Nishida *, Taku Nonomura

Department of Aeronautics and Astronautics, University of Tokyo, 3-1-1 Yoshinodai, Sagamihara 229 8510, Japan

ARTICLE INFO

Article history:
Received 3 October 2007
Received in revised form 21 January 2009
Accepted 28 January 2009
Available online 6 February 2009

Keywords:
Magnetohydrodynamics
Implicit scheme
ADI-SGS scheme

1. Introduction

The magnetohydrodynamics (MHD) is often used to describe many important problems in astrophysics, space propulsions, magnetic-confinement fusion and so on. For these problems, the numerical simulation is a powerful and important analytical tool.

Time-marching MHD simulations are, in particular, challenging because the MHD model contains a wide range of time scales even in the ideal MHD limit. In the ideal MHD model, there are three types of MHD waves, each with a characteristic wave speed; the fast wave, the Alfvén wave and the slow wave. Although the transit time of the MHD wave is very small in the presence of a strong magnetic field, research often focuses on the steady states of the field and phenomena with much a longer time scale. Thus, for explicit time integration, the calculations need an enormous number of time steps because the time step interval is restricted by the fastest MHD wave (the fast wave) from the CFL conditions.

There have been a number of applications of the implicit methods to MHD equations (for example, see [1–5]). Whereas standard central difference methods were used in most previous works, few have applied an approximate Riemann solver to implicit methods [1,2].

The motivation of our study is to introduce a simple and easy to code implicit scheme to MHD, using an approximate Riemann solver and a Jacobian-free technique. We adopted a fully implicit scheme called the ADI-SGS scheme [6,7], which is used for hydrodynamic equations, and simply applied this scheme to ideal one-fluid MHD equations. The ADI-SGS scheme is derived by combining alternative direction implicit (ADI) factorization [8] with the lower–upper symmetric–Gauss–Seidel (LU-SGS) method [9]. Coding of the ADI-SGS scheme is easier than the ADI scheme, and vectorization or parallelization is much easier than the LU-SGS scheme. Therefore, the ADI-SGS scheme seems to be more suitable for large scale computing.

* Corresponding author. Tel.: +81 42 759 8557; fax: +81 42 759 8458.
E-mail address: nishida@isas.jaxa.jp (H. Nishida).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.01.032

2. Formulation of the ADI-SGS scheme

The basic formulations are briefly explained. The ADI-SGS scheme can be derived by combining ADI factorization and the LU-SGS method. Detailed derivations of the formulations can be found in the references noted in this section.

The ideal MHD equations are expressed in general curvilinear coordinates (ξ,η,ζ) as follows [10]:

$$\frac{\partial \boldsymbol{Q}}{\partial t}=-\left(\frac{\partial \boldsymbol{E}}{\partial \xi}+\frac{\partial \boldsymbol{F}}{\partial \eta}+\frac{\partial \boldsymbol{G}}{\partial \zeta}\right). \tag{1}$$

Here, $\boldsymbol{Q}$ is the vector of conservative variables, and $\boldsymbol{E},\boldsymbol{F}$ and $\boldsymbol{G}$ are the flux vectors in the ξ, η and ζ directions, respectively. The detail general curvilinear coordinate transformations of the MHD equations are in Ref. [10]. In an implicit scheme, the right-hand side of Eq. (1) is evaluated at the $n+1$ time level, and Eq. (1) can be written by discretizing the time term in the first-order form as

$$\Delta \boldsymbol{Q}^n=-\Delta t\left(\frac{\partial \boldsymbol{E}}{\partial \xi}+\frac{\partial \boldsymbol{F}}{\partial \eta}+\frac{\partial \boldsymbol{G}}{\partial \zeta}\right)^{n+1}. \tag{2}$$

Here, Δt is the time step interval and $\Delta \boldsymbol{Q}^n=\boldsymbol{Q}^{n+1}-\boldsymbol{Q}^n$. Eq. (2) is rewritten using the linearization of the flux vectors proposed by Beam and Warming [8], and then is expressed in the grid point as follows:

$$\left(\boldsymbol{I}+\Delta t\frac{\partial}{\partial \xi}\boldsymbol{A}^n+\Delta t\frac{\partial}{\partial \eta}\boldsymbol{B}^n+\Delta t\frac{\partial}{\partial \zeta}\boldsymbol{C}^n\right)_{i,j,k}\Delta \boldsymbol{Q}^n_{i,j,k}=-\Delta t\left(\frac{\partial \boldsymbol{E}}{\partial \xi}+\frac{\partial \boldsymbol{F}}{\partial \eta}+\frac{\partial \boldsymbol{G}}{\partial \zeta}\right)^n_{i,j,k}. \tag{3}$$

Here, $\boldsymbol{A},\boldsymbol{B}$ and $\boldsymbol{C}$ are the flux Jacobian matrix of $\boldsymbol{E},\boldsymbol{F}$ and $\boldsymbol{G}$, respectively. In a finite-volume method, $\boldsymbol{Q}_{i,j,k}$ is the cell-averaged conserved variable vector, and the numerical fluxes on the right side of the Eq. (3) are evaluated at the cell interface by any MHD solvers. Eq. (3) may be modified specifically by the ADI factorization proposed by Beam and Warming [8] as described below:

$$\left(\boldsymbol{I}+\Delta t\frac{\partial}{\partial \xi}\boldsymbol{A}^n\right)_{i,j,k}\left(\boldsymbol{I}+\Delta t\frac{\partial}{\partial \eta}\boldsymbol{B}^n\right)_{i,j,k}\left(\boldsymbol{I}+\Delta t\frac{\partial}{\partial \zeta}\boldsymbol{C}^n\right)_{i,j,k}\Delta \boldsymbol{Q}^n_{i,j,k}=\boldsymbol{RHS}^n_{i,j,k}. \tag{4}$$

Three operators in the left-hand side of this equation are calculated in order. We adopt the LU-SGS method [9] to calculate each operator.

Diagonally dominant LDU factorization is applied to each directional operator on the left side of Eq. (4). This factorization is used in LU-ADI scheme [11] and LU-SGS scheme [9]. The operator in the ξ direction becomes

$$\left(\boldsymbol{I}-\frac{\Delta t}{\Delta \xi}\boldsymbol{A}^{n-}_{i,j,k}+\Delta t\delta^b_\xi \boldsymbol{A}^{n+}\right)\left(\boldsymbol{I}+\frac{\Delta t}{\Delta \xi}\left(\boldsymbol{A}^{n+}_{i,j,k}-\boldsymbol{A}^{n-}_{i,j,k}\right)\right)^{-1}\left(\boldsymbol{I}+\frac{\Delta t}{\Delta \xi}\boldsymbol{A}^{n+}_{i,j,k}+\Delta t\delta^f_\xi \boldsymbol{A}^{n-}\right)\Delta \tilde{\tilde{\boldsymbol{Q}}}^n_{i,j,k}=\boldsymbol{RHS}^n_{i,j,k}. \tag{5}$$

Here, $\boldsymbol{A}^+$ and $\boldsymbol{A}^-$ are the flux Jacobian matrices which have only positive and negative eigenvalues, and δ^f_ξ and δ^b_ξ are the forward and backward derivatives, respectively. $\Delta \tilde{\tilde{\boldsymbol{Q}}}^n_{i,j,k}$ is

$$\Delta \tilde{\tilde{\boldsymbol{Q}}}^n_{i,j,k}=\left(\boldsymbol{I}+\Delta t\frac{\partial}{\partial \eta}\boldsymbol{B}^n\right)_{i,j,k}\left(\boldsymbol{I}+\Delta t\frac{\partial}{\partial \zeta}\boldsymbol{C}^n\right)_{i,j,k}\Delta \boldsymbol{Q}^n_{i,j,k}. \tag{6}$$

The operators of Eq. (5), $(\boldsymbol{I}-\Delta t\boldsymbol{A}^{n-}_{i,j,k}/\Delta\xi+\Delta t\delta^b_\xi\boldsymbol{A}^{n+})$ and $(\boldsymbol{I}+\Delta t\boldsymbol{A}^{n+}_{i,j,k}/\Delta\xi+\Delta t\delta^f_\xi\boldsymbol{A}^{n-})$, lead to the lower (L) and upper (U) block-triangular matrices, respectively, and $(\boldsymbol{I}+\Delta t(\boldsymbol{A}^{n+}_{i,j,k}-\boldsymbol{A}^{n-}_{i,j,k})/\Delta\xi)^{-1}$ leads to the block-diagonal (D) matrix. Eq. (5) is calculated in two steps as follows:

1. $$\left(\boldsymbol{I}-\frac{\Delta t}{\Delta \xi}\boldsymbol{A}^{n-}_{i,j,k}+\Delta t\delta^b_\xi \boldsymbol{A}^{n+}\right)\Delta \tilde{\tilde{\boldsymbol{Q}}}^{n*}_{i,j,k}=\boldsymbol{RHS}^n_{i,j,k}, \tag{7}$$
2. $$\left(\boldsymbol{I}+\frac{\Delta t}{\Delta \xi}\boldsymbol{A}^{n+}_{i,j,k}+\Delta t\delta^f_\xi \boldsymbol{A}^{n-}\right)\Delta \tilde{\tilde{\boldsymbol{Q}}}^{n}_{i,j,k}=\left(\boldsymbol{I}+\frac{\Delta t}{\Delta \xi}\left(\boldsymbol{A}^{n+}_{i,j,k}-\boldsymbol{A}^{n-}_{i,j,k}\right)\right)\Delta \tilde{\tilde{\boldsymbol{Q}}}^{n*}_{i,j,k}. \tag{8}$$

In the LU-SGS method, $\boldsymbol{A}^\pm$ is approximated as $\boldsymbol{A}^\pm=(\boldsymbol{A}\pm\sigma_\xi)/2$, where σ_ξ is the spectral radius of $\boldsymbol{A}$. From this approximation, we can get the following discretized forms of Eqs. (7) and (8);

1. $$\left(1+\frac{\Delta t}{\Delta \xi}\sigma^n_{\xi i,j,k}\right)\Delta \tilde{\tilde{\boldsymbol{Q}}}^{n*}_{i,j,k}=\boldsymbol{RHS}^n_{i,j,k}+\frac{\Delta t}{\Delta \xi}\boldsymbol{A}^{n+}_{i-1,j,k}\Delta \tilde{\tilde{\boldsymbol{Q}}}^{n*}_{i-1,j,k}, \tag{9}$$
2. $$\left(1+\frac{\Delta t}{\Delta \xi}\sigma^n_{\xi i,j,k}\right)\Delta \tilde{\tilde{\boldsymbol{Q}}}^{n}_{i,j,k}=\left(1+\frac{\Delta t}{\Delta \xi}\sigma^n_{\xi i,j,k}\right)\Delta \tilde{\tilde{\boldsymbol{Q}}}^{n}_{i,j,k}-\frac{\Delta t}{\Delta \xi}\boldsymbol{A}^{n-}_{i+1,j,k}\Delta \tilde{\tilde{\boldsymbol{Q}}}^{n}_{i+1,j,k}. \tag{10}$$

$\Delta \tilde{\tilde{\boldsymbol{Q}}}^{n*}$ can be obtained from the forward i-sweep calculation using Eq. (9), and $\Delta \tilde{\tilde{\boldsymbol{Q}}}^{n}$ can be obtained from the backward i-sweep calculation using Eq. (10). The operator in the η and ζ directions of Eq. (4) can be calculated using the same procedures, and finally we obtain $\Delta \boldsymbol{Q}^n$.

On the right side of Eqs. (9) and (10), the Jacobian matrices have to be evaluated at each cell and matrix operations are required. For the MHD equations, analytical evaluation and operation of these matrices are extremely complex procedures [10]. However, these operations can be eliminated by following approximations [12]:

$$\boldsymbol{A}^{n+}_{i,j,k}\Delta\tilde{\tilde{\boldsymbol{Q}}}^{n*}_{i,j,k}=\frac{1}{2}\left(\boldsymbol{A}^{n}_{i,j,k}+\sigma^{n}_{\tilde{\xi}i,j,k}\right)\Delta\tilde{\tilde{\boldsymbol{Q}}}^{n*}_{i,j,k}=\frac{1}{2}\Delta\tilde{\tilde{\boldsymbol{E}}}^{n*}_{i,j,k}+\frac{1}{2}\sigma^{n}_{\tilde{\xi}i,j,k}\Delta\tilde{\tilde{\boldsymbol{Q}}}^{n*}_{i,j,k}, \tag{11}$$

$$\boldsymbol{A}^{n-}_{i,j,k}\Delta\tilde{\tilde{\boldsymbol{Q}}}^{n}_{i,j,k}=\frac{1}{2}\left(\boldsymbol{A}^{n}_{i,j,k}-\sigma^{n}_{\tilde{\xi}i,j,k}\right)\Delta\tilde{\tilde{\boldsymbol{Q}}}^{n}_{i,j,k}=\frac{1}{2}\Delta\tilde{\tilde{\boldsymbol{E}}}^{n}_{i,j,k}-\frac{1}{2}\sigma^{n}_{\tilde{\xi}i,j,k}\Delta\tilde{\tilde{\boldsymbol{Q}}}^{n}_{i,j,k}, \tag{12}$$

where

$$\Delta\tilde{\tilde{\boldsymbol{E}}}^{n*}_{i,j,k}=\boldsymbol{E}\left(\boldsymbol{Q}^{n}_{i,j,k}+\Delta\tilde{\tilde{\boldsymbol{Q}}}^{n*}_{i,j,k}\right)-\boldsymbol{E}\left(\boldsymbol{Q}^{n}_{i,j,k}\right), \tag{13}$$

$$\Delta\tilde{\tilde{\boldsymbol{E}}}^{n}_{i,j,k}=\boldsymbol{E}\left(\boldsymbol{Q}^{n}_{i,j,k}+\Delta\tilde{\tilde{\boldsymbol{Q}}}^{n}_{i,j,k}\right)-\boldsymbol{E}\left(\boldsymbol{Q}^{n}_{i,j,k}\right). \tag{14}$$

This scheme is an efficient scheme for obtaining a steady state solution. The approximation "$\boldsymbol{A}^{\pm}=(\boldsymbol{A}\pm\sigma_{\xi})/2$" introduces large numerical dissipation and hence this scheme cannot be straightforwardly applied to unsteady problems [9]. However, we can make this method achieve higher-order time accuracy by applying the Newton–Raphson iterative algorithm [13]. The advantages of this scheme are that the vectorization and parallelization are quite easy, because the ξ, η and ζ directional operators can be separately calculated.

The ADI-SGS scheme turns into the LU-ADI scheme for linear problems, and therefore, the stability of the former for the linear hyperbolic system can be discussed as being the same as the latter. The stability of the LU-ADI scheme is similar to the standard ADI scheme [14], and linear stability analysis was conducted by Warming and Beam [15] for a standard ADI scheme applied to linear partial differential equations. Their analysis showed unconditional stability in two dimensions, but unconditional instability in three dimensions. However, it was pointed out that the instability in three dimensions was weak and could be controlled by numerical dissipation [14,16]. In actual practice with nonlinear problems, stability bounds exist, although they are much less strict than the explicit stability bounds, and therefore, the actual stability bounds in a particular problem have to be determined by numerical experimentation [16].

3. Numerical tests

To verify the effectiveness of the ADI-SGS scheme, a two-dimensional stationary MHD flow with a shock is simulated by the ADI-SGS scheme with a finite-volume approximate Riemann solver. We adopt the TVD Lax–Friedrich scheme [17], which

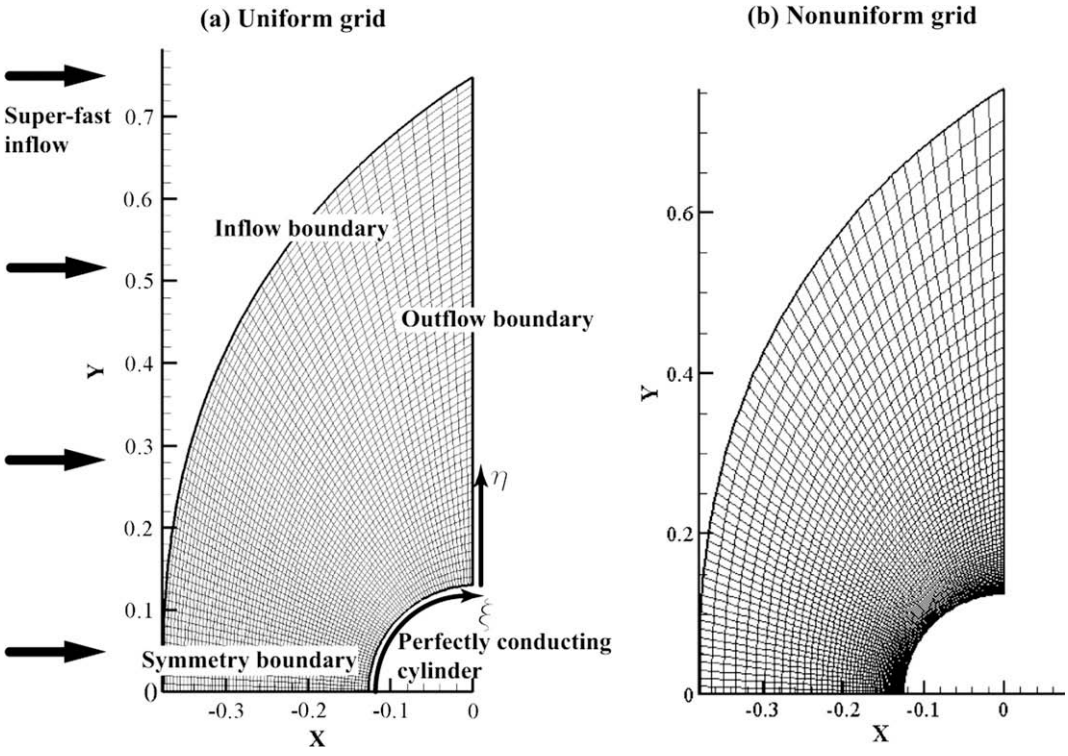

Fig. 1. Computational settings and computational grid systems.

is one of the most simple Riemann solver, and the MUSCL method with the MINMOD limiter is used to obtain 2nd-order spatial accuracy. To treat the errors in $\nabla \cdot \boldsymbol{B}$, the Powell's source terms and the projection scheme are used [18]. Note that the Powell's source terms are evaluated explicitly for simplicity.

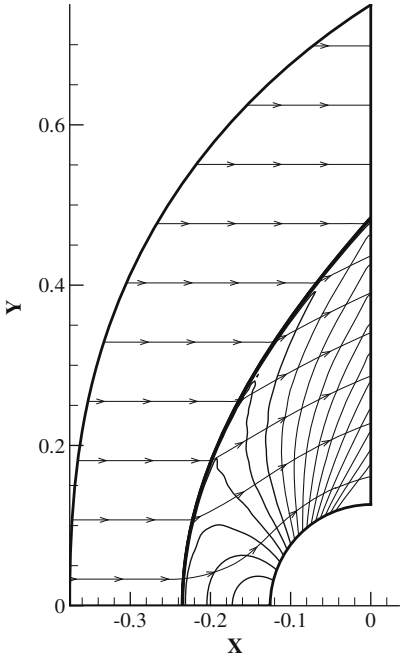

Fig. 2. Simulation results: the density contours and the streamlines.

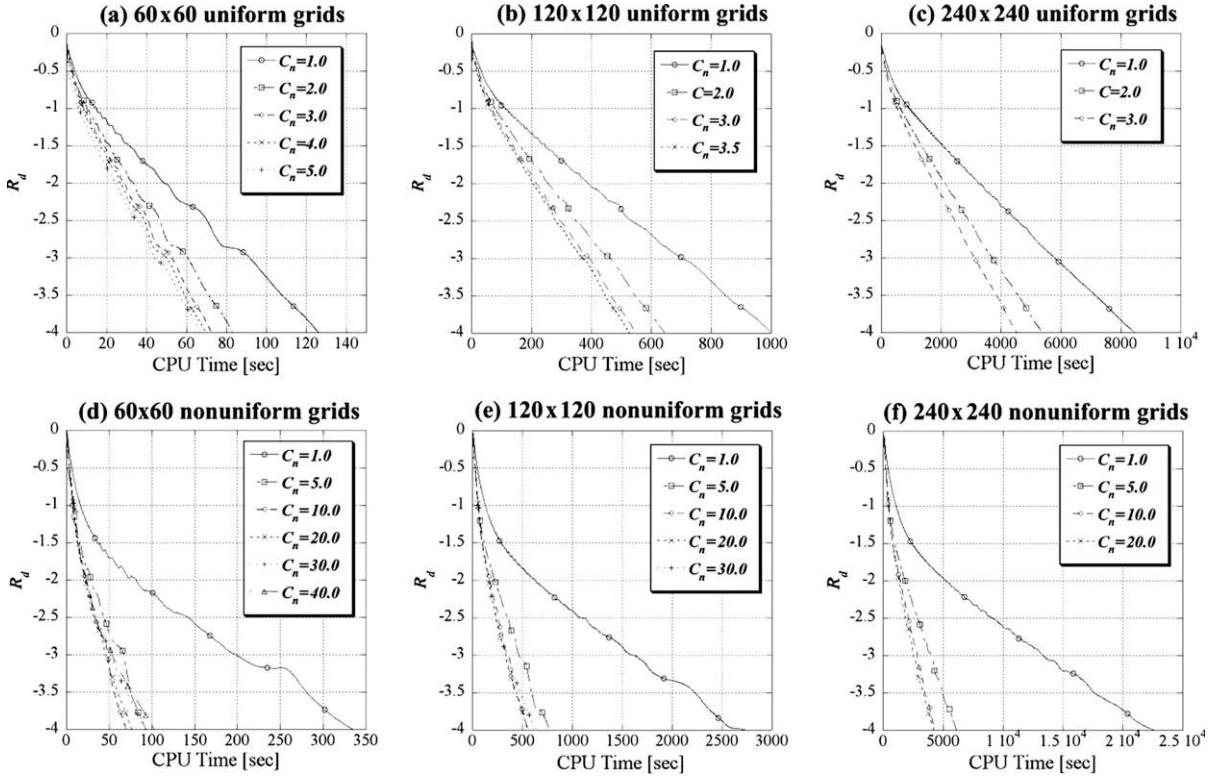

Fig. 3. Convergence history of the density residual toward a steady state.

The numerical test problem is a stationary bow shock flow, in which the bow shock is formed by the obstruction of a uniform super-fast incoming flow by a rigid perfectly conducting circular cylinder [19]. Fig. 1(a) shows the computational settings. Boundary conditions are implemented by using ghost cells. A uniform super-fast inflow with magnetic fields parallel to the flow is imposed at the inflow boundary with the density $\rho = 1$, the pressure $p = 0.2$, the flow velocity $v_x = 2$ and the magnetic flux density $B_x = 0.1$. The radius of the perfectly conducting cylinder is 0.125, and at the cylinder wall boundary and $y = 0$ boundary, the ideal symmetry condition is implemented. The free outflow condition is implemented at the $x = 0$ boundary, that is the state variables of the ghost cells are extrapolated linearly with the last two physical cells. Initially, the physical state in the computational domain is uniform with $\rho = 1, p = 0.2, v_x = 2$ and $B_x = 0.1$.

The grid spacing in the ξ direction is uniform, but both the uniform and nonuniform grid spacing are used in the η direction as shown in Fig. 1. In the nonuniform grid, the largest grid spacing is about 20 times larger than the smallest grid spacing. In addition, to analyze the grid convergence, the simulations are conducted in 3 grid cases; $60 \times 60, 120 \times 120$ and 240×240. The grid spacings of the 120×120 and 240×240 grids are half the size of that of 60×60 and 120×120 grids, respectively.

Fig. 2 shows the simulation results of the flow field. The density contours and streamlines are shown in this picture, and the streamlines are also magnetic field lines. This result is in good agreement with the result in the previous study [19].

Fig. 3 shows the convergence history of the residual density (R_d) toward the steady state solution as a function of the CPU time. The R_d is defined as

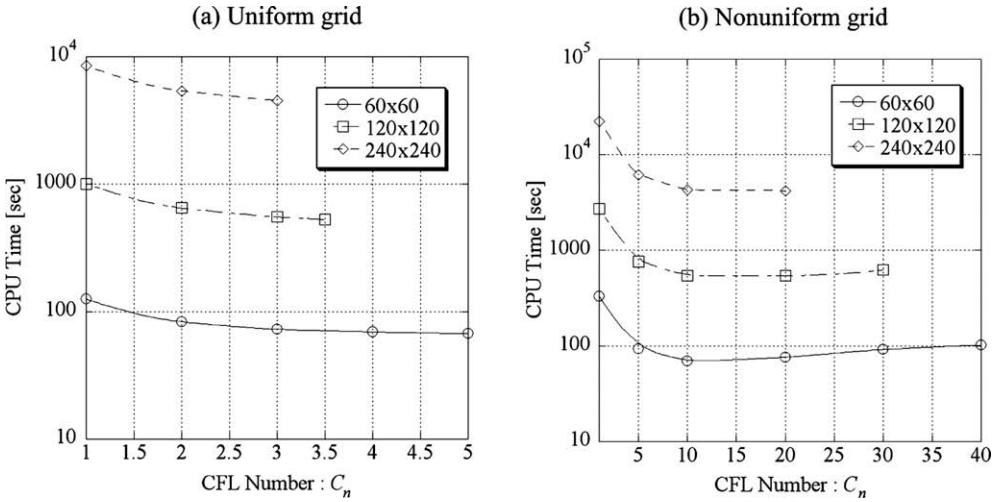

Fig. 4. Convergence time as a function of the CFL number.

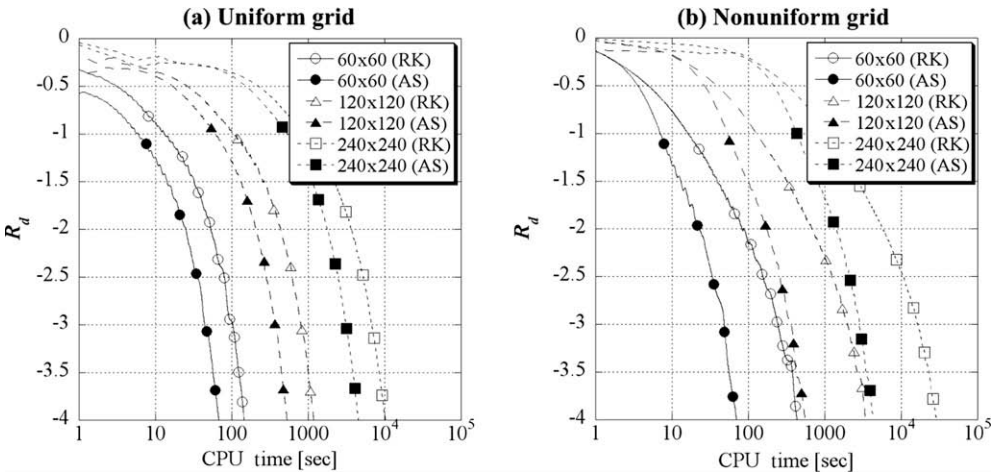

Fig. 5. Comparison of the ADI-SGS scheme with the 2nd-order Runge–Kutta scheme: the convergence histories of the ADI-SGS scheme (AS) and the Runge–Kutta scheme (RK).

$$R_d^n = \log\left(C^0\sqrt{\frac{\sum_{i,j}\left(\rho_{i,j}^n - \rho_{i,j}^{n-1}\right)^2}{i_{\max}j_{\max}}}\right). \tag{15}$$

Here $i_{\max}$ and $j_{\max}$ are the number of cells and the C^0 is chosen to satisfy $R_d^1 = 0$. We define that the steady state is achieved when the R_d drops 4-orders of magnitude. The simulations are conducted for various CFL numbers (C_n) to investigate the convergence performance of the scheme. The time step interval is derived at every time step as

$$\Delta t = C_n \min_{i,j}\left(\frac{\Omega_{i,j}}{\max\left((|\boldsymbol{v}_{i,j}\cdot\boldsymbol{n}_\xi| + c^f_{\xi i,j})l_{\xi i,j}, (|\boldsymbol{v}_{i,j}\cdot\boldsymbol{n}_\eta| + c^f_{\eta i,j})l_{\eta i,j}\right)}\right). \tag{16}$$

$\Omega_{i,j}$ is the area of the cell, $\boldsymbol{n}_{\xi,\eta}$ are unit directional vectors in the ξ and η direction and $l_{\xi,\eta}$ are the width of the cell in the ξ and η direction, respectively.

First, the dependency of the convergence performance on the grid resolution is discussed. The convergence times to the steady state are plotted as a function of the CFL numbers for each grid resolution in Fig. 4. In the case of the uniform grid, the maximum CFL numbers are 5.0, 3.5 and 3.0 in the 60×60, 120×120 and 240×240 grids, respectively. When larger CFL numbers than these maximum values are chosen, the computations are unstable. Fig. 4(a) shows that the shortest convergence time can be got at the largest CFL numbers. On the other hand, in the case of the nonuniform grid, the maximum CFL numbers are 40.0, 30.0 and 20.0 in the 60×60, 120×120 and 240×240 grids, respectively, however the shortest convergence time can be got when the CFL number is between 10.0 and 20.0 in all grid cases (as shown in Fig. 4(b)).

Next, the convergence performance of the ADI-SGS scheme is compared with that of the 2nd-order Runge–Kutta explicit scheme. Fig. 5 shows the convergence histories of the ADI-SGS scheme and the 2nd-order Runge–Kutta scheme for each grid resolution. Here, the CFL number is 0.7 in the case of the Runge–Kutta scheme, and in the case of the ADI-SGS scheme, the CFL numbers are 5.0, 3.5, 3.0 and 10.0 in the uniform 60×60, 120×120, 240×240 grids and the nonuniform grids, respectively. Note that computational time for one time step of the Runge–Kutta scheme is almost equal to that of the ADI-SGS scheme. Fig. 5(a) shows that the ADI-SGS scheme can get the steady state about twice as rapid as the Runge–Kutta scheme for all grid resolutions in the case of the uniform grid. On the other hand, Fig. 5(b) shows that the ADI-SGS scheme can get the steady state about six times as rapid as the Runge–Kutta scheme for all grid resolutions in the case of the nonuniform grid.

In these simulations, the maximum CFL number is restricted due to the initial transients in which the physical states vary considerably. We can choose much larger CFL number by computing some dozen initial time steps with a smaller CFL number; the CFL number could be chosen larger than 100 in all grid cases in our simulations.

4. Conclusions

The ADI-SGS scheme are implemented to the ideal magnetohydrodynamic equations with a finite-volume method using a Riemann solver. The ADI-SGS scheme is an implicit time integration scheme used for the hydrodynamic equations. In our approach, the Jacobian-free technique is also used. To verify the convergence performance, a steady bow shock flow problem was simulated, and following results were obtained.

- Smaller CFL number has to be used in the finer grid resolution, and much larger CFL number can be used in the nonuniform grid than in the uniform grid. However, the maximum CFL number is restricted due to the initial transients and so computing some dozen initial time steps with a smaller CFL number is effective way.
- The ADI-SGS scheme improves the convergence performance compared with the Runge–Kutta scheme, and the effectiveness of the ADI-SGS scheme is much better in the nonuniform grid than in the uniform grid.

Acknowledgment

This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for JSPS Fellows, 19-2798, 2007.

References

[1] O.S. Jones, U. Shumlak, D.S. Eberhardt, An implicit scheme for nonideal magnetohydrodynamics, J. Comput. Phys. 130 (1997) 231.
[2] R. Keppens, G. Tóth, M.A. Botchev, A. Van Der Ploeg, Implicit and semi-implicit scheme: algorithms, Int. J. Numer. Method Fluids 30 (1999) 335.
[3] L. Chacon, D. Knoll, J. Finn, An implicit, nonlinear reduced resistive MHD solver, J. Comput. Phys. 178 (2002) 15.
[4] D.R. Reynolds, R. Samtaney, C.S. Woodward, A fully implicit numerical method for single-fluid resistive magnetohydrodynamics, J. Comput. Phys. 219 (2006) 144.
[5] S.C. Jardin, J.A. Breslau, N. Ferraro, A high-order implicit finite element method for integrating the two-fluid magnetohydrodynamic equations in two dimensions, J. Comput. Phys. 226 (2007) 2146.
[6] S. Obayashi, K. Fujii, S. Gavali, Navier–Stokes simulation of wind-tunnel flow using LU-ADI factorization algorithm, NASA-TM-100042, 1998.

[7] N. Iizuka, Study of Mach Number Effect on the Dynamic Stability of a Blunt Re-entry Capsule, Ph.D. Thesis, University of Tokyo, 2006.
[8] R.M. Beam, R.F. Warming, An implicit factored scheme for the compressible Navier–Stokes equations, AIAA J. 16 (1976) 393.
[9] S. Yoon, A. Jameson, Lower–upper Symmetric–Gauss–Seidel method for the Euler and Navier–Stokes equations, AIAA J. 26 (1988) 1025.
[10] J. Augustinus, K.A. Hoffmann, S. Harada, Effect of magnetic field on the structure of high-speed flows, J. Spacecraft Rockets 35 (1998) 639.
[11] K. Fujii, S. Obayashi, Practical application of improved LU-ADI scheme for the three-dimensional Navier–Stokes computations of transonic viscous flows, AIAA J. 25 (1987) 369.
[12] K. Nakahashi, D. Sharov, S. Kano, M. Kodera, Applications of unstructured hybrid grid method to high-Reynolds number viscous flow, Int. J. Numer. Method Fluids 31 (1999) 97.
[13] S.R. Chakravarthy, Relaxation methods for unfactored implicit upwind schemes, AIAA paper 84-0165, 1984.
[14] S. Yoon, D. Kwak, Implicit Navier–Stokes solver for three-dimensional compressible flows, AIAA J. 30 (1992) 2653.
[15] R.F. Warming, R.M. Beam, On the construction and application of implicit factored schemes for conservation laws, SIAM-AMS Proc. 11 (1977) 85.
[16] T.H. Pulliam, D.S. Chaussee, A diagonal form of an implicit approximate-factorization algorithm, J. Comput. Phys. 63 (1986) 157.
[17] A.A. Barmin, A.G. Kulikovskiy, N.V. Pogorelov, Shock-capturing approach and nonevolutionary solution in magnetohydrodynamics, J. Comput. Phys. 126 (1996) 77.
[18] G. Tóth, The $\nabla \cdot \boldsymbol{B}$ constraint in shock-capturing magnetohydrodynamics codes, J. Comput. Phys. 161 (2000) 605.
[19] H. De Sterck, A. Csik, D.V. Abeele, S. Poedts, H. Deconinck, Stationary two-dimensional magnetohydrodynamic flows with shocks: characteristic analysis and grid convergence study, J. Comput. Phys. 166 (2001) 28.