Astronomy and Computing 30 (2020) 100341

Contents lists available at ScienceDirect

Astronomy and Computing

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

Full length article

An efficient parallel semi-implicit solver for anisotropic thermal conduction in the solar corona

J. Ye [a,b,*], C. Shen [c], J. Lin [a,b,d], Z. Mei [a,b]

[a] *Yunnan Observatories, Chinese Academy of Sciences, P.O. Box 110, Kunming, Yunnan 650216, China*
[b] *Center for Astronomical Mega-Science, Chinese Academy of Sciences, Beijing 100012, China*
[c] *Smithsonian Astrophysical Observatory, 60 Garden Street, MS 15, Cambridge, MA 02138, United States of America*
[d] *University of Chinese Academy of Sciences, Beijing 100049, China*

ARTICLE INFO

Article history:
Received 17 April 2019
Accepted 19 October 2019
Available online 23 October 2019

Keywords:
Magnetohydrodynamics
Spitzer conductivity
Anisotropic diffusion
Numerical methods
Parallel algorithm

ABSTRACT

Anisotropic thermal conduction plays an important role in determining the structure of the hot plasma in the solar corona. When hot plasma appears, the conductivity rises with temperature and becomes highly nonlinear. Explicit solvers for parabolic problems often lead to much smaller time-steps limited by a Courant–Friedrichs–Lewy (CFL) condition in comparison with hyperbolic Magnetohydrodynamics (MHD) equations. In this work, we present a pseudo-linear, directionally-split, semi-implicit method allowing for large time-steps as well as the optimized parallelization algorithm, integrated with the MHD solver. Our scheme can perfectly preserve the monotonicity and the geometry of shocks and discontinuities in complex MHD problems. Two sets of numerical tests show that an increase in time step of $\sim$ 600 can be easily achieved with an acceptable error by our scheme compared to explicit methods, and the use of large time-steps can still follow fast dynamic processes reliably. In addition, the extendibility studies have proven that the associated parallel efficiency is comparably high. This method is also useful for any kind of time-dependent conductivity problems for the solar applications in the future.

1. Introduction

Solar activity is usually recognized in temperature profiles from observations at different wavelengths, and anisotropic thermal conduction plays a non-negligible role for the associated hot structures. Particularly, in the process of solar eruptions, the released magnetic energy is converted partially into the thermal energy, which contributes to a significant variation on the temperature of plasma. For instance, Interface Region Imaging Spectrograph (IRIS) bombs (Peter et al., 2014) were observed in the low chromosphere, of which the temperature reached to 8×10^4 K. Also, micro-flares of 10^6 K observed in the chromosphere might be related to the coronal heating (Shimizu, 2015). Thermal conduction also has a drastic effect on the spectra for low-frequency nanoflares diagnosis (Bradshaw et al., 2012). In the corona, X-ray flux was successfully identified on the top of hot flare loops (Liu, 2013; Liu et al., 2013), and even hotter regions of the order of 10^7 K can be found in the coronal mass ejection (CME)/flare reconnecting current sheet (Ciaravella et al., 2002; Reeves and Golub, 2011). Lemen et al. (2012) summarized the effective range of the temperature response function in each Atmospheric Imaging Assembly (AIA) and IRIS band, and associated hot structures at different heights. The spatial distribution of temperature observed by various devices can provide a better understanding for the energy conversion and release in CME events. On the other hand, due to specific objectives of diverse observational instruments, the observation bands from the filters should greatly differ from each other in order to obtain a series of distinguishable images in each band.

In order to explain the fundamental physical mechanisms related to the hot structures in observations, astrophysical MHD simulations are used, where thermal conduction is anisotropic. Sharma and Hammett (2007) have pointed out that simple finite-differencing of the anisotropic diffusion equation leads to negative temperatures at large temperature gradients, because unphysical numerical fluxes are created at cell surfaces. This eventually results in an imaginary sound speed and the stationary temperature becomes complex. They also show that the use of slope limiters (Leveque, 2002) to calculate the fluxes at cell surfaces keeps the temperature oscillations suppressed and prevents the creation of artificial extrema. Thus, the temperature is always positive as required physically for a CFL stable timestep. A similar fully implicit method was developed by Kannan et al. (2016, 2017) to simulate the black hole feedback taking account into thermal conduction with respect to a moving mesh.

[*] Corresponding author at: Yunnan Observatories, Chinese Academy of Sciences, P.O. Box 110, Kunming, Yunnan 650216, China.
E-mail address: yj@ynao.ac.cn (J. Ye).

https://doi.org/10.1016/j.ascom.2019.100341

Current open source MHD codes, such as ZEUS (Stone and Norman, 1992a,b; Stone et al., 1992), ATHENA (Gardiner and Stone, 2005, 2008; Stone and Gardiner, 2009) and NIRVANA (Ziegler, 2005, 2008, 2011), solve explicitly the MHD equations including thermal conduction. The timesteps of fully explicit methods are limited by the CFL condition. In some of astronomical environments, the timestep provided by thermal conduction as $\delta t < \Delta x^2/2\kappa_\parallel$, where Δx is the grid spacing and $\kappa_\parallel$ is the anisotropic diffusion coefficient, can be much smaller than the MHD CFL time limiter. In our previous work about the cascading (turbulent) reconnection during solar eruptions (Ye et al., 2019), thermal conduction was not included; otherwise it would lead to an enormous computation time because of the above limitation ($\sim$ 1000 times slower). For this reason, thermal conduction alone is expected to be treated implicitly, while the MHD parts are treated explicitly. Balsara et al. (2008) have developed a temporally second-order implicit–explicit solver to do so, but it is not monotonicity-preserving, i.e. the temperature oscillations remain till late times of the simulation. Further, the implementation of their iterative method is somewhat difficult, because it requires a fairly large number of iterations per timestep when the diffusion is strongly non-linear. We also experimented with the classical Newton's method (Atkinson, 1989) to achieve the fully implicit thermal conduction solver, but the convergence becomes worse. Since the Jacobian matrix is often ill-posed (i.e. of a large condition number) when the iterative solution approaches the exact solution, a large timestep might lead to a bad behavior in finding the optimized updates and eventually a failure in convergence. Alternatively, we considered the alternate direction implicit (ADI) schemes which are unconditionally stable for isotropic diffusion applications, but the explicit treatment for the transverse terms $\partial^2 T/\partial x\partial y$ makes the scheme unstable for timesteps larger than a few times the CFL timestep. Sharma and Hammett (2011) then report a directionally-split semi-implicit method for constant-coefficient anisotropic diffusion in the intracluster medium, which realizes a speed-up of $\sim$ 1000 in contrast to fully explicit methods.

For a realistic application in the corona, namely magnetic reconnection during the flares, ejecting hot plasma to form turbulent hot structures of flare loops, the parallel and perpendicular conductivity coefficients, $\kappa_\parallel$ and $\kappa_\perp$, are modeled by Spitzer theory (Spitzer, 1962), which are highly nonlinear with respect to temperature. In addition, it is found in most solar applications that $\kappa_\perp \ll \kappa_\parallel$, conduction is mainly along the magnetic field. The temperature-dependent nonlinearity gives rise to a high risk of instabilities for large timesteps. The Super TimeStepping method (Meyer et al., 2012) is also very popular in many codes, which provides a Runge–Kutta kind of corrections allowing for large, second-order accurate time-steps. A $\sim s^2$ times larger timestep needs s inner iterations at each time, and the number s should be an odd integer for the sake of stability. In other words, this permits an s-fold gain in computation cost over explicit forward Euler sub-cycling.

We present here a novel semi-implicit scheme for the nonlinear diffusion problem as well as an optimization of the parallelization algorithm. This method combines the fluxes directionally-split technique from Sharma and Hammett (2011) and the ADI schemes for nonlinear diffusion equations as described in Witelski and Bowen (2003) and Bruno and Jimenez (2014). The resulting scheme is not only unconditionally stable and $\sim$ 600 times faster than fully explicit methods, but also easy to parallelize in any cluster.

The paper is organized as follows: Section 2 presents the detailed numerical methods and parallelization algorithm. Two tests regarding the accuracy and the parallel efficiency are presented in Section 3. Lastly, conclusion and discussion are described in Section 4.

2. Numerical methods

We consider the dimensionless magnetohydrodynamic equations including the resistivity and the anisotropic diffusion simplified as follows:

$$\frac{\partial\rho}{\partial t} + \nabla\cdot[\rho\mathbf{v}] = 0, \tag{1}$$

$$\frac{\partial\rho\mathbf{v}}{\partial t} + \nabla\cdot[\rho\mathbf{v}\mathbf{v} - \mathbf{B}\mathbf{B} + P^*] = 0, \tag{2}$$

$$\frac{\partial E}{\partial t} + \nabla\cdot[(E + P^*)\mathbf{v} - \mathbf{B}(\mathbf{B}\cdot\mathbf{v})]$$
$$= \nabla\cdot(\eta\mathbf{B}\times(\nabla\times\mathbf{B}) + \kappa_\parallel(\nabla T\cdot\hat{\mathbf{B}})\hat{\mathbf{B}}), \tag{3}$$

$$\frac{\partial\mathbf{B}}{\partial t} = \nabla\times(\mathbf{v}\times\mathbf{B} - \eta\nabla\times\mathbf{B}), \tag{4}$$

$$P^* = P + \frac{\mathbf{B}\cdot\mathbf{B}}{2}, \tag{5}$$

$$E = P/(\gamma-1) + \frac{\rho(\mathbf{v}\cdot\mathbf{v})}{2} + \frac{\mathbf{B}\cdot\mathbf{B}}{2}, \tag{6}$$

$$P = \rho T. \tag{7}$$

where ρ is the plasma mass density, $\mathbf{v}$ is the velocity field, E is the total energy density, $\mathbf{B}$ is the magnetic field, $\hat{\mathbf{B}} = \mathbf{B}|\mathbf{B}|$ is the unit vector in the direction of the magnetic field, T is the temperature, P is the gas pressure, η is the normalized magnetic resistivity and γ is the ratio of specific heats set to 5/3 (ideal gas). The conductivity coefficient $\kappa_\parallel$ used in the heat conduction term is assumed either constant value or nonlinear function of temperature. Regarding most solar applications, conduction along the magnetic field is dominated by electrons, and Spitzer (1962) gives the description of the parallel conductivity in normalized form

$$\kappa_\parallel = c_1\cdot\frac{1.84\times10^{-10}}{Z\ln\Lambda}T^{5/2}. \tag{8}$$

Here, $\ln\Lambda$ is the Coulomb logarithm set to 30, Z is the mean ionic charge number set to 1, the normalization factor $c_1 = \mu_0 T_N^{7/2}/(B_N^2 L_N v_{AN})$. We choose T_N, B_N, L_N and ρ_N as the normalization units for temperature, magnetic field, length and mass density, respectively. Actually, T_N can be deduced from B_N and ρ_N. Thus the reference Alfvén speed is given by $v_{AN} = B_N/\sqrt{\mu_0\rho_N}$, where the magnetic permeability $\mu_0 = 4\pi\times10^{-7}$. Note that we do not include the cross-field conduction term in the above MHD model, because the perpendicular conductivity $\kappa_\perp$ is computed to be several orders of magnitude smaller than the parallel one in the corona. In practice, we only introduce the numerical diffusion instead of the cross-field diffusion. To complete the above equations, the divergence-free condition ($\nabla\cdot\mathbf{B} = 0$) should be satisfied at all times. Current popular astrophysical codes, as mentioned previously, are used to solving the whole system by two steps using Strang splitting. Within the spirit of such operator splitting schemes one formally rewrites Eqs. (1)–(7) as

$$\frac{\partial U}{\partial t} = L_{MHD}(U) + L_{cond}(U),$$

where U is the conservative parameters, L_{MHD} and L_{cond} represent the gradients of the MHD fluxes and the conduction fluxes, respectively. The first step consists of solving the resistive MHD equations $\partial U/\partial t = L_{MHD}(U)$; the second step updates only the energy change by $\partial E/\partial t = \nabla\cdot(\kappa_\parallel(\nabla T\cdot\hat{\mathbf{B}})\hat{\mathbf{B}})$. For simplicity, the second step is equivalent to solve a nonlinear thermal conduction problem rewritten by

$$\frac{\partial T}{\partial t} = \nabla\cdot(\kappa(T)(\nabla T\cdot\hat{\mathbf{B}})\hat{\mathbf{B}}),$$
$$+\text{Boundary conditions.} \tag{9}$$

Here, $\kappa(T)$ is proportional to $\kappa_{\parallel}$. A number of well-developed explicit schemes are used to solve the MHD step, such as operator splitting finite-differencing (Stone and Norman, 1992b) or unsplit Godunov schemes (Stone and Gardiner, 2009). We often follow the rules that the MHD time-step is computationally expensive, and that should not be reduced in any way. Thus the MHD solver takes it largest time-step permitted by its CFL condition at each time t_n:

$$\begin{aligned}\delta t =& C_0 \min(\frac{\Delta x}{|v^n_{x,i,j,k}| + C^n_{f,i,j,k}}, \frac{\Delta y}{|v^n_{y,i,j,k}| + C^n_{f,i,j,k}}, \\ & \frac{\Delta z}{|v^n_{z,i,j,k}| + C^n_{f,i,j,k}}),\end{aligned} \tag{10}$$

where $C_0 \leq 0.5$ is the CFL number, v^n_x, v^n_y, v^n_z are the fluid velocities in x, y, z−directions and C^{n+1}_f denotes the fast magnetosonic speed evaluated at cell centers. Therefore, the next step for solving the parabolic conduction equation is crucial for a proper timestep δt to compete with the one obtained from the MHD step.

2.1. Anisotropic diffusion

Our work is dedicated to optimize the method to solve the anisotropic thermal conduction problem embedded in the resistive MHD system. Without loss of generality, we consider the two dimensional uniform grid in Cartesian coordinates. Eq. (9) is integrated in a single cell V_{ij} as

$$\begin{aligned}\int_{V_{ij}} \frac{\partial T}{\partial t} d\mathbf{V} =& \oint_{S_{ij}} \kappa(T)(\nabla T \cdot \hat{\mathbf{B}})\hat{\mathbf{B}} \cdot \vec{\mathbf{n}} d\mathbf{S} \\ =& \int_{y_{j-1/2}}^{y_{j+1/2}} \left[\kappa(T)\frac{\partial T}{\partial x}\hat{B}_x\hat{B}_x\right]_{x_{i-1/2}}^{x_{i+1/2}} dy \\ &+ \int_{x_{i-1/2}}^{x_{i+1/2}} \left[\kappa(T)\frac{\partial T}{\partial y}\hat{B}_y\hat{B}_y\right]_{y_{j-1/2}}^{y_{j+1/2}} dx \\ &+ \int_{y_{j-1/2}}^{y_{j+1/2}} \left[\kappa(T)\frac{\partial T}{\partial y}\hat{B}_x\hat{B}_y\right]_{x_{i-1/2}}^{x_{i+1/2}} dy \\ &+ \int_{x_{i-1/2}}^{x_{i+1/2}} \left[\kappa(T)\frac{\partial T}{\partial x}\hat{B}_x\hat{B}_y\right]_{y_{j-1/2}}^{y_{j+1/2}} dx.\end{aligned} \tag{11}$$

The terms on the right side are the fluxes flowing through the cell faces S_{ij}. In the manner of finite volumes discretization, the system is updated by two-steps Strang splitting along the x- and y-direction. Then, the fluxes are taken semi-implicitly in each direction as follows:

$$\begin{aligned}\frac{T^*_{i,j} - T^n_{i,j}}{\Delta t} =& \kappa(T^*_{i+1/2,j})\hat{B}^2_{x,i+1/2,j}\frac{T^*_{i+1,j} - T^*_{i,j}}{\Delta x^2} \\ & -\kappa(T^*_{i-1/2,j})\hat{B}^2_{x,i-1/2,j}\frac{T^*_{i,j} - T^*_{i-1,j}}{\Delta x^2} \\ & +\kappa(T^n_{i+1/2,j})\frac{\hat{B}_{x,i+1/2,j}\hat{B}_{y,i+1/2,j}}{\Delta x \Delta y}\Delta\bar{T}^n_{i+1/2,j} \\ & -\kappa(T^n_{i-1/2,j})\frac{\hat{B}_{x,i-1/2,j}\hat{B}_{y,i-1/2,j}}{\Delta x \Delta y}\Delta\bar{T}^n_{i-1/2,j},\end{aligned} \tag{12}$$

$$\begin{aligned}\frac{T^{n+1}_{i,j} - T^*_{i,j}}{\Delta t} =& \kappa(T^{n+1}_{i,j+1/2})\hat{B}^2_{y,i,j+1/2}\frac{T^{n+1}_{i,j+1} - T^{n+1}_{i,j}}{\Delta y^2} \\ & -\kappa(T^*_{i-1/2,j})\hat{B}^2_{y,i,j-1/2}\frac{T^{n+1}_{i,j} - T^{n+1}_{i,j-1}}{\Delta y^2} \\ & +\kappa(T^*_{i,j+1/2})\frac{\hat{B}_{y,i,j+1/2}\hat{B}_{x,i,j+1/2}}{\Delta x \Delta y}\Delta\bar{T}^*_{i,j+1/2} \\ & -\kappa(T^*_{i,j-1/2})\frac{\hat{B}_{y,i,j-1/2}\hat{B}_{x,i,j-1/2}}{\Delta x \Delta y}\Delta\bar{T}^*_{i,j-1/2},\end{aligned} \tag{13}$$

where $\{i, j\}$ are the coordinates of grid cells and $\{i \pm 1/2, j \pm 1/2\}$ represent the appropriate cell faces. T^* stands for the intermediate state for temperature, T^n and T^{n+1} indicate the state at $t = t_n$ and $t = t_{n+1}$, respectively. Unlike the ADI scheme (implicit meaning), only the cross-gradient terms are treated explicitly here (i.e. semi-implicit), and our method is first order accurate in time. The notation

$$\begin{aligned}\Delta\bar{T}_{i+1/2,j} =& L(T_{i+1,j+1} - T_{i+1,j}, T_{i+1,j} - T_{i+1,j-1}, \\ & cT_{i,j+1} - T_{i,j}, T_{i,j}a - T_{i,j-1}),\end{aligned} \tag{14}$$

$$\begin{aligned}\Delta\bar{T}_{i,j+1/2} =& L(T_{i+1,j+1} - T_{i,j+1}, T_{i,j+1} - T_{i-1,j+1}, \\ & T_{i+1,j} - T_{i,j}, T_{i,j} - T_{i-1,j})\end{aligned} \tag{15}$$

are the central differences of the temperature at appropriate cell faces, and L stands for the slope limiter of van Leer (1977) to capture the temperature extrema, given by

$$\begin{aligned}L(a, b) =& \frac{2ab}{a+b}, \quad \text{if} \quad ab > 0, \\ =& 0, \quad \text{else}.\end{aligned} \tag{16}$$

Furthermore, its version for 4 arguments is described as $L(a, b, c, d) = L(L(a, b), L(c, d))$, which is symmetric by the definition. However, other slope limiters are also available, such as minmod and monotonized central (MC) limiters (see Leveque, 2002 for more details). We have also experimented with the sharper limiter MC, and found that the condition of the matrix is more sensible to large time-steps than the more diffusive limiter such as van Leer. Unlike the constant-coefficient linear problem in Sharma and Hammett (2011), we still have the unknown heat conduction coefficient $\kappa(T^*)$ in Eq. (12) as well as $\kappa(T^{n+1})$ in Eq. (13). To skip the complexity of the nonlinear solve at each step, we use the predictor of the solution at the time t^{n+1} that result from qth order of extrapolatory approximations with known values $\{T^n, \ldots, T^{n-q+1}\}$ at previous time levels. It functions as a predictor–corrector estimation at each time-step. It is found that the use of this predictor can still preserve the discontinuity on temperature profiles by our experiments. Particularly, the approximation of T^{n+1} takes the polynomial form, given by

$$\tilde{T}^{n+1} = \sum_{i=0}^{q-1} \omega_i T^{n-i}, \tag{17}$$

where the coefficients ω_i take the values from Table 1. If we do not make the prediction at each time-step, then they are equivalent to Eqs. (12)–(13) with the time-lagged explicit coefficients $\kappa(T)$, hence we call them pseudo-linear approximations for (17). In particular, we have $T^{n+1} = \tilde{T}^{n+1} + O(\Delta t^q)$ as $\Delta t \rightarrow 0$. To ensure the convergence of our method, the temperature should be always positive valued, and we made a correction with a predefined tiny number ($\sim 10^{-20}$) as the temperature floor. In most applications, this temperature floor also can be defined depending on local physical environments, such as 10^5 K in solar corona and 10^4 K in solar chromosphere, so that $\tilde{T}^{n+1} = max\{\tilde{T}^{n+1}, \text{TinyNumber}\}$ at each time step. Substituting T^* and T^{n+1} by $\tilde{T}^{n+1}$ into the heat conduction coefficients, we get thus

$$\kappa(T^*) \approx \kappa(\tilde{T}^{n+1}), \kappa(T^{n+1}) \approx \kappa(\tilde{T}^{n+1}). \tag{18}$$

Thanks to these approximations, we transform Eqs. (12) and (13) into a linear problem for T^{n+1}. Accordingly, this substitution, whatever the choice of q is, preserves the first order accuracy in time found in Eqs. (12), (13). Since the high order approximations for $q \geq 4$ are more sensible to the numerical errors proven by our tests, we set practically $q = 2$ for all test simulations in Section 3.

Table 1
Extrapolation coefficients ω_i for T^{n+1}.

q	ω_0	ω_1	ω_2	ω_3	ω_4
2	2	−1	0	0	0
3	3	−3	1	0	0
4	4	−6	4	−1	0
5	5	−10	10	−5	1

Generally, these two equations can be represented in a linear algebra problem as follows:

$$a_i x_{i-1} + b_i x_i + c_i x_{i+1} = f_i, \qquad i = 1, \ldots, N-1. \tag{19}$$

Hence, a tridiagonal matrix system is obtained for each directional update. To solve such systems, the Thomas' algorithm is available, and the solution can be obtained in $O(N)$ operations instead of $O(N^3)$ required by Gaussian elimination. Normally, this method is stable when the matrix is diagonally dominant.

Since the conductivity coefficient varies in magnitude, we have defined a minimum , κ_{min} and a maximum, κ_{max} without loss of generality, so that $\kappa_{min} \le \kappa(T) \le \kappa_{max}$. For a fixed volume V_{ij}, we have the following relationship at time $t = t^{n+1}$ deduced from Eq. (12) for x-direction:

$$\begin{aligned}\frac{T^*_{i,j} - T^n_{i,j}}{\Delta t} \le\ & \kappa_{max}\hat{B}^2_{x,i+1/2,j}\frac{T^*_{i+1,j} - T^*_{i,j}}{\Delta x^2} \\ & -\kappa_{max}\hat{B}^2_{x,i-1/2,j}\frac{T^*_{i,j} - T^*_{i-1,j}}{\Delta x^2} \\ & +\kappa_{max}\frac{\hat{B}_{x,i+1/2,j}\hat{B}_{y,i+1/2,j}}{\Delta x \Delta y}\Delta \bar{T}^n_{i+1/2,j} \\ & -\kappa_{max}\frac{\hat{B}_{x,i-1/2,j}\hat{B}_{y,i-1/2,j}}{\Delta x \Delta y}\Delta \bar{T}^n_{i-1/2,j}, \text{ if } \quad dT^* > 0,\end{aligned} \tag{20}$$

or

$$\begin{aligned}\frac{T^*_{i,j} - T^n_{i,j}}{\Delta t} \le\ & \kappa_{min}\hat{B}^2_{x,i+1/2,j}\frac{T^*_{i+1,j} - T^*_{i,j}}{\Delta x^2} \\ & -\kappa_{min}\hat{B}^2_{x,i-1/2,j}\frac{T^*_{i,j} - T^*_{i-1,j}}{\Delta x^2} \\ & +\kappa_{min}\frac{\hat{B}_{x,i+1/2,j}\hat{B}_{y,i+1/2,j}}{\Delta x \Delta y}\Delta \bar{T}^n_{i+1/2,j} \\ & -\kappa_{min}\frac{\hat{B}_{x,i-1/2,j}\hat{B}_{y,i-1/2,j}}{\Delta x \Delta y}\Delta \bar{T}^n_{i-1/2,j}, \text{ if } \quad dT^* < 0,\end{aligned} \tag{21}$$

where $dT^* = T^*_{ij} - T^n_{ij}$. Similarly, for y-direction, we have

$$\begin{aligned}\frac{T^{n+1}_{i,j} - T^*_{i,j}}{\Delta t} \le\ & \kappa_{max}\hat{B}^2_{x,i+1/2,j}\frac{T^{n+1}_{i+1,j} - T^{n+1}_{i,j}}{\Delta x^2} \\ & -\kappa_{max}\hat{B}^2_{x,i-1/2,j}\frac{T^*_{i,j} - T^*_{i-1,j}}{\Delta x^2} \\ & +\kappa_{max}\frac{\hat{B}_{x,i+1/2,j}\hat{B}_{y,i+1/2,j}}{\Delta x \Delta y}\Delta \bar{T}^n_{i+1/2,j} \\ & -\kappa_{max}\frac{\hat{B}_{x,i-1/2,j}\hat{B}_{y,i-1/2,j}}{\Delta x \Delta y}\Delta \bar{T}^n_{i-1/2,j}, \\ & \text{if } \quad dT^{n+1} > 0,\end{aligned} \tag{22}$$

or

$$\begin{aligned}\frac{T^{n+1}_{i,j} - T^*_{i,j}}{\Delta t} \le\ & \kappa_{min}\hat{B}^2_{x,i+1/2,j}\frac{T^{n+1}_{i+1,j} - T^{n+1}_{i,j}}{\Delta x^2} \\ & -\kappa_{min}\hat{B}^2_{x,i-1/2,j}\frac{T^*_{i,j} - T^*_{i-1,j}}{\Delta x^2} \\ & +\kappa_{min}\frac{\hat{B}_{x,i+1/2,j}\hat{B}_{y,i+1/2,j}}{\Delta x \Delta y}\Delta \bar{T}^n_{i+1/2,j} \\ & -\kappa_{min}\frac{\hat{B}_{x,i-1/2,j}\hat{B}_{y,i-1/2,j}}{\Delta x \Delta y}\Delta \bar{T}^n_{i-1/2,j}, \\ & \text{if } \quad dT^{n+1} > 0,\end{aligned} \tag{23}$$

where $dT^{n+1} = T^{n+1}_{ij} - T^*_{ij}$. To prove the numerical stability, we substitute the temperature eigenmode $T(x, y, t) = T_0 r(t)$ $e^{-i(k_x x + k_y y)}$ into Eqs. (20), (21), (22), (23) to determine the amplification factor $r(t)$, where k_x and k_y are the wavenumbers along x-axis and y-axis. The amplification factor can be rewritten by $r = r_1 r_2$, with r_1 and r_2 the amplification factors in x- and y-direction, respectively. For simplicity, we assume that $\hat{B}_x$, $\hat{B}_y$ are constant in space. The calculation is similar to that in Sharma and Hammett (2011). We have thus, for x-direction,

$$r_1 \le \frac{1 - \frac{\kappa_{max}\Delta t}{\Delta x \Delta y}\hat{B}_x\hat{B}_y sin(k_x\Delta x) sin(k_y\Delta y)}{1 + 4\frac{\kappa_{max}\Delta t}{\Delta x^2}\hat{B}^2_x sin^2(k_x\Delta x/2)}, \tag{24}$$

or

$$r_1 \le \frac{1 - \frac{\kappa_{min}\Delta t}{\Delta x \Delta y}\hat{B}_x\hat{B}_y sin(k_x\Delta x) sin(k_y\Delta y)}{1 + 4\frac{\kappa_{min}\Delta t}{\Delta x^2}\hat{B}^2_x sin^2(k_x\Delta x/2)}. \tag{25}$$

And, for y-direction,

$$r_2 \le \frac{1 - \frac{\kappa_{max}\Delta t}{\Delta x \Delta y}\hat{B}_x\hat{B}_y sin(k_x\Delta x) sin(k_y\Delta y)}{1 + 4\frac{\kappa_{max}\Delta t}{\Delta y^2}\hat{B}^2_y sin^2(k_x\Delta x/2)}, \tag{26}$$

or

$$r_2 \le \frac{1 - \frac{\kappa_{min}\Delta t}{\Delta x \Delta y}\hat{B}_x\hat{B}_y sin(k_x\Delta x) sin(k_y\Delta y)}{1 + 4\frac{\kappa_{min}\Delta t}{\Delta y^2}\hat{B}^2_y sin^2(k_x\Delta x/2)}. \tag{27}$$

Consequently, one can prove that the full time step factor $|r| = |r_1 r_2| \le 1$ is guaranteed for any Δt in two dimensions at the given time $t = t^{n+1}$, and our method is unconditionally stable as is usual for implicit methods. For three dimensions, the expansion of our method is not unconditionally stable any more, because the amplification factor $|r|$ could be greater than 1 for a large Δt. However, Sharma and Hammett (2011) also suggest a 4-step method in three dimensions, which was numerically verified to be unconditionally stable for a constant conductivity. We leave it for future work, because for the applications of our current interest the thermal structure would be better understood in 2D (or 2.5D).

In closing, the accuracy of the present solver is only first order in time and second order in space. Although the above linear stability analysis predicts an unconditional stability, our method is not numerically stable for an enormous Δt, because the explicit 'transverse' flux terms ($\partial^2 T/\partial x \partial y$) in Eqs. (12) and (13) are symmetrized with respect to the $x-$ and $y-$updates. Additionally, extremely nonlinearity of the heat conduction coefficient or limiters might lead to a high risk of instability to large time-steps.

In order to speed up the simulations, we made an exhaustive effort to parallelize the thermal conduction problem (second step) to integrate with the MHD updates (first step). There are two major parallel algorithms that are widely used for the tridiagonal system (19). One is the domain decomposition algorithm (Wang and Georgiadis, 1991), the other uses higher dimensional linear recurrences to solve the diatriangular linear system (Wang and Vanka, 1993). However, to be in harmony with the parallelized method for the first step, we focus here on the optimization of the domain decomposition algorithm in this work.

In the domain decomposition algorithm, the main domain is divided into many subdomains, where the number of subdomains is usually chosen to be a power of 2. The data in a subdomain depend on other domains only through the inner boundaries between those domains. Within a subdomain, the nodal data have a

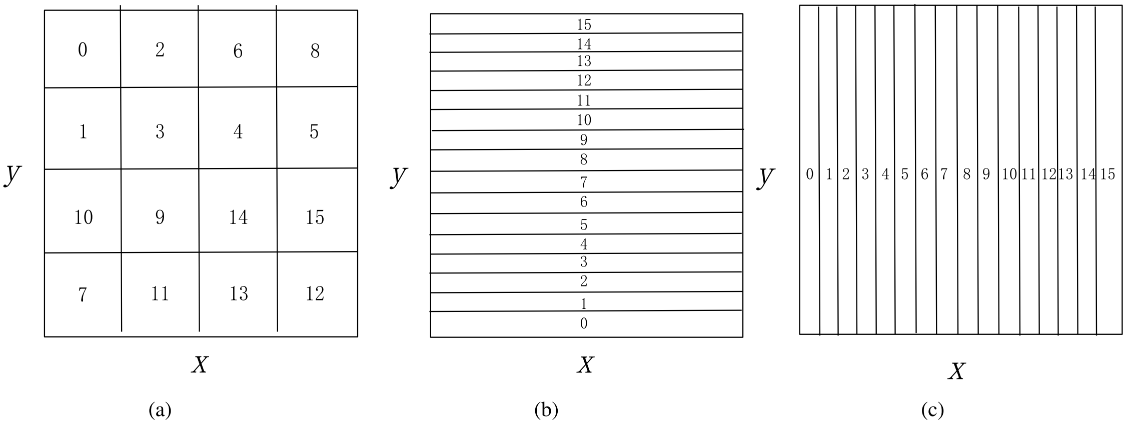


Fig. 1. Domain decomposition based on local line inversion.

well-defined interdependency. For the linear system (19), the solution consists of decomposing the matrix A into sub-blocks and implementing the line inversion algorithm in parallel. The key of the execution is to remove the interprocessor data dependency and carry out the local line inversion in each processor.

The standard ADI decomposition consists of collecting the block of data from each thread along the splitting direction into one thread to perform the line inversion. But this increases a lot the burden of the computation work. Thus, we stripped the original block of data of each thread first and then combined them along the splitting direction, in order to keep the computation complexity consistent with the explicit method in the original block of data. For instance, uniform Cartesian grids are initially decomposed into 4 × 4 subdomains with 16 processors in parallel (Fig. 1). The first step updates explicitly the conservative parameters from the MHD equations without thermal conduction in each subdomain by a single processor, and the rank of each processor is randomly arranged in Fig. 1a. At the second step, the domain is decomposed vertically into 16 subdomains by the increasing rank for the x-update and each processor uses the horizontal lines of each subdomain to solve Eq. (19) to obtain a new x_i or T_i (Fig. 1b). Based on those values, the domain is decomposed again into perpendicular subdomains for the y-update, and Eq. (19) is then solved in the vertical direction in parallel (Fig. 1c). Lastly, the updated values are rearranged into the corresponding subdomains in Fig. 1a before the next iteration in time. Note that the boundary node value of each horizontal domain or vertical domain is always taken from the physical boundary conditions. Because the heat conduction coefficient node value resulting from Eq. (17) is taken from previous data, it will take more iterations to converge if massively parallel processors are used.

Using the present parallelization algorithm, the whole linear tridiagonal equation system is divided into P subsystems with a matrix transformation on matrix A. Then P processors are assigned to deal with each subsystem in parallel. This method needs many transforms to take into account the effect of the internal boundary node, which increases a little the communication between processors. However, a comparison between the iteration time and the communication time, through our numerical tests for a 1000 × 1000 grid, indicates that the time spent on communicating the data is relatively short compared to that for solving the associated subsystems. In fact, the communication of inner boundaries between parallel nodes is also required in explicit solvers such as pure sub-time stepping or super-time stepping methods when performing parallel calculations. Therefore, our parallel algorithm is considerably faster to solve the linear system (19).

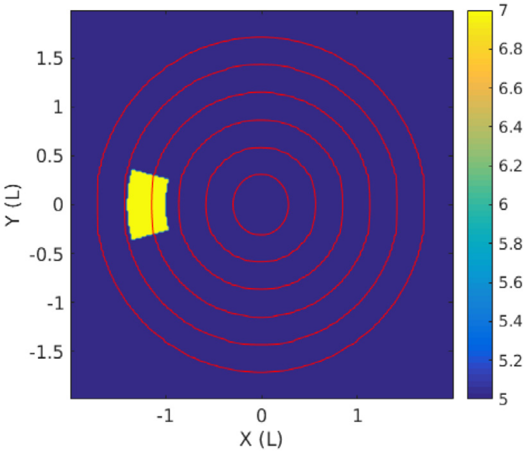


Fig. 2. Initial log_{10} temperature distribution and magnetic field lines.

3. Numerical tests

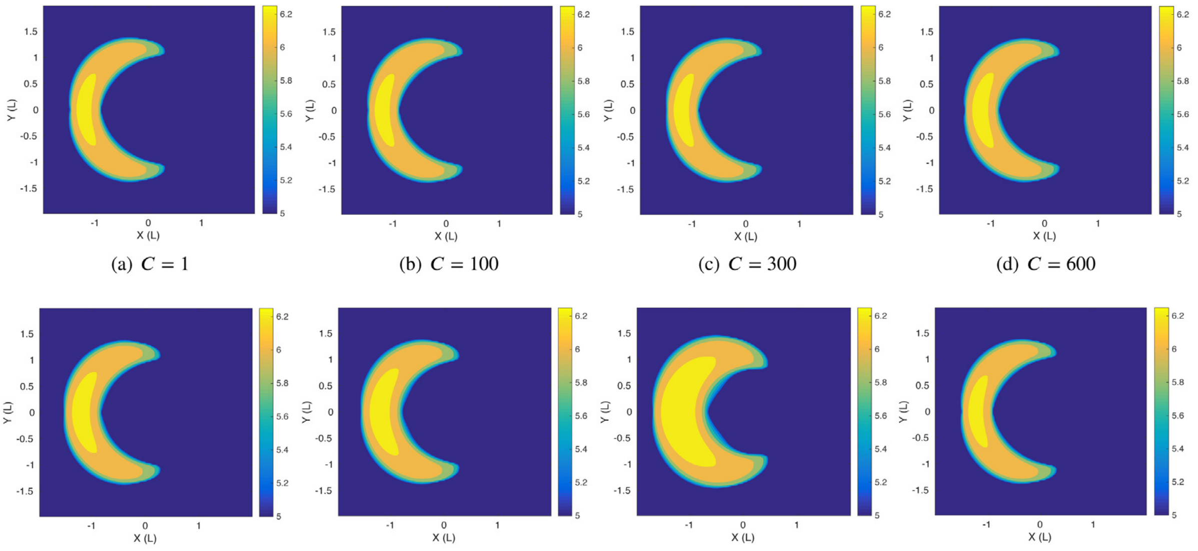
In this section, we study exhaustively two tests for our new algorithm.

3.1. Diffusion in a static ring

Considering a magnetostatic state of the low Corona, a hot patch is located in the fixed circular magnetic field lines. The problem is equivalent to solving only the thermal conduction equation (9). To be consistent with parameters in the realistic corona, we reconsider Eq. (9) in SI units. This is a crucial test to check monotonicity properties of the anisotropic diffusion method, because field lines make all possible angles with respect to the Cartesian grid. For the steady regime, the temperature is expected to be uniform along each magnetic field line in the ring as $t \rightarrow \infty$. We have given a characteristic length $L = 10^5$ m, so that the computational domain is $[-2L, 2L] \times [-2L, 2L]$ in 2D Cartesian coordinates. The initial temperature distribution is

$$\begin{aligned} T_0 &= 10^7 \text{ K} \quad \text{if} \quad L < r < 1.4L \quad \text{and} \quad \frac{11}{12}\pi < \theta < \frac{13}{12}\pi, \\ &= 10^5 \text{ K} \quad \text{else}, \end{aligned} \tag{28}$$

where $r = \sqrt{x^2+y^2}$ and $\tan\theta = y/x$, and $\hat{B}_x = -y/\sqrt{x^2+y^2}$, $\hat{B}_y = x/\sqrt{x^2+y^2}$ (see Fig. 2). The nonlinear diffusivity is $\kappa(T) = 1 \times 10^{-9} T^{5/2}$ (measured in SI units). The magnetic field and the conduction vanish outside $r = 2L$. Adiabatic boundary conditions



(a) $C = 1$ (b) $C = 100$ (c) $C = 300$ (d) $C = 600$

(e) $C = 1000$ (f) $C = 1500$ (g) $C = 2500$ (h) explicit

Fig. 3. *Log*$_{10}$ temperature contour plots at $t = 3000$ s for evaluating only the anisotropic diffusion scheme using a 128 × 128 grid. Different CFL numbers C were used to yield the Δt in Eqs. (12) and (13). Additionally, the temperature plot with fully explicit method is given for comparison.

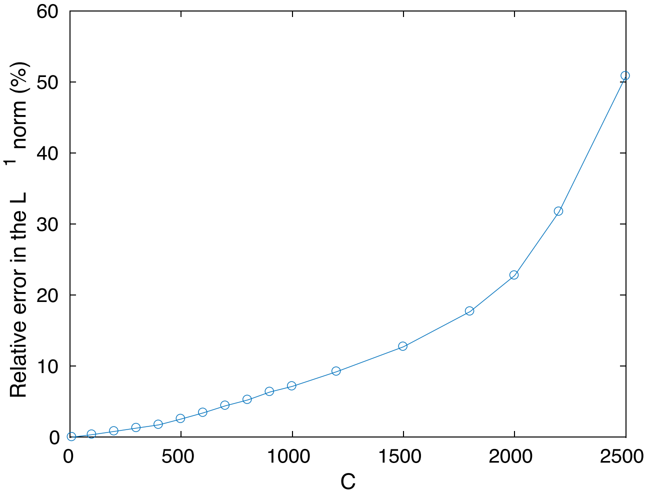


Fig. 4. The relative error defined in Eq. (30) as a function of the CFL number C.

($\partial T/\partial x = 0$ at boundaries in the x-direction, $\partial T/\partial y = 0$ at boundaries in the y-direction) are utilized for the temperature.

The CFL condition predicts an explicit timestep with

$$\Delta t_{CFL} = \Delta x^2/4\kappa_{\max}, \qquad (29)$$

measured in seconds. Despite the fact that our method is unconditionally stable in 2D, a very large Δt is still not appropriate because of numerical errors. Hence, we want to figure out roughly the maximum increase in time step with respect to a CFL number C, so that $\Delta t = C\Delta t_{CFL}$, without degrading the solution compared with the explicit method. A 128 × 128 mesh is used, and then we have $\Delta t_{CFL} \approx 7.72 \times 10^{-3}$ s according to Eq. (29). Fig. 3 shows the logarithm temperature at $t = 3000$ s solved by Eqs. (12) and (13) with different CFL numbers (C) as well as the solution solved by the fully explicit method for comparison. We stress that the fully explicit methods utilized the same slope limiters as the new solver, in order to exclude the effect of limiters. As expected, our method is unconditionally stable for $C \geq 1$, but the solution degrades as C increases. One can find that the solution for $C = 600$ is still very similar with the explicit solution. A very large time step, such as $C = 2500$, leads to a significant numerical diffusion in the perpendicular direction to the local magnetic field (see Fig. 3g) as well as the enlargement of the temperature extrema region. In order to determine the largest possible value for C so that the solution does not deteriorate too much, we compute also the relative error at $t = 3000$ s as

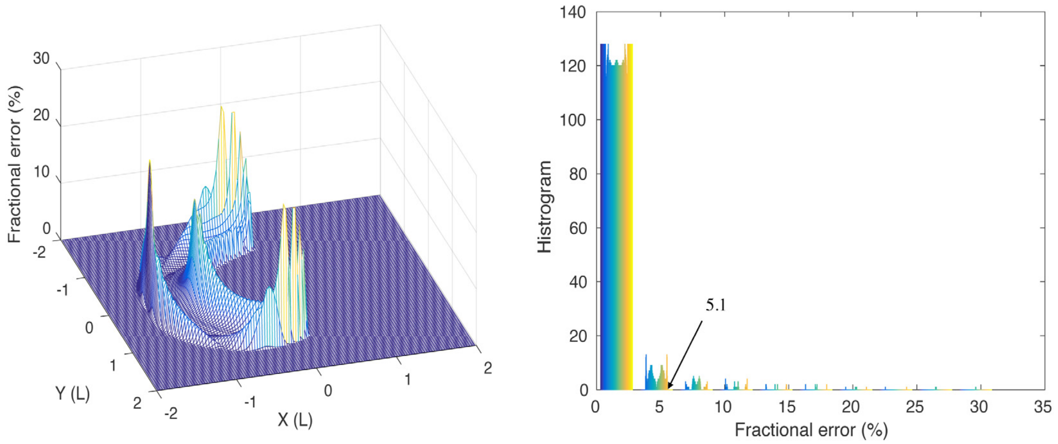
$$err = \frac{\|T_C(x,y) - T_{ex}(x,y)\|_{L^1}}{\|T_{ex}(x,y)\|_{L^1}}, \qquad (30)$$

where T_C is the solution obtained by our method with a CFL number C, T_{ex} is the solution obtained by the fully explicit scheme, and $\|\cdot\|_{L^1}$ represents the norm in the L^1 space. As shown in Fig. 4a, the corresponding error in the L^1 space is an exponentially monotonically increasing function with respect to the CFL number, and the circles indicate the values of $C = 10, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1200, 1500, 1800, 2000, 2200, 2500$.
The relative error varies from 0.0085% at $C = 10$ to 50.8% at $C = 2500$. In our experiments, when the value of C approaches to 3000, temperature profiles close to zero begin to appear. Clearly, the solution for C above 2500 may not be reliable due to the large error. To maintain both the satisfiable computation efficiency and the high numerical accuracy of simulations, a compromise between them should be made on the choice of the value C. As a guide, errors below 3.5% would normally be considered acceptable for nonlinear simulations. This figure shows an increase in time step of 600 compared to the fully explicit method for the Spitzer kind of thermal conduction model, with a relative error of 3.42%. The temperature tends to be steady at latter times, and the relative error decreases as time goes for a fixed constant C. Since the conductivity varies with temperature, the prediction of Eq. (17) validates better for a smooth evolution, but delays somewhat the fast dynamics in our experiments.

In particular, taking $C = 600$, we computed the fractional error distribution as

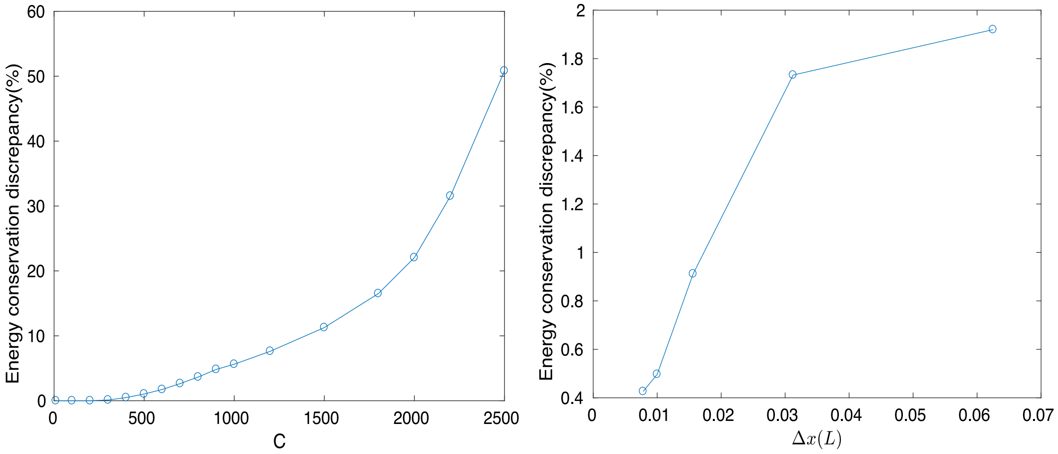
$$err_f = \frac{|T_C(x,y) - T_{ex}(x,y)|}{\|T_{ex}(x,y)\|_{L^\infty}}, \qquad (31)$$

where $\|\cdot\|_{L^\infty}$ is the L^∞ norm. As shown in Fig. 5a, the largest values locate near the edge of the hot region because of the discontinuous initial condition. If excluding the values on the edge, one can find the maximum fractional error of $\sim$ 5.1% inside the hot region according to the histogram of the fractional error in Fig. 5b.



(a) (b)

Fig. 5. The fractional error distribution (a) and the associated histogram (b).



(a) (b)

Fig. 6. The energy conservation discrepancy as a function of (a) the CFL number C and (b) the grid resolution Δx.

It is expected that the energy conservation holds for any time step Δt. So, the conservation law was checked by integrating the temperature profiles in the entire simulation domain. One can find in Fig. 6a that the energy conservation discrepancy between $t = 0$ s and $t = 3000$ s increases with the CFL number C. And that is 1.73% at $C = 600$. With a large CFL number $C > 600$, the discrepancy becomes unacceptable, and that is related to the numerical perpendicular diffusion coming from the prediction of the temperature in Eq. (17). In our experiments, we observed that some fluctuations happened around the areas of steep temperature gradients initially at first several time-steps, which contribute the main part of the numerical error responsible for the failure of the energy conservation. If a smoother initial condition is considered, the associated discrepancy could be much smaller. Also, we plot the conservation discrepancy as a function of the grid resolution at $C = 600$ in Fig. 6b, showing that using high resolution grids suppresses greatly the error. All the analyses above approve that the increase in time step of 600 agrees well with the explicit methods and the extra numerical errors can be minor.

The CFL number is expected to be of the same order (C) for all resolutions, and this statement is confirmed numerically by our experiments. Fig. 7 shows the logarithm temperature contour at $t = 3000$ s for different grid resolutions with a fixed CFL number ($C = 600$). Clearly, a higher grid resolution (see Fig. 7d) preserves more accurately the temperature extrema and reduces the perpendicular numerical diffusion for the fixed CFL number, while a lower grid resolution generates a less accurate temperature profile and the parallel diffusion seems to be suppressed (see Fig. 7a). In our work, the temperature is well bounded by the initial minimum temperature ($T_{min} = 10^5$ K) for the CFL number from $C = 1$ to $C = 2500$ or for the grid resolution from $n = 32$ to $n = 512$; the relative discrepancy is under 10^{-11}, which means that the monotonicity properties of the scheme is perfectly preserved.

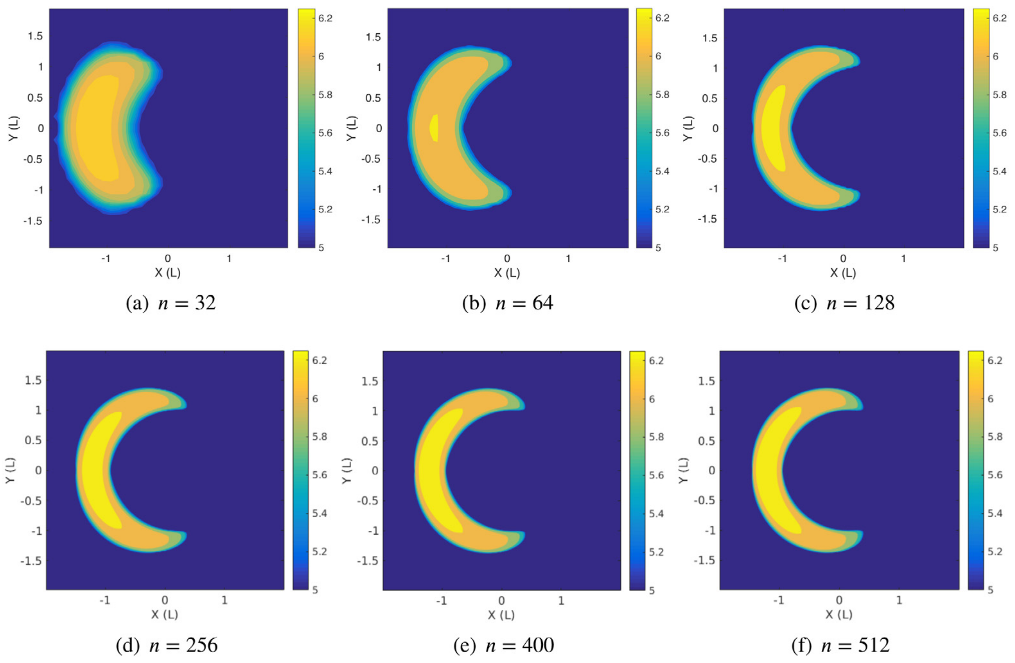
Despite of the non-physical perpendicular diffusion in the model, a rough estimate of the time averaged numerical perpendicular diffusion can be obtained by adding a numerical term $\kappa_{\perp,num}$ in Eq. (9) as follows:

$$\frac{\partial T}{\partial t} = \nabla \cdot \left(\kappa(T)(\nabla T \cdot \hat{\mathbf{B}})\hat{\mathbf{B}} + \kappa_{\perp,num}(\nabla T - (\nabla T \cdot \hat{\mathbf{B}})\hat{\mathbf{B}}) \right) \tag{32}$$

Analytically, the temperature diffuses only in the ring. However, because of the finite numerical diffusion in the perpendicular direction, it spreads out of the ring. Thus, $\kappa_{\perp,num}$ is given by

$$\kappa_{\perp,num} = \frac{\int (T_f - T_i) dV}{\int dt (\int \nabla^2 T dV)}, \tag{33}$$

where the space integral is taken over the region out of the hot ring ($L < r < 1.4L$), the time integral is taken from $t = 0$ s to



(a) $n = 32$ (b) $n = 64$ (c) $n = 128$

(d) $n = 256$ (e) $n = 400$ (f) $n = 512$

Fig. 7. Log_{10} temperature contour plots at $t = 3000$ s solved by our method on a $n \times n$ grid for a fixed CFL number ($C = 600$).

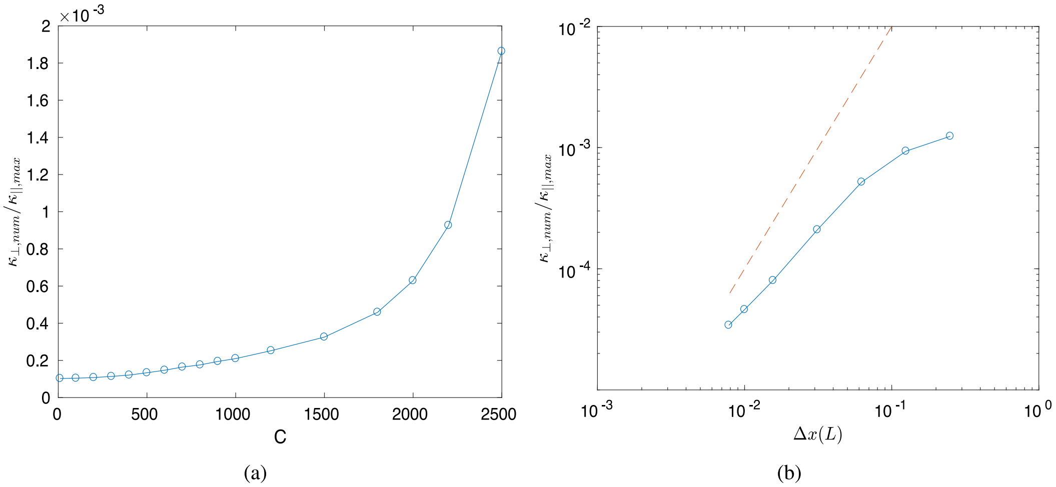


Fig. 8. The ratio of perpendicular (numerical) to the initial maximum parallel diffusivity (circles) at $t = 3000$ s as a function of (a) the CFL number C on the 128×128 mesh; (b) the grid size with a fixed $C = 600$.

$t = 3000$ s, and T_i and T_f are the initial and the final temperature distributions accordingly. Then we normalize the result by the initial maximum parallel diffusivity as $\kappa_{\parallel,max} = \kappa_{T_{0,max}}$ to obtain the ratio of perpendicular (numerical) to parallel diffusivity. Fig. 8a shows the ratio $\kappa_{\perp,num}/\kappa_{\parallel,max}$ as a function of the CFL number C, and the plot is monotonically increasing, as confirmed in Fig. 3. Because of the temperature-dependent conductivity, a larger CFL number can lead to a stronger perpendicular numerical diffusion. Meanwhile, the spacial resolution introduces a significant effect on the perpendicular diffusivity at a fixed CFL number ($C = 600$) as shown in Fig. 8b. The circles represent the grid size of $N = 16, 32, 64, 128, 256, 400, 512$ and the red dashed line indicates a second order convergence. Unlike the constant diffusivity cases of Sharma and Hammett (2011), the convergence rate of our results is between $O(\Delta x)$ and $O(\Delta x^2)$. The reasonable explanation is that the prediction of Eq. (17) for T^{n+1} brings extra numerical errors to depress the second-order accurate convergence. Since both the CFL number and the grid size affect the perpendicular numerical diffusion, the resulting plot deviates from the expected second order convergence.

In fact, the CFL number should be $\lesssim (l/\Delta x)^2$ for practical applications, where l is the characteristic length. Depending on how accurate we want the temperature to be, we can adjust the CFL number to obtain a better computation efficiency (Fig. 3). Specifically, for applications in corona, the increase in time step by $C = 1500$ is also acceptable without disrupting the general diffusion process, but the error on the temperature profile can reach $> 10\%$. A compromise between the accuracy and the computation efficiency should be made before launching the simulation. However, an increase in time step of ~ 600 can be easily achieved by our method to meet the accuracy requirement.

3.2. Plasmoid instability in a corona current sheet

Plasmoid instability is an important topic for energy dissipation in solar physics. For a common application in the corona,

Table 2
Summary of normalization units used for diverse thermal conduction models using different solvers.

Case	T_N	L_N	B_N	ρ_N	S	$\kappa_\parallel$	Solver
	(10^7 K)	(10^7 m)	(0.005 T)	(1.204×10^{-10} kg m^{-3})			for thermal conduction
A	1	1	1	1	$\frac{4}{6} \times 10^6 T_0^{3/2}$	0	No
B	1	1	1	1	$\frac{4}{6} \times 10^6 T_0^{3/2}$	$\kappa_\parallel$ of Eq. (8)	Fully explicit
C	1	1	1	1	$\frac{4}{6} \times 10^6 T_0^{3/2}$	$\kappa_\parallel$ of Eq. (8)	Our new scheme
D	1	1	1	1	$\frac{4}{6} \times 10^7 T^{3/2}$	0	No
E	1	1	1	1	$\frac{4}{6} \times 10^7 T^{3/2}$	$\kappa_\parallel$ of Eq. (8)	Fully explicit
F	1	1	1	1	$\frac{4}{6} \times 10^7 T^{3/2}$	$\kappa_\parallel$ of Eq. (8)	Our new scheme
G	1	1	0.2	0.04	$\frac{4}{6} \times 10^7 T^{3/2}$	$\kappa_\parallel$ of Eq. (8)	Our new scheme

we present MHD simulations of a reconnecting current sheet as implemented on ATHENA 4.2 (Gardiner and Stone, 2005, 2008; Stone and Gardiner, 2009) including our new method to solve the thermal conduction problem. When solving such a complex MHD problem, the question that how would this perform around shocks or discontinuities needs to be answered. The plasmoid test starts to investigate this issue by following the work of Ni et al. (2012), except that our initials conditions are more realistic. Considering the dimensionless MHD equations (1)–(7), we set the simulation domain as $(x, y) \in [0, 1] \times [0, 4]$. We choose a Harris current sheet as the initial condition for the magnetic field:

$$B_{y0} = b_0 \tanh\left(\frac{x - 0.5}{\lambda}\right), \quad B_{x0} = 0, \tag{34}$$

where λ is the initial width of the current sheet set to 0.05 and $b_0 = 1$. To make the tearing instabilities happen, the current sheet must be thin enough so that the condition can be satisfied: $2/\lambda(1/k\lambda - k\lambda) > 0$, where $k = 2\pi/l_y$ is the wave number of the initial perturbations and l_y stands for the length in the y-direction ($= 4$ for this case). Deriving the initial equilibrium from Eq. (2) with $\mathbf{v}_0 = 0$, we have

$$\nabla \cdot (p_0 \mathbf{I}) = -\nabla \cdot \left[\frac{1}{2}|\mathbf{B}_0|^2 \mathbf{I} - \mathbf{B}_0 \mathbf{B}_0\right]. \tag{35}$$

Substituting $\mathbf{B}_0 = B_{y0}\hat{\mathbf{y}}$, where $\hat{\mathbf{y}}$ is the unit vector in the y-direction, into the above equation, one can get the initial equilibrium gas pressure as

$$p_0 = -\frac{1}{2}B_{y0}^2 + C_0, \tag{36}$$

with C_0 a constant to be determined later. Since $|B_{y0}| = 1$ at $x = 0, 1$ from Eq. (34), and the plasma β is given by $\beta = p/2B^2$, C_0 is calculated as $C_0 = \beta_0 + 1/2$, where β_0 is the initial plasma β at the x-boundaries. Then the initial pressure reads

$$p_0 = \frac{1 + \beta_0 - B_{y0}^2}{2}, \tag{37}$$

and the initial equilibrium total energy is

$$e_0 = p_0/(\gamma - 1) + B_{y0}^2/2. \tag{38}$$

Supposing an isothermal atmosphere and the ideal gas law, we derive the initial equilibrium mass density and temperature as

$$\rho_0 = p_0/T_0 = \frac{1 + \beta_0 - B_{y0}^2}{\beta_0}, \quad T_0 = \frac{\beta_0}{2}. \tag{39}$$

Additionally, we add small initial perturbations on the magnetic field to trigger plasmoid instabilities in the current sheet in the form

$$b_{x1} = -\varepsilon \cdot 0.5\sin(\pi x)\cos(\pi y/2), \tag{40}$$

$$b_{y1} = \varepsilon \cdot \cos(\pi x)\sin(\pi y/2). \tag{41}$$

The amplification factor ε is set to 0.05 in our simulations. As described in Ni et al. (2012), this kind of perturbations can yield a large primary magnetic island, and eventually many secondary plasmoids due to tearing instabilities if the Lundquist number is above 10^4 (Shen et al., 2011). Reflecting boundary conditions are used in the x-direction and periodic boundary conditions are used in the y-direction, respectively.

Seven groups of tests with different configurations are discussed in this paper. The normalization units for temperature, length, magnetic field and density are set to $T_N = 10^7$ K, $L_N = 10^7$ m, $B_N = 0.005$ T, and $\rho_N = 1.204 \times 10^{-10}$ kg m^{-3}, respectively. According to the definition of c_1 in Eq. (8), the choice of the normalization units affects the significance of thermal conduction in these cases. Table 2 summaries the normalization units used for diverse thermal conduction models using different solvers in order to get a general comparison in terms of the accuracy and efficiency. The initial plasma β_0 is set to 0.2 for all groups, thus we have $T_0 = \beta_0/2 = 0.1$. Notice that the constant Lundquist number is only related to the initial temperature $S = 4/6 \times 10^6 T_0^{3/2}$ for A, B and C, so that S is the order of 10^4 and no secondary magnetic islands appear. For the others, we choose $S = 4/6 \times 10^7 T^{3/2}$ in order to make the Lundquist number sufficiently high to yield the secondary occurrence plasmoid instabilities in the current sheet.

Accordingly, the initial temperature T_0 in the dimensional space is $T_I = 0.1 \cdot T_N = 10^6$ K. The magnetic field is of the order of 50G and the mass density is of order of 1.204×10^{-10} kg m^{-3}. The configuration for A–F corresponds more likely to the low corona, and we add also the Case G with the typical $B_N = 0.001$ T and $\rho_N = 2.883 \times 10^{-12}$ kg m^{-3} for the higher corona. As a result, the value c_1 in Eq. (8) for Case G is greater than that in Case $A - F$. Notice that, among all the cases above, the resulting characteristic temperature is $T_N = 10^7$ K, the Alfvén speed is $v_{AN} = 406.6$ km/s and the Alfvén time is $t_A = 24.6$ s.

The simulations are implemented on a uniform Cartesian grid of 320 × 1280 using 64 CPUs for parallelization. To ensure that this resolution is appropriate for the plasmoid instability study, we resimulated the case D with the grids of 160 × 640 and 640 × 2560 for comparison with our choice. The reconnection rates versus time are very similar in general for the meshes of 320 × 1280 and 640 × 2560. We observe that the higher the resolution is, the earlier the plasmoid instability starts, but their peak values are almost the same. In other words, the physical resistivity dominates in the overall evolution. In order to have a faster convergence, we only use the grid of 320 × 1280 here. Lastly, the simulation time is $40t_A$ for all groups. Fig. 9 shows the temperature profiles at times $t = 6, 10, 20, 36t_A$ for Case A–C. A large primary island is formed and the current sheet shrinks as time goes, but no secondary island appears. In the first row of the figure, Case A establishes a reference without thermal conduction for the other two cases. The second row for Case B, which involves the Spitzer thermal conduction model, is the solution obtained

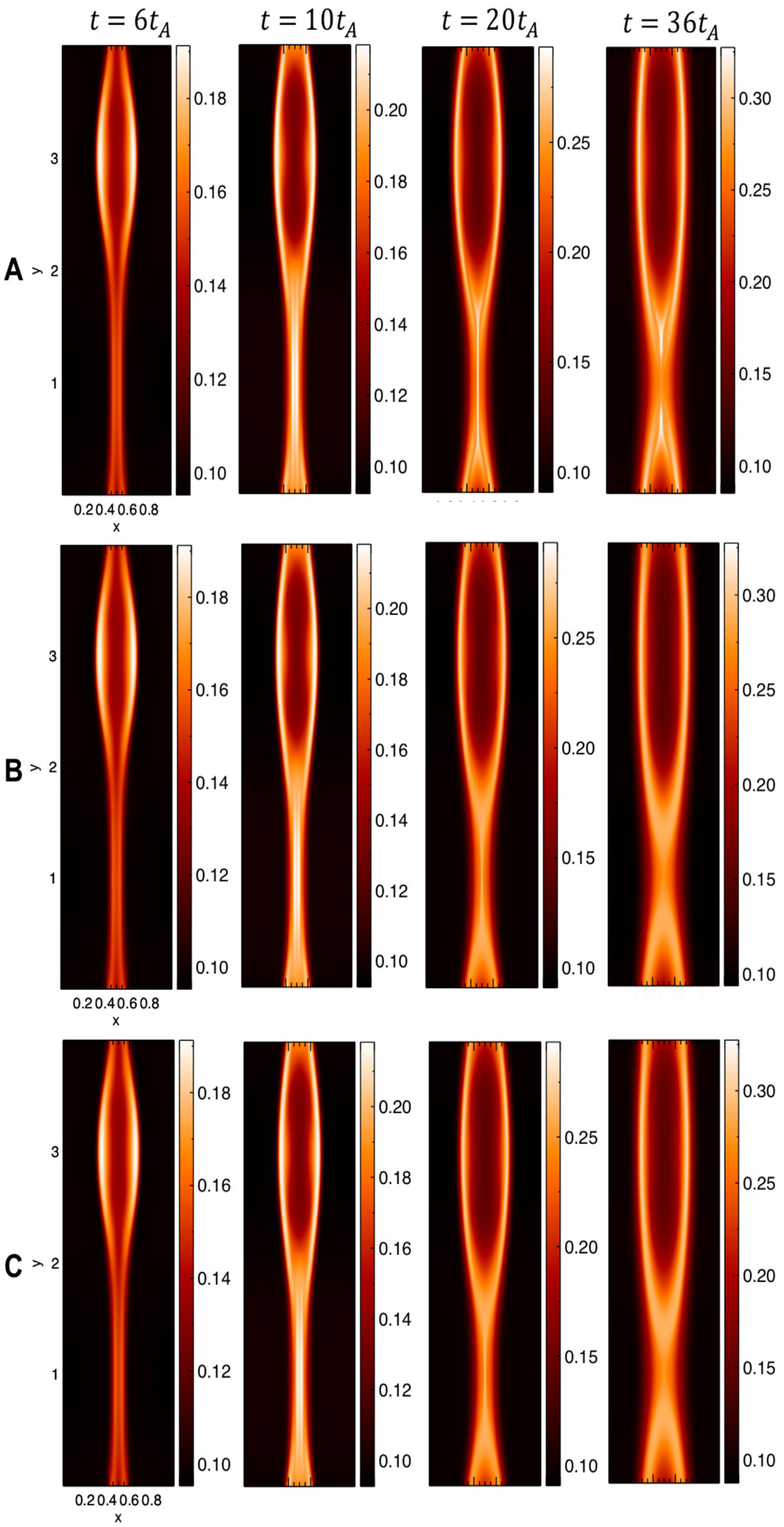


Fig. 9. Temperature plots at times $t = 6, 10, 20, 36t_A$ for Case A (1st row), B (2nd row) and C (3rd row).

by the fully explicit method. At each time iteration, the time step takes the minimum between Δt_{MHD} provided by the MHD part (first step) and Δt_{cond} provided by the thermal conduction part (second step), so that $\Delta t = \min(\Delta t_{MHD}, \Delta t_{cond})$. The number of the time steps is 725 486 and the total CPU time is about 110.9 h. Because the maximum temperature can reach around 0.3 at latter times, so that the corresponding $\Delta t = 3.49 \times 10^{-5} t_A$. The third row for Case C is the solution provided by our scheme with a fixed $C = 600$, so that the number of time steps is 39 917 and the CPU time is 6.1 h (18 times faster than B). Δt_{MHD} is slightly smaller than Δt_{cond} at the beginning, and the acceleration ratio $\Delta t_{MHD}/\Delta t_{cond}$ can reach $\sim$ 12 at later times. The fact is that

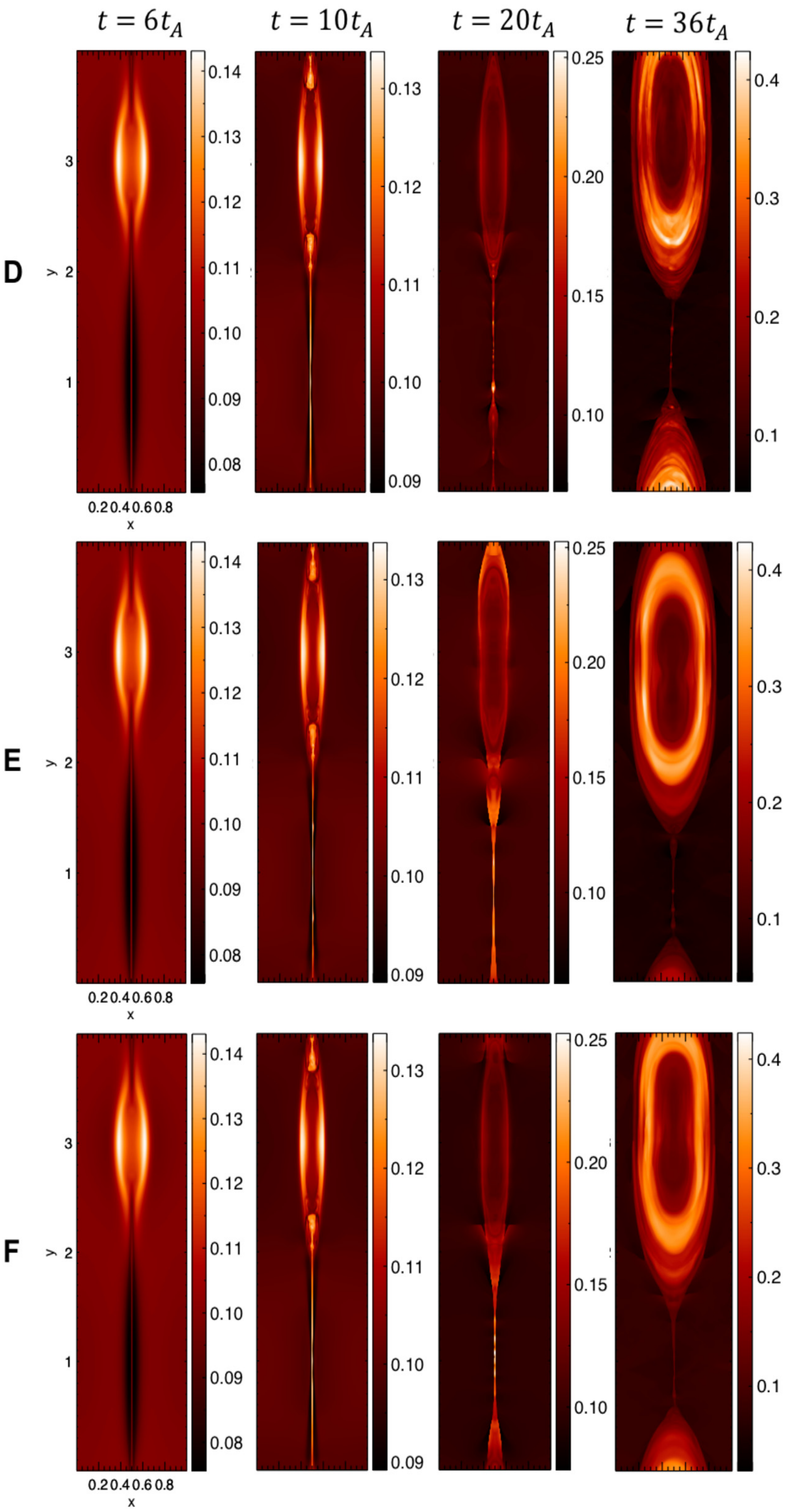
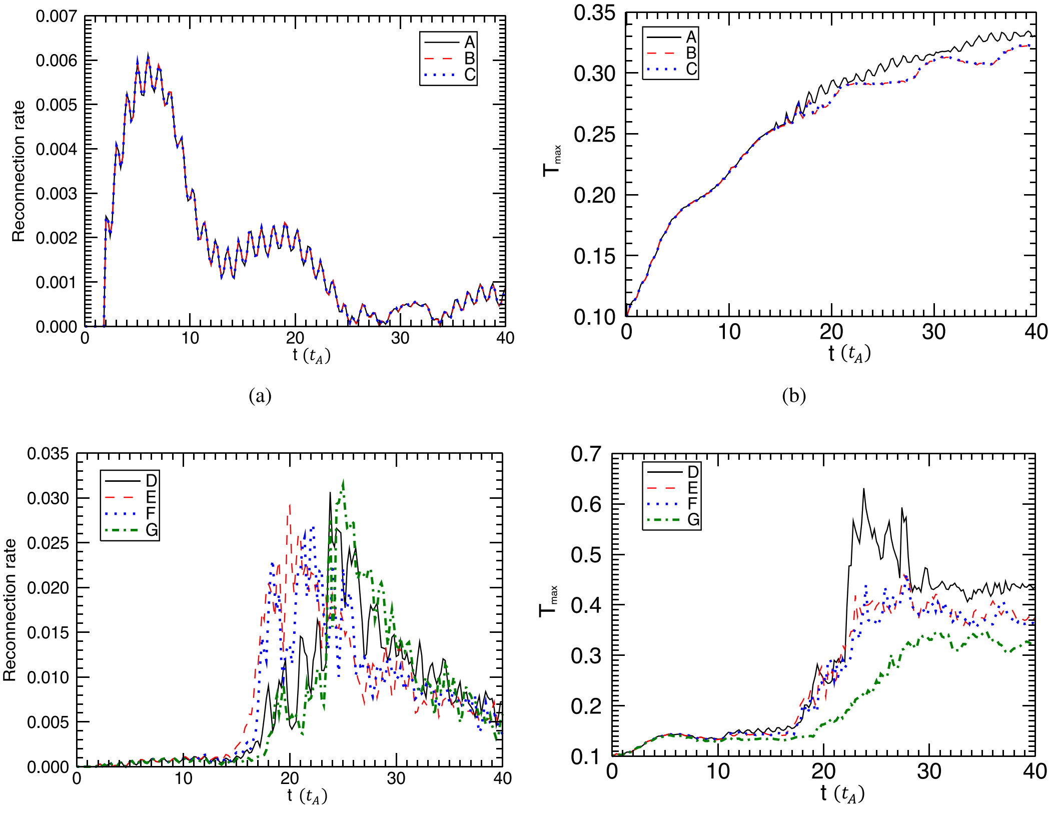


Fig. 10. Temperature plots at times $t = 6, 10, 20, 36t_A$ for Case D (1st row), E (2nd row) and F (3rd row).

thermal conduction results in the smooth spreading temperature as expected, the profiles of B and C are identical at all times.

Similarly, the results of Case D–F are presented in Fig. 10. And, Case D gives the reference without thermal conduction for the other two cases in presence of plasmoid instabilities. Case E and F is the solution provided by the fully explicit method and our scheme, respectively. The number time steps of Case E is 966 700 taking 144.4 h, while that of Case F is 51 060 taking 7.96 h. Indeed, the later is nearly 19 times faster, and the maximum $\Delta t_{MHD}/\Delta t_{cond} = 19.3$ is attained. Particularly, for Case G with the larger value of c_1 in Eq. (8), we have a better maximum acceleration ratio $\Delta t_{MHD}/\Delta t_{cond} =\sim 220$. Clearly, Case E and



(a)

(b)

(c)

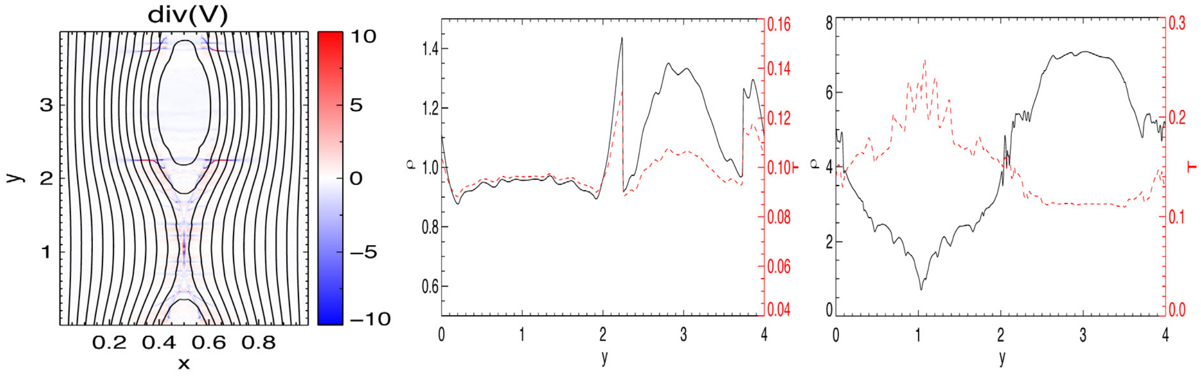
(d)

Fig. 11. Reconnection rate (left column) and maximum temperature (right column) versus time for Case A–G.

F are identical at $t = 6, 10t_A$, but they deviate slightly from each other at $t = 20, 36t_A$ because of the fast dynamics of plasmoid instabilities. As mentioned in Section 3.1, the use of larger time steps used by our scheme can delay the generation of the small-scale plasmoids.

To ensure that our scheme does not alter the physical processes in comparison with fully explicit methods, we calculate the time-dependent reconnection rate as $\gamma(t) = \partial(\psi_X(t) - \psi_O(t))/\partial t$, where ψ_X and ψ_O are magnetic flux functions at the main reconnection X-point and the primary O-point. In this way, $\gamma(t)$ is the global reconnection rate over the entire current sheet. Fig. 11 presents the temporal reconnection rates and maximum temperature for Case A–G. In Fig. 11ab, thermal conduction has no effect on reconnection rates in absence of plasmoid instabilities, but lowers the temperature amplitude. However, B and C are identical at all times. In Fig. 11c, secondary plasmoid instabilities start earlier for both E and F than Case D because of thermal conduction, and eventually result in faster reconnection rates. We observe that the rise in the reconnection rate of Case F provided by our scheme is slightly delayed compared with that of Case E from the occurrence of secondary plasmoid instabilities, but their dynamics are very similar in general. Another comparison can be found in the temperature profiles in Fig. 11d. The results for E and F are the same with a little shifting, and their temperatures are smaller than Case D as expected. Also, we have investigated the profiles around shocks and discontinuities from the solution obtained by our scheme. In here, we focus on the above scheme property at discontinuity layers and employ the same conduction coefficient as shown in Eq. (8) instead of investigating more microscale physical effects of electron collision processes between pre-shocked and post-shocked plasma. Fig. 12a shows the divergence of fluid velocity and the magnetic field at $t = 20t_A$ for Case F. We observe that the shocks locate mainly inside the current sheet and around the large plasmoid. Then the density and temperature along the line $x = 0.35$ across the shock are plotted in Fig. 12b. Our scheme preserves well the jump of the density at the shock front near $y = 2.2$ as well as the temperature. Similarly, Fig. 12c presents the associated profiles along the current sheet $x = 0.5$, and the discontinuities caused by small-scale magnetic islands are accurately captured. Overall, our scheme works well for the current sheet tests, where there is sudden heating. But it does delay slightly the fast dynamics in simulations, such as plasmoid instability.

Our scheme also has some advantages in comparison with the popular Super TimeStepping method (STS). Technically, the STS method for a s^2 larger time step requires s extra sub-loops at each time step, with s an even integer. For a $N \times N$ grid, a STS method recognizes a computation complexity of $O(sN^2)$, while our scheme can get to $O(N^2)$ as same as the full explicit methods, when having $\Delta t_{MHD}/\Delta t_{cond} \approx s^2$. Thus, as s becomes larger in an extremely high-temperature environment or fine spatial resolution simulations, our scheme can work much better than the STS method. In terms of the parallel efficiency, we performed a series of tests to study the extendibility of our method, because the use of the semi-implicit techniques (see Fig. 1) requires a massive data communication between processors. This evaluation consists of two standard indicators in general. The first indicator (blue line in Fig. 13) is the strong extendibility (SE) , in which we resimulated Case F with a grid of 640 × 2560 using 32, 64, and 128 processors on Milkyway-2 supercomputer centre in Guangzhou, respectively. In fact, each node on the cluster possesses 24 CPU cores and we want to compare the computation time from multiple nodes. Taking the 32 processors job as the reference, it is 2.05 times faster using 64 processors and the associated efficiency of each thread is 100%. Similarly, the efficiency for 128 processors is about 85.3%. On the other hand,



(a) (b) (c)

Fig. 12. (a) Divergence of fluid velocity and magnetic field at $t = 20t_A$ for Case F (b) The density and temperature along the line $x = 0.35$ across the shock (c) The density and temperature along the central line $x = 0.5$.

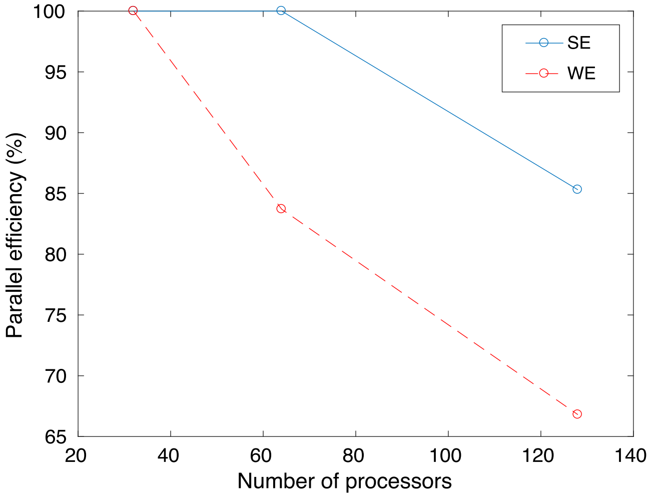


Fig. 13. The parallel efficiency as a function of the number of processors via the strong (blue) and weak (red) extendibility studies.

we resimulated Group F with three sizes of grids using 32, 64 and 128 processors. The principle of the second indicator (red line in Fig. 13) is to keep the quantity of data calculated by each thread unchanged in three cases in order to estimate the total time consumed by the data communication between processors, namely, the weak extendibility (WE). We show the results using the grids of 320 × 1280, 640 × 1280, and 640 × 2560 for $n_t = 49013$ time steps here. Note that $n_t = 49013$ is the total time step number required for a simulation time of $40t_A$ in Group F considering the 320 × 1280 mesh. Again, the 32 processors job for the grid of 320 × 1280 takes totally 7.41×10^3 s, and the computation time for the 64 processors job of 640 × 1280 is 8.84×10^3 s. Thus, the acceleration ratio is $2 \times 7.41 \times 10^3/8.84 \times 10^3 \approx 1.675$ and the corresponding efficiency of each thread is 83.7%. Similarly, we have the efficiency of 66.8% for 128 processors. Both SE and WE studies are very stable. Our parallelism seems not great, because our parallelization algorithm will increase by a factor of a few the burden of the data communication between processors which eventually weakens the acceleration efficiency for each thread, but its general performance for thinner grids is still satisfactory.

However, our scheme is only first order accurate in time. If cooperating with MHD codes which are of higher order accurate in time, this method will degrade the global accuracy to the first order as well. It is straightforward to expand the present method into a higher order in time, but it will increase the complexity of solving the linear systems (with large sparse matrix) and the data communication work for each thread. Substantial improvements in terms of the anisotropic diffusion solver and the parallelization algorithm can be made in the future.

4. Discussion and conclusions

Anisotropic thermal conduction is important for solar applications to predict realistic hot structures in numerical experiments. In this work, we present a semi-implicit solver for the nonlinear anisotropic diffusion problem within a numerical MHD scheme as well as the associated parallelization strategy. This solver is linearly stable for large timesteps and easily realizes an increase in time step of ∼ 600 compared with fully explicit methods for the problem in Section 3.1. Its accuracy is only first order in time and second order in space in present work, and the use of slope limiters for large temperature gradients can preserve well the temperature extremes (monotonicity). The polynomial approach for prediction to the nonlinear conductivity at the posteriori time successfully linearizes the system without causing instabilities. Benefitting from the directional split, our scheme has two independent tridiagonal matrices to solve, whose complexity is $O(2N)$ on a $N \times N$ mesh instead of $O(N^2)$ for iterative implicit methods (Atkinson, 1989).

The parallelization strategy is based on the domain decomposition method, and we decomposed the domain according to the splitting direction for each implicit update. The advantage of this action is that the quantity of data available for computing in each processor remains unchanged in both explicit updates of MHD parts and implicit updates of thermal conduction, but it does increase the data communication work somewhat.

Two numerical tests are exhaustively studied in this work. The first one shows that a large timestep leads to a greater numerical perpendicular diffusion and the convergence as the function of the spacial resolution stays between first order and second order accurate. It achieved easily an increase in time step of 600 with a relative error under 3.42%. The second one represents a more realistic case of Ni et al. (2012) using ATHENA4.2. We can find out that the evolution obtained show no difference in general, but the semi-implicit solver does delay slightly the generation of the secondary islands at later times. However, our scheme provides the correct reconnection rates as well as the temperature magnitudes, by which we are most interested. Also, our results confirm that thermal conduction does increase the reconnection rate in presence of plasmoid instabilities as reported by Ni et al. (2012). Moreover, we have studied the extendibility (strong, weak) of the parallelization algorithm, the efficiency for each processor turns out to be satisfactory. So far, this scheme has been working well for our present purposes, but there is still some progress to make in the future, such as using higher dimensional linear recurrences to solve the diatriangular linear system (Wang and Vanka, 1993). The use of non-uniform or AMR meshes is possible as well.

For most solar applications, thermal conduction is primarily along the magnetic field for hot plasmas and the Spitzer conductivity is used for highly ionized gases. Actually, our method can be applied to any kind of conductivity (constant, space-dependent, time-dependent, or temperature-dependent). Our scheme is perfectly monotonicity-preserving and also fast so that the conduction time-steps can be the same order of the MHD time-steps. It will be quite useful for our future applications to study the hot flare plasmas and the fine (turbulent) structure of the reconnection current sheet.

CRediT authorship contribution statement

J. Ye: Conceptualization, Methodology, Writing — original draft. **C. Shen:** Data curation.

Acknowledgments

J. Ye is grateful to J. C. Raymond for valuable discussions. The authors would like to thank the anonymous referees for the valuable and constructive comments and suggestions that helped improve this work significantly. This work was supported by NSFC, China grants 11603070, 11333007, U1631130, 11933009, CAS, China grants QYZDJ-SSWSLH012 and XDA15010900, the Strategic Priority Research Program of CAS, China with grant XDA17040507, and the grant associated with the project of the Group for Innovation of Yunnan Province 2018HC023, the Yunnan Ten-Thousand Talents Plan-Yunling Scholar Project as well as the Special Program for Applied Research on Super Computation of the NSFC-Guangdong Joint Fund (the second phase), China under Grant No. U1501501. C. Shen acknowledges the support by National Science Foundation of U.S. grants AGS-1358342, AGS-1723313, AGS-1723425 to the Smithsonian Astrophysical Observatory.

References

Atkinson, K.E., 1989. An Introduction to Numerical Analysis. John Wiley & Sons, Inc.

Balsara, D.S., Tilley, D.A., Howk, J.C., 2008. Simulating anisotropic thermal conduction in supernova remnants - I. Numerical methods. Mon. Not. R. Astron. Soc. 386, 627–641. doi:10.1111/j.1365-2966.2008.13085.x.

Bradshaw, S.J., Klimchuk, J.A., Reep, J.W., 2012. Astrophys. J. 758, 53. doi:10.1088/0004-637X/758/1/53.

Bruno, O.P., Jimenez, E., 2014. J. Fluids Eng. 136, 060904. doi:10.1115/1.4026868.

Ciaravella, A., Raymond, J.C., Li, J., Reiser, P., 2002. Astrophys. J. 575, 1116. doi:10.1086/341473.

Gardiner, T.A., Stone, J., 2005. An unspilt godunov method for ideal mhd via constraint transport. J. Comput. Phys. 205, 509. doi:10.1016/j.jcp.2004.11.016.

Gardiner, T.A., Stone, J.M., 2008. An unspilt godunov method for ideal mhd via constraint transport in three dimensions. J. Comput. Phys. 227, 4123. doi:10.1016/j.jcp.2007.12.017.

Kannan, R., Springel, V., Pakmor, R., Marinacci, F., 2016. Mon. Not. R. Astron. Soc. 458, 410. doi:10.1093/mnras/stw294.

Kannan, R., Vogelsberger, M., Pfrommer, C., Weinberger, R., Springel, V., Hernquist, L., Puchwein, E., Pakmor, R., 2017. Astrophys. J. Lett. 837, L18. doi:10.3847/2041-8213/aa624b.

van Leer, B.J., 1977. J. Comput. Phys. 23, 276.

Lemen, J.R., Title, A.M., Akin, D.J., Boerner, P.F., Chou, C., Drake, J.F., Duncan, D.W., Edwards, C.G., Friedlaender, F.M., Heyman, G.F., Hurlburt, N.E., Katz, N.L., Kushner, G.D., Levay, M., Lindgren, R.W., Mathur, D.P., McFeaters, E.L., Mitchell, S., Rehse, R.A., Schrijver, C.J., Springer, L.A., Stern, R.A., Tarbell, T.D., Wuelser, J.-P., Wolfson, C.J., Yanari, C., Bookbinder, J.A., Cheimets, P.N., Caldwell, D., Deluca, E.E., Gates, R., Golub, L., Park, S., Podgorski, W.A., Bush, R.I., Scherrer, P.H., Gummin, M.A., Smith, P., Auker, G., Jerram, P., Pool, P., Soufli, R., Windt, D.L., Beardsley, S., Clapp, M., Lang, J., Waltham, N., 2012. Sol. Phys. 275, 17. doi:10.1007/s11207-011-9776-8.

Leveque, R.J., 2002. Finite Volume Methods for Hyperbolic Problems. Cambridge University Press, Cambridge.

Liu, R., 2013. Mon. Not. R. Astron. Soc. 434, 1309. doi:10.1093/mnras/stt1090.

Liu, W., Chen, Q.R., Petrosian, V., 2013. Astrophys. J. 767, 168. doi:10.1088/0004-637X/767/2/168.

Meyer, C.D., Balsara, D.S., Aslam, T.D., 2012. Mon. Not. R. Astron. Soc. 422, 2102. doi:10.1111/j.1365-2966.2012.20744.x.

Ni, L., Roussev, I.I., Lin, J., Ziegler, U., 2012. Astrophys. J. 758, 20. doi:10.1088/0004-637X/758/1/20.

Peter, H., Tian, H., Curdt, W., Schmit, D., Innes, D., De Pontieu, B., Lemen, J., Title, A., Boerner, P., Hurlburt, N., Tarbell, T.D., Wuelser, J.P., Martínez-Sykora, J., Kleint, L., Golub, L., McKillop, S., Reeves, K.K., Saar, S., Testa, P., Kankelborg, C., Jaeggli, S., Carlsson, M., Hansteen, V., 2014. Science 346, 1255726. doi:10.1126/science.1255726.

Reeves, K.K., Golub, L., 2011. Astrophys. J. Lett. 727, L52. doi:10.1088/2041-8205/727/2/L52.

Sharma, P., Hammett, G.W., 2007. J. Comput. Phys. 227, 123. doi:10.1016/j.jcp.2007.07.026.

Sharma, P., Hammett, G.W., 2011. J. Comput. Phys. 230, 4899. doi:10.1016/j.jcp.2011.03.009.

Shen, C., Lin, J., Murphy, N.A., 2011. Astrophys. J. 737, 14. doi:10.1088/0004-637X/737/1/14.

Shimizu, T., 2015. Phys. Plasmas 22, 101207. doi:10.1063/1.4933056.

Spitzer, L., 1962. Physics of Fully Ionized Gases. Interscience, New York.

Stone, J.M., Gardiner, T.A., 2009. A simple unspilt godunov method for multidimensional mhd. New Astron. 14, 139. doi:10.1016/j.newast.2008.06.003.

Stone, J.M., Mihalas, D., Norman, M.L., 1992. Astrophys. J. Suppl. Ser. 80, 819–845.

Stone, J.M., Norman, M.L., 1992a. Astrophys. J. Suppl. Ser. 80, 753–790.

Stone, J.M., Norman, M.L., 1992b. Astrophys. J. Suppl. Ser. 80, 791–818.

Wang, M., Georgiadis, J.G., 1991. Numer. Heat Transfer B 20, 41.

Wang, M., Vanka, S.P., 1993. Numer. Heat Transfer B 24, 143. doi:10.1080/10407799308955886.

Witelski, T.P., Bowen, M., 2003. Appl. Numer. Math. 45, 331–351. doi:10.1016/S0168-9274(02)00194-0.

Ye, J., Shen, C.C., Raymond, J.C., Lin, J., Ziegler, U., 2019. Mon. Not. R. Astron. Soc. 482, 588. doi:10.1093/mnras/sty2716.

Ziegler, U., 2005. Comput. Phys. Comm. 170, 153. doi:10.1016/j.cpc.2005.04.002.

Ziegler, U., 2008. Comput. Phys. Comm. 179, 227. doi:10.1016/j.cpc.2008.02.017.

Ziegler, U., 2011. J. Comput. Phys. 230, 1035. doi:10.1016/j.jcp.2010.10.022.