ELSEVIER

Additive Schwarz-based fully coupled implicit methods for resistive Hall magnetohydrodynamic problems ☆

S. Ovtchinnikov [a], F. Dobrian [b,c], X.-C. Cai [a,*], D.E. Keyes [c]

[a] *Department of Computer Science, University of Colorado, Boulder, CO 80309, United States*
[b] *Department of Computer Science, Old Dominion University, Norfolk, VA 23529, United States*
[c] *Department of Applied Physics and Applied Mathematics, Columbia University, New York, NY 10027, United States*

Abstract

A parallel, fully coupled, nonlinearly implicit Newton–Krylov–Schwarz algorithm is proposed for the numerical simulation of a magnetic reconnection problem described by a system of resistive Hall magnetohydrodynamics equations in slab geometry. A key component of the algorithm is a restricted additive Schwarz preconditioner defined for problems with doubly periodic boundary conditions. We show numerically that with such a preconditioned nonlinearly implicit method the time step size is no longer constrained by the CFL number or the convergence of the Newton solver. We report the parallel performance of the algorithm and software on machines with thousands of processors.
© 2007 Elsevier Inc. All rights reserved.

Keywords: Restricted additive Schwarz preconditioner; Newton–Krylov; Fully implicit; Parallel computing; Magnetohydrodynamics

1. Introduction

Magnetic reconnection is an important process in magnetically confined plasmas and has been observed both in space plasmas and in laboratory experiments [2,14,26]. Experimental investigations of the magnetic reconnection are often difficult and numerical simulations are becoming increasingly useful for reconnection studies. Mathematically, the reconnection can be described by a system of magnetohydrodynamics equations (MHD) [18], where plasma is treated as a conducting fluid satisfying the Navier–Stokes equations augmented by the electromagnetic terms such as the Lorentz force. Solving the MHD equations numerically is a challenge because of the complex, not yet fully understood behavior of the solution. The system admits phenomena such as Alfvén waves and their instabilities, and one of the intrinsic features of the system is the formation of a nearly-singular current density sheet [31], which is linked to the reconnection of magnetic field lines.

During the process of magnetic reconnection, the magnetic field configuration undergoes a topological rearrangement that results in conversion of magnetic energy into kinetic flow energy and heat over a relatively short period of time. In the strictly ideal MHD limit, magnetic field lines are "frozen" into the plasma and magnetic reconnection cannot occur. Capturing the change of the magnetic field topology requires a more general model than the ideal MHD [27]. In this paper, we focus on a resistive Hall MHD model for the magnetic reconnection.

In order to simulate this multi-scale, multi-physics phenomenon, a robust solver is needed to deal with the high degree of nonlinearity and the non-smooth, nearly-singular behavior in the system. One of the popular approaches to the numerical solution of the MHD system is based on the splitting of the system into two parts, where equations for the current and the vorticity are advanced in time, and the corresponding potentials are obtained by solving Poisson-like equations in a separate step. In such an explicit approach, to satisfy the Courant–Friedrichs–Lewy (CFL) condition, the time step may become very small, especially in the case of fine meshes.

Recently, several implicit methods have been introduced in order to overcome the time step size issue. For example, in [10,11], implicit nonlinear solvers are proposed as solution methods for systems of reduced and reduced Hall MHD equations in two-dimensional space. The solver is a matrix-free Newton–Krylov method with a physics-based preconditioner. Excellent results with large time steps are reported based on single processor calculations. In [28], a fully parallel, conservative, nonlinearly implicit numerical method is proposed for the integration of the single-fluid resistive MHD system of equations, where a variant of the matrix-free Newton–Krylov method, without preconditioning, is used in conjunction with an adaptive time integration scheme and variable spatial discretization accuracy. In this approach, the time step is not restricted by the CFL, but is restricted to some extent by the nonlinear solver as the time step needs to be cut when the unpreconditioned GMRES is unable to solve the Jacobian system.

In our paper, we develop a fully parallel nonlinearly implicit method based on a high-order time integration scheme, an inexact Newton's method with an explicitly calculated Jacobian matrix, and a Krylov subspace linear solver with a restricted additive Schwarz preconditioner defined on overlapping subdomains. We show numerically that this approach allows large time steps that are not restricted by the CFL or the linear/nonlinear solvers and works well on machines with thousands of processors. The MHD system that we consider has four equations: two linear elliptic and two nonlinear and time dependent. We take a "fully coupled" approach such that no operator splitting is applied to the system of MHD equations. Note that most existing approaches are based on some form of operator splitting which considers the unknowns of a single physical variable across all mesh points as a subproblem.

We first apply a second- or third-order implicit time integration scheme, and then, to guarantee the nonlinear consistency, we use a Newton–Krylov–Schwarz algorithm to solve the large sparse nonlinear system of algebraic equations containing all physical variables at every time step. In a Newton–Krylov–Schwarz algorithm, a system is solved by applying outer Newton iterations, whose Jacobian systems are solved with a preconditioned Krylov subspace method, where the preconditioning is accomplished via the parallel Schwarz technique. An explicit algorithm is used to compare the results obtained with the implicit approach. Both implementations are based on the PETSc (Portable Extensible Toolkit for Scientific computation) [1] library thus providing a convenient test bed for investigations of parallel properties of the algorithms. We focus on scalability studies on fine meshes and on machines with thousands of processors.

The remainder of this paper is organized as follows. In Section 2, we discuss the model MHD problem and provide some useful definitions. Spatial and temporal discretizations are covered in Section 3. Algorithmic details are described in Sections 4 and 5, and numerical results are reported in Section 6. The paper is concluded in Section 7.

2. Model MHD problem

The system of equations we model can be derived starting from the momentum transfer equations. Following [3,16], we can write

$$\eta \left(\frac{\partial \mathbf{V_e}}{\partial t} + (\mathbf{V_e} \cdot \nabla)\mathbf{V_e} \right) = -\nabla_n - \eta (\mathbf{E} + \mathbf{V_e} \times \mathbf{B}) + v \nabla^2 \mathbf{V_e} + \eta \mathbf{j} \tag{2.1}$$

for the species of electrons and

$$\eta \left(\frac{\partial \mathbf{V_i}}{\partial t} + (\mathbf{V_i} \cdot \nabla)\mathbf{V_i} \right) = -\nabla_n + \eta (\mathbf{E} + \mathbf{V_i} \times \mathbf{B}) + v \nabla^2 \mathbf{V_i} - \eta \mathbf{j} \tag{2.2}$$

for the species of ions. In (2.1) and (2.2), the plasma is considered to be quasi neutral, ions are singly charged, ion/electron number density is n, the resistivity is η, the ion/electron viscosity is given by $v_{i,e}$, $\mathbf{E}$ is the electric field strength, $\mathbf{B}$ is the magnetic induction, $\mathbf{j}$ is the current density, $m_{i,e}$ is the ion/electron mass, $\mathbf{V_{i,e}}$ is the ion/electron velocity, and $p_{i,e}$ is the ion/electron pressure. Additionally, we can introduce

$$\eta (\mathbf{V_i} - \mathbf{V_e}) \equiv \mathbf{j}. \tag{2.3}$$

Maxwell's equations enter the picture via the following three equations:

$$\nabla \cdot \mathbf{B} = 0, \tag{2.4}$$

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{j}, \tag{2.5}$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}. \tag{2.6}$$

(2.1)–(2.6) provide a full description of the plasma, given certain assumptions on $p_{i,e}$. The incompressibility condition

$$\nabla \cdot \mathbf{V_{i,e}} = 0 \tag{2.7}$$

is added if the plasma is considered incompressible.

The above equations can be viewed as the general description of plasma in the magnetohydrodynamics formalism. Special workable systems of equations may be obtained by employing reduction procedures, where certain relative magnitude ordering schemes are introduced for such physical quantities as pressure and the magnetic field induction. For example, in [15], where a magnetized, two-species (electron and ion), quasi neutral plasma with singly charged ions of mass m_i and a constant, uniform number density n_0 is considered, the reduction procedure is used to obtain systems of equations corresponding to "high-", "low-" and "zero-" β approximations with the dimensionless β, which is defined as the ratio of the plasma pressure to the magnetic field pressure [15]. In the derivation, a normalization scheme is adopted [16], such that all lengths are measured in terms of some scale a, all magnetic fields are measured in terms of some scale B_0, all velocities in terms of characteristic Alfvén speed $V_A = B_0/\sqrt{\mu_0 n_0}$, and all pressures in terms of B_0^2/μ_0. By introducing the stream functions ϕ and ψ, one can write $\mathbf{B} = \nabla \psi \times \hat{\mathbf{z}} + B \hat{\mathbf{z}}$ and $\mathbf{V_i} = \hat{\mathbf{z}} \times \nabla \phi + V \hat{\mathbf{z}}$, where $\mathbf{B}$ is the magnetic induction, $\mathbf{V_i}$ is the ion velocity, B_z and V_z are the magnetic field and ion velocity in z-direction, respectively. Furthermore, other quantities such as vorticity and the current density are expressed as $U = \nabla^2 \phi$ and $= -\nabla^2 \psi$ using the stream functions. In this paper we work with a model MHD problem described in [4,19], which also corresponds to the strongly-magnetized, "zero-β" case in [15]:

$$\begin{cases} \nabla^2 \phi = U, \\ \nabla^2 \psi = \frac{1}{2}(\psi - F), \\ \frac{\partial U}{\partial t} + [\phi, U] = \frac{1}{2}[F, \psi] + v\nabla^2 U, \\ \frac{\partial F}{\partial t} + [\phi, F] = \rho_s^2[U, \psi] + \eta \nabla^2 (\psi - \psi^0), \end{cases} \tag{2.8}$$

where U is the vorticity, F is the canonical momentum, ϕ and ψ are the stream functions for the vorticity and current density, respectively, v is the plasma viscosity, η is the normalized resistivity, $= /\omega$ is the inertial skin depth, ρ_s is the sound ion Larmor radius [19], and ψ^0 is the equilibrium component of ψ that is defined below. The current density is obtained via $= (F - \psi)/\ ^2$. The Poisson bracket is defined as $[A, B] \equiv (\partial A/\partial\ )(\partial B/\partial \kappa) - (\partial A/\partial \kappa)(\partial B/\partial\ )$.

For the case we wish to study every variable in the system is assumed to be the sum of an equilibrium and a perturbation component; i.e. $\phi = \phi^0 + \phi^1$, $\psi = \psi^0 + \psi^1$, $U = U^0 + U^1$, and $F = F^0 + F^1$, where $\phi^0 = U^0 = 0$,

$\psi^0 = \cos(\ )$, and $F^0 = (1 + \ ^2)\cos(\ )$ are the equilibrium components [26]. After substitutions, we arrive at the following system for the perturbed variables:

$$
\begin{cases}
\nabla^2 \phi^1 = U^1, \\
\nabla^2 \psi^1 = \frac{1}{2}(\psi^1 - F^1), \\
\frac{\partial U^1}{\partial t} + [\phi^1, U^1] = \frac{1}{2}[F^1, \psi^1] + v\nabla^2 U^1 + \frac{1}{2}\left(\frac{\partial \psi^1}{\partial \kappa}F_{\ \rho} + \frac{\partial F^1}{\partial \kappa}B_{\ \rho\kappa}\right), \\
\frac{\partial F^1}{\partial t} + [\phi^1, F^1] = \rho^2[U^1, \psi^1] + \eta\nabla^2 \psi^1 + \left(\frac{\partial \phi^1}{\partial \kappa}F_{\ \rho} + \rho^2\frac{\partial U^1}{\partial \kappa}B_{\ \rho\kappa}\right),
\end{cases}
\tag{2.9}
$$

where $F_{\ \rho} = -(1 + \ ^2)\sin(\ )$ and $B_{\ \rho\kappa} = \sin(\ )$. The system is defined on a rectangular domain $\Omega \equiv [\ ,\ _\kappa] \equiv [2\pi, 4\pi]$, and doubly periodic boundary conditions are assumed. For initial conditions, we use a non-zero initial perturbation in ϕ^1 and a zero initial perturbation in ψ^1. The exact form of the perturbation follows after some useful definitions. The aspect ratio is $\epsilon = \ /_\kappa$. The magnitude of the perturbation is scaled by $\xi = 10^{-4}$. We define $\tilde{\ } = \max\{\ , \rho\}$ and $\gamma = \epsilon$. For the initial value of the ϕ perturbation we use

$$
\phi^1(\ ,\kappa,0) = \begin{cases}
\xi\frac{\gamma}{\epsilon}\mathrm{erf}\left(\frac{\ }{\sqrt{2}\ }\right)\sin(\epsilon\kappa) & \text{if} \quad 0 \leqslant \ < \frac{\pi}{2} \\
-\xi\frac{\gamma}{\epsilon}\mathrm{erf}\left(\frac{\ -\pi}{\sqrt{2}\ }\right)\sin(\epsilon\kappa) & \text{if} \quad \frac{\pi}{2} \leqslant \ < \frac{3\pi}{2} \\
\xi\frac{\gamma}{\epsilon}\mathrm{erf}\left(\frac{\ -2\pi}{\sqrt{2}\ }\right)\sin(\epsilon\kappa) & \text{if} \quad \frac{3\pi}{2} \leqslant \ \leqslant 2\pi.
\end{cases}
\tag{2.10}
$$

Other field quantities are set as $U^1(\ ,\kappa,0) = \nabla^2\phi^1(\ ,\kappa,0)$ and $F^1(\ ,\kappa,0) = \psi^1(\ ,\kappa,0) - \ \nabla^2\psi^1(\ ,\kappa,0)$. From now on, we drop the superscript and assume that the four fields ϕ, ψ, U and F represent the perturbed components only. In (2.9) two of the equations are time independent and linear; another two equations are time dependent and nonlinear. Solving (2.9) numerically is the focus of this paper.

3. Discretizations

In this section, we describe some nonlinearly implicit discretization schemes for solving (2.9). We cover the computational domain with a uniform $\{M\ , M_\kappa\}$ mesh, and the corresponding mesh sizes are $\ = 2\pi/M\ $ and $_\kappa = 4\pi/M_\kappa$, respectively. Standard second-order central finite difference methods are used for all the spatial derivatives. For the time derivatives, we use a fixed time step Δt throughout the computation and we implement two time integration schemes of orders 2 and 3. At time level t^k, we denote the grid values of the unknown functions $\phi(\ ,\kappa,t)$, $\psi(\ ,\kappa,t)$, $U(\ ,\kappa,t)$, and $F(\ ,\kappa,t)$, as $\phi_{\ ,\ }$, $\psi_{\ ,\ }$, $U_{\ ,\ }$, and $F_{\ ,\ }$.

For each grid point, the equilibrium values are given as

$$
F_{\ \rho}(\ ,\ ) = -(1 + \ ^2)\sin(\ ),
\tag{3.1}
$$
$$
B_{\kappa\ \rho}(\ ,\ ) = \sin(\ ).
\tag{3.2}
$$

Eq. (3.1) corresponds to the F's x-equilibrium field component and Eq. (3.2) corresponds to B's y-equilibrium component. At time level k, the stream function components are expressed as (ignoring the k index):

$$
R_\phi(\ ,\ ) \equiv \frac{\phi_{\ +1,\ } - 2\phi_{\ ,\ } + \phi_{\ -1,\ }}{\ ^2} + \frac{\phi_{\ ,\ +1} - 2\phi_{\ ,\ } + \phi_{\ ,\ -1}}{\ _\kappa^2} - U_{\ ,\ } = 0
\tag{3.3}
$$

and

$$
R_\psi(\ ,\ ) \equiv \psi_{\ ,\ } - \ ^2\left(\frac{\psi_{\ +1,\ } - 2\psi_{\ ,\ } + \psi_{\ -1,\ }}{\ ^2} + \frac{\psi_{\ ,\ +1} - 2\psi_{\ ,\ } + \psi_{\ ,\ -1}}{\ _\kappa^2}\right) - F_{\ ,\ } = 0.
\tag{3.4}
$$

The spatial components of the other two equations are discretized as

$$R_U(\ ,\ ) \equiv \left[\left(\ _n \frac{U_{,} - U_{-1,}}{} + \ \ \frac{U_{+1,} - U_{,}}{} + \ _{\kappa_n} \frac{U_{,} - U_{,-1}}{\kappa} + \ _\kappa \frac{U_{,+1} - U_{,}}{\kappa} \right) \right.$$
$$- \left(B_{\ n} \left(\frac{F_{,} - F_{-1,}}{} + F_{\ \rho} \right) + B_{\ } \left(\frac{F_{+1,} - F_{,}}{} + F_{\ \rho} \right) + B_{\kappa_n} \left(\frac{F_{,} - F_{,-1}}{\kappa} \right) \right.$$
$$\left. \left. + B_\kappa \left(\frac{F_{,+1} - F_{,}}{\kappa} \right) \right) / \ ^2 - v \left(\frac{U_{+1,} - 2U_{,} + U_{-1,}}{^2} + \frac{U_{,+1} - 2U_{,} + U_{,-1}}{\kappa^2} \right) \right] \tag{3.5}$$

and

$$R_F(\ ,\ ) \equiv \left[\left(\ _n \left(\frac{F_{,} - F_{-1,}}{} + F_{\ \rho} \right) + \ \ \left(\frac{F_{+1,} - F_{,}}{} + F_{\ \rho} \right) + \ _{\kappa_n} \frac{F_{,} - F_{,-1}}{\kappa} + \ _\kappa \frac{F_{,+1} - F_{,}}{\kappa} \right) \right.$$
$$- \left(B_{\ n} \left(\frac{U_{,} - U_{-1,}}{} \right) + B_{\ } \left(\frac{U_{+1,} - U_{,}}{} \right) + B_{\kappa_n} \left(\frac{U_{,} - U_{,-1}}{\kappa} \right) \right.$$
$$\left. \left. + B_\kappa \left(\frac{U_{,+1} - U_{,}}{\kappa} \right) \right) \rho^2 - \eta \left(\frac{\psi_{+1,} - 2\psi_{,} + \psi_{-1,}}{^2} + \frac{\psi_{,+1} - 2\psi_{,} + \psi_{,-1}}{\kappa^2} \right) \right]. \tag{3.6}$$

We have used the following notations in the above equations:

$$= -\frac{\phi_{,+1} - \phi_{,-1}}{2 \ _\kappa}, \tag{3.7}$$

$$_\kappa = \frac{\phi_{+1,} - \phi_{-1,}}{2 \ }, \tag{3.8}$$

$$_n = \ = \frac{}{2}, \qquad _{\kappa_n} = \ _\kappa = \frac{\kappa}{2}, \tag{3.9}$$

and

$$B_{\ } = \frac{\psi_{,+1} - \psi_{,-1}}{2 \ _\kappa}, \tag{3.10}$$

$$B_\kappa = -\frac{\psi_{+1,} - \psi_{-1,}}{2 \ } + B_{\kappa\ \rho}(\ ,\ ), \tag{3.11}$$

$$B_{\ n} = B_{\ } = \frac{B}{2}, \tag{3.12}$$

$$B_{\kappa_n} = B_\kappa = \frac{B_\kappa}{2}. \tag{3.13}$$

We note that above spatial discretization is second-order accurate. Two of the four equations in (2.9) are time dependent. As suggested in [23], temporal discretization of higher order often provides for a better solution accuracy in numerical simulations of fluid dynamics problems, despite the fact that a lower spatial discretization order is used. For instance, the temporal discretization may be of order 3 or 4, while the order of the spatial discretization is 2. In this paper, we employ backward differentiation formulas (BDF) [21] for the implicit methods and Adams formula [21] for the explicit version of our algorithm. Using multistep formulas requires the solutions from previous time steps. For example, the third-order BDF for system (2.9) with the help of (3.3)–(3.6) results in

$$\begin{cases} G_\phi^{+1}(\ ,\ ) \equiv R_\phi^{+1}(\ ,\ ) = 0, \\ G_\psi^{+1}(\ ,\ ) \equiv R_\psi^{+1}(\ ,\ ) = 0, \\ G_U^{+1}(\ ,\ ) \equiv \frac{1}{6\Delta t}(11U_{,}^{+1} - 18U_{,} + 9U_{,}^{-1} - 2U_{,}^{-2}) - R_U^{+1}(\ ,\ ) = 0, \\ G_F^{+1}(\ ,\ ) \equiv \frac{1}{6\Delta t}(11F_{,}^{+1} - 18F_{,} + 9F_{,}^{-1} - 2F_{,}^{-2}) - R_F^{+1}(\ ,\ ) = 0, \end{cases} \tag{3.14}$$

where $(U\ , F\ )$, (U^{-1}, F^{-1}), (U^{-2}, F^{-2}) are solutions for (U, F) obtained at k, -1 and -2 time steps, respectively. Therefore, a high-order BDF requires sufficient solution history to be accumulated at the begin-

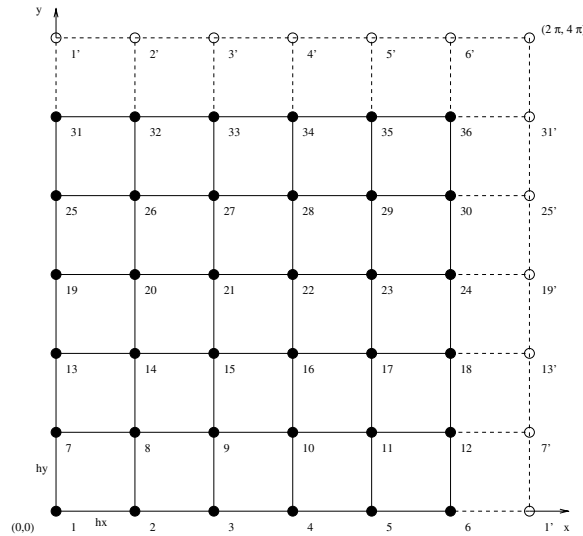Fig. 1. Rectangular domain $\Omega = [0, 2\pi] \times [0, 4\pi]$ with doubly periodic boundaries and a sample 6×6 mesh. The solid circles indicate genuine mesh points while empty circles denote virtual points that correspond to the boundary mesh points on the opposite side of the domain.

ning of the time integration process. In our approach, the time integration process begins, starting from an initial guess at $t = 0$, with a BDF of order one (backward Euler), gradually increasing the BDF order up to a desired value as more and more solution history becomes available. For comparison purposes, we also implement an explicit scheme:

$$\begin{cases} R_{\phi}^{+1}(\,,\,) = 0, \\ R_{\psi}^{+1}(\,,\,) = 0, \\ U_{,}^{+1} - U_{,} - \Delta t \left(\frac{3}{2} R_U(\,,\,) - \frac{1}{2} R_U^{-1}(\,,\,) \right) = 0, \\ F_{,}^{+1} - F_{,} - \Delta t \left(\frac{3}{2} R_F(\,,\,) - \frac{1}{2} R_F^{-1}(\,,\,) \right) = 0. \end{cases} \tag{3.15}$$

The system of PDEs (2.9) is defined on a rectangular domain and doubly periodic boundary conditions are used [19]. There are several ways to numerically implement the doubly periodic boundary conditions [24], and we take the so-called "wraps around" approach that is provided as an option in [1], in which the top, the bottom as well as the left and the right boundaries are glued together. Fig. 1 helps to illustrate the point by showing an example of the 6×6 mesh. The standard five-point-stencil discretization of the PDE centered at the mesh point number 8 requires corresponding information from four "neighboring" points: 2, 9, 14 and 7. For the discretization centered at the boundary mesh point 7 the four "neighboring" points are 1, 8, 13 and 12. In some sense, a system with the periodic boundary conditions is more global than the one with Dirichlet or Neumann boundaries and this "globalization" may have an impact on the performance of the domain decomposition [30,32] method used to solve the system of equations.

4. One-level Newton–Krylov–Schwarz methods

At each time step, we compute the solution by solving a system of nonlinear algebraic equations $G(E) = 0$, which is obtained by putting the finite difference equations (3.14) in a certain order. For some algorithms, the orderings of unknowns and the finite difference equations are not important, but for our algorithm to work, it is crucial to order them in the "fully coupled" fashion. More precisely, we define

$$E = (\phi_{11}, \psi_{11}, U_{11}, F_{11}, \phi_{21}, \psi_{21}, U_{21}, F_{21}, \cdots)^{\mathrm{T}},$$

and

$$G = (G_\phi(1,1), G_\psi(1,1), G_U(1,1), G_F(1,1), G_\phi(2,1), G_\psi(2,1), G_U(2,1), G_F(2,1), \dots)^\mathrm{T},$$

where the notations of (3.14), ignoring the superscript, are used. With this kind of ordering the four physical variables at the same mesh points are always together throughout the computations. The mesh points are ordered subdomain by subdomain for the purpose of parallel processing. The system is solved with a one-level Newton–Krylov–Schwarz (NKS) [6,7], which is a general purpose parallel algorithm for solving systems of nonlinear algebraic equations. The Newton iteration is given as $E_{k+1} = E_k - \lambda_k J(E_k)^{-1} G(E_k)$, $k = 0, 1, \dots$, where E_0 is a solution obtained at the previous time step, $J(E_k) = G'(E_k)$ is the Jacobian at E_k, and λ_k is the steplength determined by a linesearch procedure [13]. We mention that another possible nonlinear solver would be the additive Schwarz preconditioned Newton method [8,22]. Due to doubly periodic boundary conditions, the Jacobian has a one-dimensional null-space that is removed by projecting out a constant [25]. The accuracy of the Jacobian solve is determined by some $\eta_k \in [0, 1)$ and the condition $\|G(E_k) + J(E_k)s_k\| \leqslant \eta_k \|G(E_k)\|$. The overall algorithm can be described as follows:

(1) Inexactly solve the linear system $J(E_k)s_k = -G(E_k)$ for s_k using a preconditioned GMRES(30) [29].
(2) Perform a full Newton step with $\lambda_0 = 1$ in the direction s_k.
(3) If the full Newton step is unacceptable, backtrack λ_0 using a backtracking procedure until a new λ is obtained that makes $E_+ = E_k + \lambda s_k$ an acceptable step.
(4) Set $E_{k+1} = E_+$, go to step 1 unless a stopping condition has been met.

In step 1 above we use a right-preconditioned GMRES to solve the linear system; i.e., the vector s_k is obtained by approximately solving the linear system $J(E_k)M^{-1}(Ms_k) = -G(E_k)$, where M^{-1} is the one-level additive Schwarz preconditioner. To formally define M^{-1}, we need to introduce a partition of Ω. We first partition the domain into non-overlapping subdomains Ω_l, $l = 1, \dots, N$, as in Fig. 2. In order to obtain an overlapping decomposition of the domain, we extend each subregion Ω_l to a larger region Ω_l', i.e., $\Omega_l \subset \Omega_l'$. Only simple box decompositions are considered in this paper – all subdomains Ω_l and Ω_l' are rectangular and made up of integral numbers of fine mesh cells. The size of Ω_l is $H_x \times H_\kappa$. The subdomain Ω_l' is not always simply connected as shown in Fig. 2, however, if we pull all the pieces together so that the subdomain is "connected", we can more conveniently define the size of Ω_l' as $H' \times H_\kappa'$, where the H's are chosen so that the overlap, δ, is uniform in the number of fine grid cells all around the perimeter, i.e.,
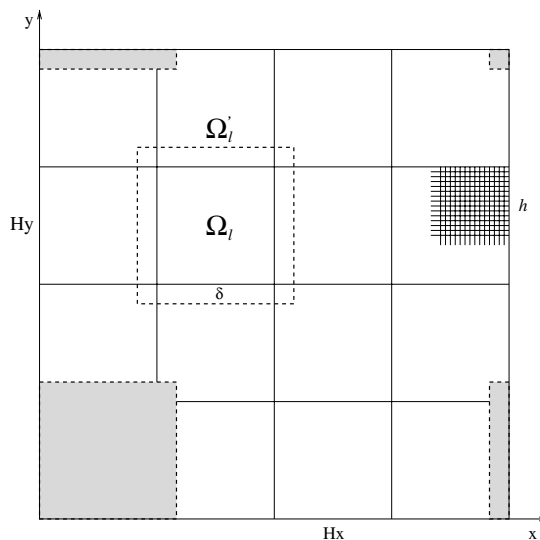

Fig. 2. Decomposition of domain Ω with an overlap δ. The solid lines indicate the partition of the domain into non-overlapping substructures Ω_l of size $H_x \times H_\kappa$ and the dashed rectangle indicates an overlapping subdomain Ω_l'. The filled rectangles show the formation of an extended boundary subdomain with the doubly periodic domain boundary $\partial \Omega$ and the incomplete fine mesh of solid lines illustrates underlying uniform subintervals with mesh size h.

$$\delta = (H' - H)/2 = (H'_\kappa - H_\kappa)/2 \tag{4.1}$$

for every subdomain. Note that the boundary subdomains are also extended all around their perimeters because of the doubly periodic physical boundary condition. On each extended subdomain Ω', we construct a subdomain preconditioner B_l, whose elements are $B = \{J\}$, where the node indexed by $(\,,\,)$ belongs to Ω'. The entry J_{ij} is calculated with finite differences

$$J = 1/(2\alpha)(G(E + \alpha) - G(E - \alpha)), \tag{4.2}$$

where $0 < \alpha \ll 1$ is a constant and a graph coloring technique is used to efficiently compute the sparse Jacobian [1,12]. We also compute the entries of the Jacobian matrix using analytical expressions for comparison purposes. Homogeneous Dirichlet boundary conditions are used on the subdomain boundary $\partial\Omega'$. The classical additive Schwarz preconditioner [5] can be written as

$$M^{-1} = \sum_{=1}^{N} (R)^{\mathrm{T}} B^{-1} R. \tag{4.3}$$

Let n be the total number of mesh points and $_\eta'$ the total number of mesh points in Ω'. Then, R^δ is an $_\eta' \times_\eta$ block matrix that is defined as its 4×4 block element (R^δ) is an identity block if the integer indices $1 \leqslant \,\leqslant_\eta'$ and $1 \leqslant \,\leqslant_\eta$ belong to a mesh point in Ω', or a block of zeros otherwise. The R^δ serves as a restriction matrix because its multiplication by a block $_\eta \times 1$ vector results in a smaller $_\eta' \times 1$ block vector by dropping the components corresponding to mesh points outside Ω'. The $_\eta' \times_\eta$ block matrix R^0 is defined similarly with the exception that its application to a $_\eta \times 1$ vector also zeros components that correspond to the mesh points laying in $\Omega' \setminus \Omega$. In this paper we use the left restricted additive Schwarz preconditioner [9] which is given by

$$M^{-1} = \sum_{=1}^{N} (R^0)^{\mathrm{T}} B^{-1} R^\delta, \tag{4.4}$$

where δ is the size of the inter subdomain overlap. The restricted additive Schwarz preconditioner can be interpreted as using the residuals from the points in the overlapping region, while discarding the computed values in the overlapping region. For the problem we are considering the restricted additive preconditioner performs much better than the classical additive Schwarz preconditioner, especially when the number of processors is large. Various inexact additive Schwarz preconditioners can be constructed by replacing the matrices B_l in (4.4) with convenient and inexpensive to compute matrices, such as those obtained with incomplete factorizations. In this paper we employ the LU factorization.

If the classical Schwarz preconditioner is applied to symmetric positive definite systems resulting from a discretization of elliptical problems, then the condition number κ of the preconditioned system satisfies $\kappa \leqslant C(1 + H/\delta)/H^2$ for the one-level method and $\kappa \leqslant C(1 + H/\delta)$ for the two-level method, where C is independent of h, H and δ. In the above formulation H is the effective subdomain diameter. The factor $1/H^2$, associated with the number of subdomains on the fine level, lends itself to an increase in numbers of iterations with the increase in the total numbers of subdomains. This increase can be justified by the need to exchange information between distant subdomains. The use of a coarse mesh facilitates the exchange and, potentially, stabilizes the number of iterations. Our problem, however, is of a mixed elliptic/parabolic type, where the growth of the linear iterations numbers does not follow exactly the theoretical predictions for elliptical problems. One consequence of such a deviation is that the growth of the linear iterations numbers is not severe and an introduction of the traditional coarse space does not seem necessary to attain a good parallel performance.

5. Explicit method

To verify the accuracy of the implicit solution, as well as to compare the parallel performance of the implicit method, we implement an explicit method, in which the equations for the vorticity and the canonical momentum are first advanced in time and, to guarantee temporal consistency, corresponding equations are solved to update ϕ and ψ. For the time integration we use a second-order explicit Adams formulas, albeit modified to allow for the adaptive time stepping, in which the time step Δt is determined by the CFL condition. In our

current implementation the CFL condition is estimated based on the maximum speed of the signal propagation on a spatial mesh. In particular, from the values of ϕ we compute the spatial components of the plasma's in-plane velocity:

$$\mathbf{v} = \hat{\mathbf{z}} \times \nabla\phi. \tag{5.1}$$

Then, values of v_x and v_y which are the in-plane velocity components in x and y directions, respectively, are used to compute the CFL estimates:

$$\text{CFL} = \left| \frac{\Delta t}{} \right| \quad \text{and} \quad \text{CFL}_\kappa = \left| \frac{\kappa \Delta t}{\kappa} \right|, \tag{5.2}$$

where Δt is the proposed time step size and h_x and h_y are the mesh intervals in x and y directions. The CFL estimate is then chosen as the maximum value of estimates from (5.2) over all mesh points. Similarly, the Alfvén speed-based CFL estimates are used for the in-plane values of the magnetic field, which relate to ψ values via

$$\mathbf{B}_\perp = \nabla\psi \times \hat{\mathbf{z}}. \tag{5.3}$$

If the computed CFL number is

$$\text{CFL} < 0.9, \tag{5.4}$$

then the time step size of Δt is accepted and the temporal updates (the second-order explicit Adams), using the same notations as in (3.15), are computed by

$$U_{,}^{+1} = U_{,} + \Delta t \left(\frac{3}{2} R_U(\,,\,) - \frac{1}{2} R_U^{-1}(\,,\,) \right),$$
$$F_{,}^{+1} = F_{,} + \Delta t \left(\frac{3}{2} R_F(\,,\,) - \frac{1}{2} R_F^{-1}(\,,\,) \right). \tag{5.5}$$

Otherwise, the time step size Δt is reduced until (5.4) is satisfied. Eqs. (5.5) are, in fact, used if no time step resize is required. However, if the time step size needs to be reduced, then the following equations are used:

$$U_{,}^{+1} = U_{,} + \Delta t R_U(\,,\,) + \frac{(\Delta t)^2}{2\Delta t_1}(R_U(\,,\,) - R_U^{-1}(\,,\,)),$$
$$F_{,}^{+1} = F_{,} + \Delta t R_F(\,,\,) + \frac{(\Delta t)^2}{2\Delta t_1}(R_F(\,,\,) - R_F^{-1}(\,,\,)), \tag{5.6}$$

where Δt_1 is the size of the previous time step and $\Delta t \neq \Delta t_1$. Eqs. (5.6) become Eqs. (5.5) assuming $\Delta t = \Delta t_1$.

After the time advancement is done, two linear equations in (3.15) are solved. In the linear solves, one-level, right-sided restricted additive Schwarz preconditioner with LU on all subdomains is used and the preconditioned systems are solved with GMRES.

We mention that the matrix–vector products in the explicit method can be computed faster in the case of the tensor product grid. In our implementation, however, we assume that the mesh is unstructured and therefore do not exploit special grid properties.

6. Numerical experiments

We use the Portable Extensible Toolkit for Scientific computation (PETSc), developed at Argonne National Laboratory [1], for our implementations of the algorithms discussed in previous sections. The codes are written in a hostless manner and allow easy switching between different numbers of processors. Each processor is assigned one subdomain, and the information pertaining to the interior of a subdomain is uniquely owned by that processor. In the implicit algorithm implementation, the processor stores subvectors and a block of the Jacobian matrix associated with an extended subdomain. At the beginning of every nonlinear iteration, local blocks of the Jacobian, as well as the preconditioning matrices, are computed. The preconditioning matrices are factored, and the upper and lower triangular parts are stored. After a solution of each subproblem is

obtained, those portions that lie within the overlapping regions are sent to neighboring processors to complete collective operations. Participating processors communicate with each other by message passing using MPI [20].

Similarly in the explicit implementation, each processor stores subvectors and blocks of linear operators associated with the linear systems. Subtracting the nonlinear solver functionality from the implicit implementation discussed above, the remaining procedures are identical in both implicit and explicit implementations. Testing platforms include a beowulf cluster and an IBM BlueGene/L system.

To illustrate the model behavior, we choose nominal values of the inertial skin depth $= 0.08$ and the ion sound Larmor radius $\rho = 0.24$. The normalized resistivity and viscosity are $\eta = 10^{-3}$ and $v = 10^{-3}$, respectively. The time in the system is normalized to the Alfvén time $\tau_A = \sqrt{4\pi_i}\ /B_{x0}$, where B_{y0} is the characteristic magnitude of the equilibrium magnetic field and l_x is the macroscopic scale length [19]. The computational domain Ω is uniformly partitioned into rectangular meshes up to 1980×1980 in size. The stopping conditions for the iterative processes are given as follows:

- relative reduction in the nonlinear solve: $\|G(E_{\cdot})\| \leqslant 10^{-7}\|G(E_0)\|$,
- absolute tolerance in the nonlinear solve: $\|G(E_{\cdot})\| \leqslant 10^{-7}$,
- relative reduction in the Jacobian solve: $\|_{\cdot}\| \leqslant 10^{-10}\|_0\|$,
- absolute tolerance in the Jacobian solve: $\|_{\cdot}\| \leqslant 10^{-7}$,
- GMRES restart value: 30.
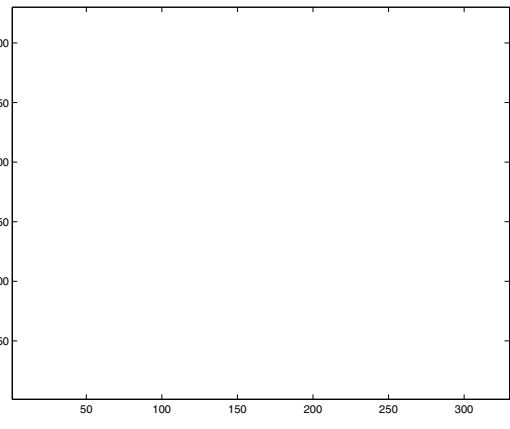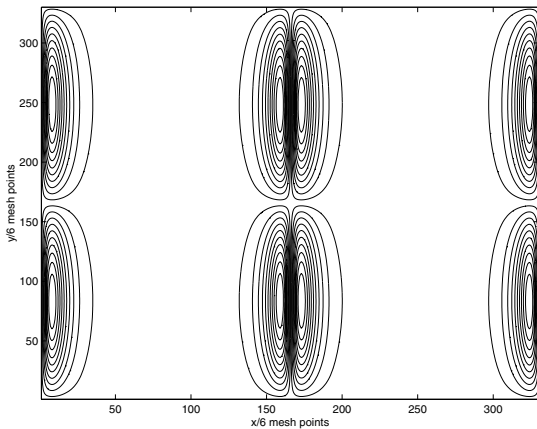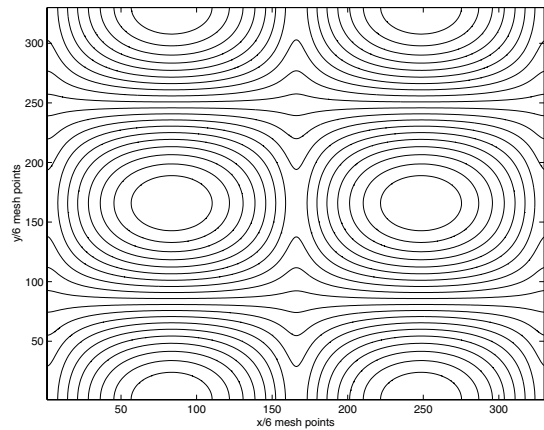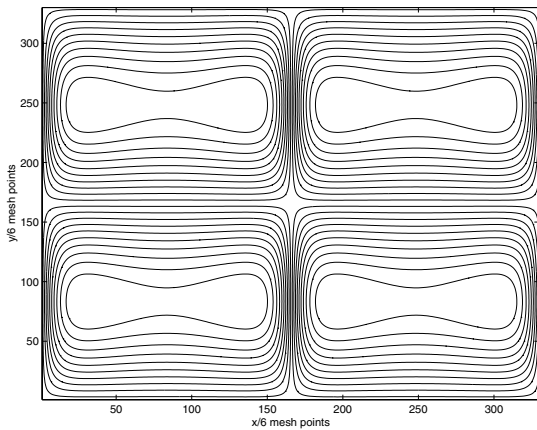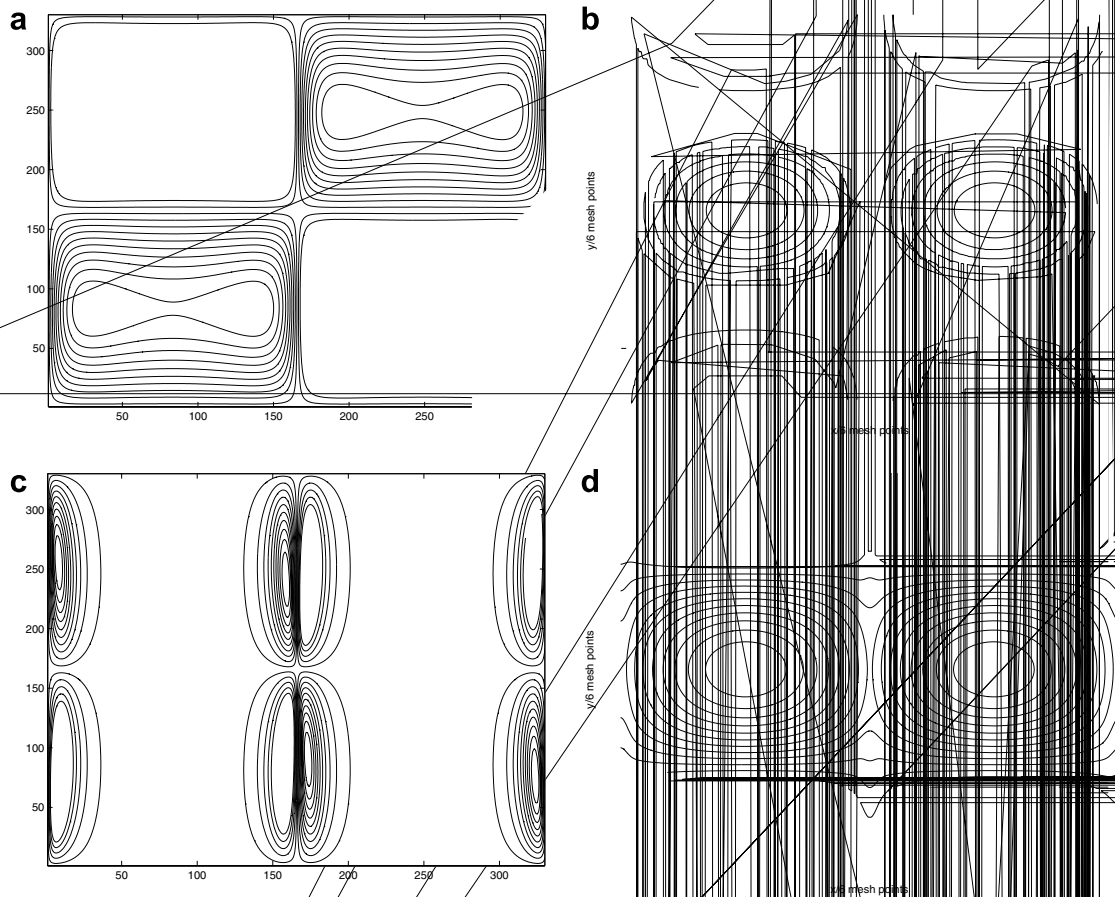
Fig. 4. Contour plots of ϕ (a), ψ (b), U (c), and F (d). The results are obtained on 1980×1980 mesh, $\Delta t = 100\tau_A$, time $t = 200\tau_A$, $\eta = 10^{-3}$, $v = 10^{-3}$, implicit time stepping.

The same stopping conditions are used f   xplicit runs except, of course, for the nonlinear solver parameters.

Typical solutions are shown in Figs. 3–5 at mulated times $t = 100\tau_A$, $t = 200\tau_A$ and $t = 290\tau_A$, respectively, and corresponding current density contours are given in Fig. 6. The initial perturbation in ϕ produces a feature-rich behavior in ψ, U, and F. The four variables in the system evolve at different rates: ϕ and ψ evolve at a slower rate than F and U. For $\eta = 10^{-3}$ and $v = 10^{-3}$ we observe an initial slow evolution of the four fields and the current density profiles up to time $100\tau_A$ and the solution blows up at time near 290τ_A. In the middle of the domain the notorious "X-point" is developed, as can be seen in the F contours, where the magnetic flux is
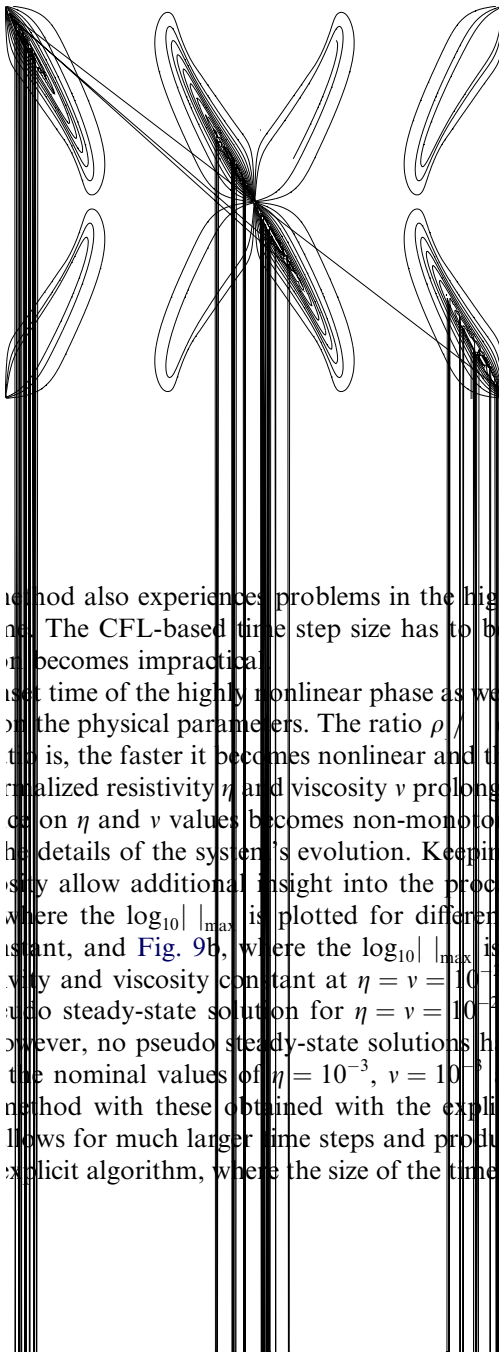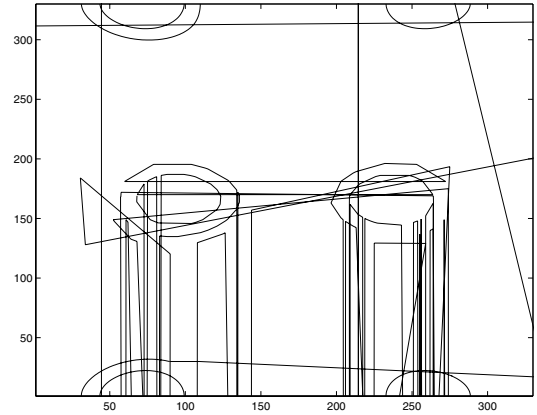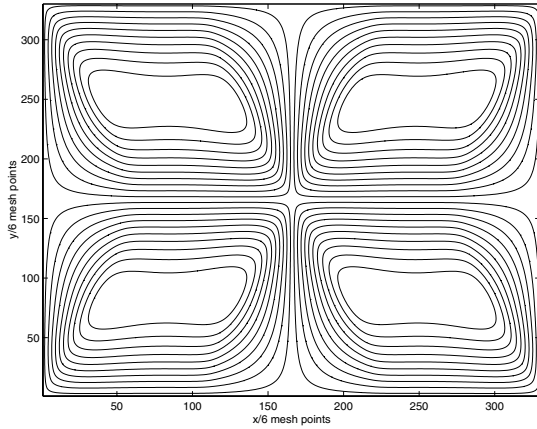
method also experiences problems in the highly nonlinear phase of evolution at about the same sim- The CFL-based time step size has to be continuously reduced to the point, where further time becomes impractical.

time of the highly nonlinear phase as well as the exact features of current density surfaces depend the physical parameters. The ratio $\rho /$ effectively dominates the behavior of the system. The lar- is, the faster it becomes nonlinear and the harder it is to solve. For $\rho / = 1.0$, reducing the val- alized resistivity η and viscosity v prolongs the linear evolution phase. As the ratio is increased, the tion η and v values becomes non-monotonic and further studies are needed in order to fully char- details of the system's evolution. Keeping the $\rho / = 3.0$ and changing the values for resistivity ity allow additional insight into the processes taking place in the system. As can be observed in where the $\log_{10}| |_{max}$ is plotted for different values for resistivity and viscosity, while keeping the tant, and Fig. 9b, where the $\log_{10}| |_{max}$ is plotted for different ratios $\rho /$, while keeping values ty and viscosity constant at $\eta = v = 10^{-1}$, there is evidence to support the claim that the system o steady-state solution for $\eta = v = 10^{-1}$, where only minor changes occur after simulated time wever, no pseudo steady-state solutions have been observed for other values of η and v.

the nominal values of $\eta = 10^{-3}$, $v = 10^{-3}$ and $\rho / = 3.0$, we compare solutions obtained by our method with these obtained with the explicit method. Fig. 10 shows that the third-order implicit lows for much larger time steps and produces a solution that is very close to the solution obtained plicit algorithm, where the size of the time step is determined by the CFL constraint. Fig. 11, where
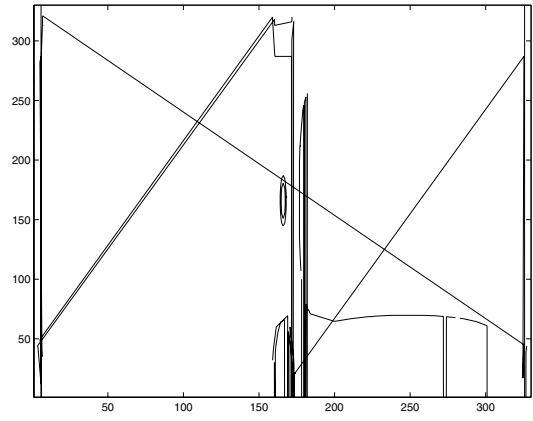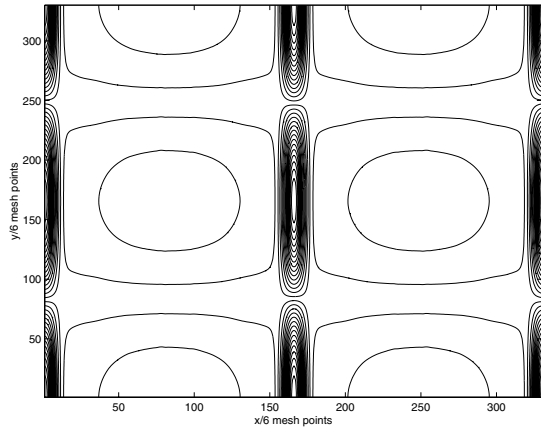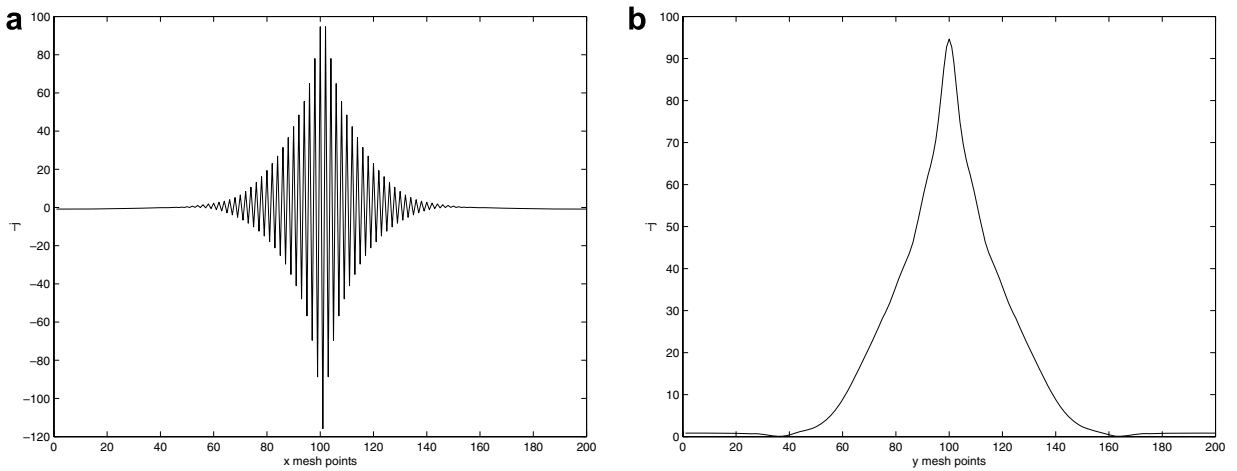
Fig. 8. The highly nonlinear phase of the system's evolution with current density peaks profiles in x-direction (a) and y-direction (b). The results are obtained on 200×200 mesh, $\eta = 10^{-3}$, $v = 10^{-3}$, $\Delta t = 1.0\tau_A$, $n = 1$, implicit algorithm.
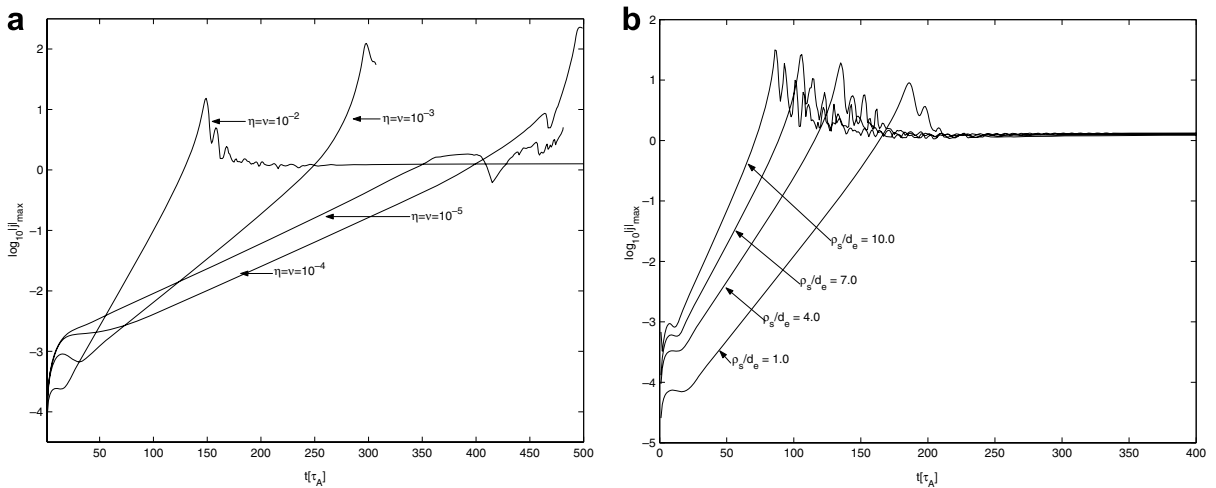

Fig. 9. Current density maxima as a function of time, 600×600 mesh, $\Delta t = 1.0\tau_A$, $n = 25$, implicit algorithm: (a) constant ratio $\rho / = 3.0$ with $\eta = v \in [10^{-5}, 10^{-2}]$ and (b) constant $\eta = v = 10^{-2}$ and $\rho / \in [1.0, 10.0]$.

compute times are plotted for both explicit and implicit methods using different meshes and varying numbers of processors, compares the performance of the implicit algorithm to the explicit one.

Next, we look at some of the machine-dependent properties of the algorithm. Our main focus is on the compute time scalability, which is an important quality in evaluating parallel algorithms. Again, the nominal values for physical parameters are chosen for our experiments. Table 1 shows parallel performance results obtained at $t = 100\tau_A$, $t = 200\tau_A$ and $t = 280\tau_A$. The restart sequence allows runs at specified simulated times, while preserving the temporal discretization order. For instance, in the top part of Table 1 the code is restarted at $t = 100\tau_A$ and run for 10 time steps of size $\Delta t = 1.0\tau_A$ to stop at $t = 110\tau_A$. The compute time as well as numbers of linear and nonlinear iterations are recorded. In these experiments the number of nonlinear function evaluations coincides with the number of nonlinear iterations. Similar techniques are also used at $t = 200\tau_A$ and $t = 280\tau_A$. The restarting functionality allows code runs with varying number of processors.
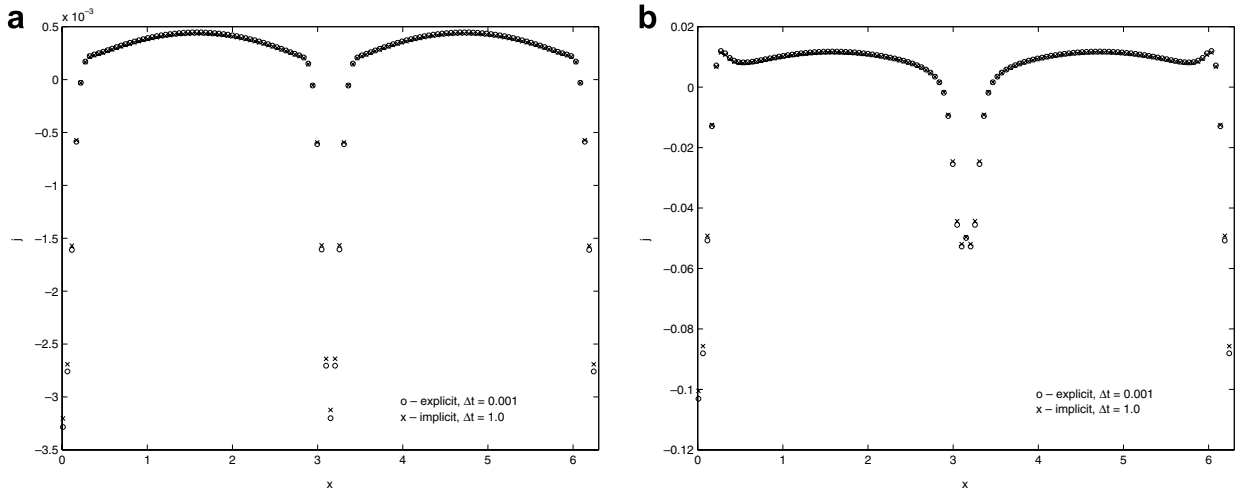
Fig. 10. Current density profiles comparisons obtained with explicit and implicit methods, 600×600 mesh, $\eta = 10^{-3}$, $v = 10^{-3}$, $\rho_{\cdot} = 0.24$, $= 0.08$, the implicit time step size $\Delta t = 1.0\tau_A$ and the explicit time step size $\Delta t = 0.001\tau_A$: (a) $t = 100\tau_A$ and (b) $t = 200\tau_A$.

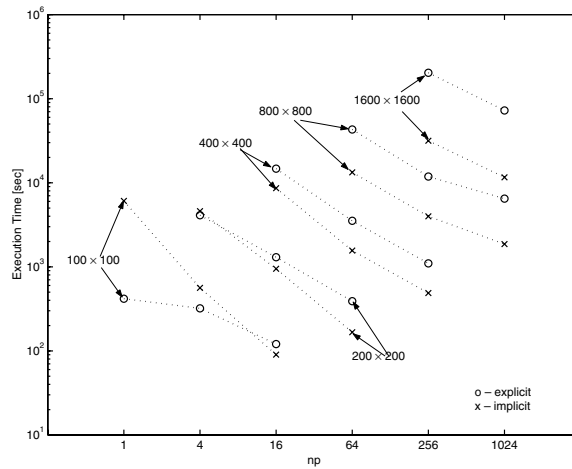Fig. 11. Execution time comparison between explicit and implicit methods on 100×100, 200×200, 400×400, 800×800 and 1600×1600 meshes obtained with $n = 1, 4, 16, 64, 256, 1024$ processors.

For a given number of subdomains, the inter-domain overlapping size is chosen to minimize the computing time. That is, referring again to Table 1, in the case of $n = 900$ at $t = 280\tau_A$ the inter-domain overlap $\delta = 13$ results in the compute time of 742.8 s, while larger and smaller overlap sizes result in longer compute times. This optimal overlap size varies for different numbers of subdomains effectively ensuing a range of the overlapping sizes $\delta \in [7, 16]$.

We show the parallel speedup curves in Fig. 12, where the speedup value is calculated as $t(225)/t(n)$, which gives a ratio of time needed to solve the problem with 225 processors to the time needed to solve the problem with np processors on 1980×1980 mesh. A comparison to the ideal speedup is made in every plot. The speedup plots are reported at simulated times $t = 100\tau_A$, $t = 200\tau_A$ and $t = 280\tau_A$. It is observed that the compute time scalability deteriorates as the problem becomes more and more nonlinear. Indeed, at $t = 100\tau_A$ the speedup is nearly linear. However, for $t = 200\tau_A$ and $t = 280\tau_A$ the experimental speedup becomes sublinear for $n > 400$. Some of the speedup degradation can be attributed to the insufficient problem size pointing

Table 1

Scalability with respect to the number of subdomains, one-level algorithm, 1980×1980 mesh, LU factorization for all subproblems, $\Delta t = 1.0\tau_A$, 10 time steps taken at simulation times $t = 100\tau_A$, $t = 200\tau_A$ and $t = 280\tau_A$

np	Compute time (s)	Total nonlinear iterations	Linear/nonlinear	δ
$t = 100\tau_A$				
225	716.8	10	42.2	9
324	457.8	10	46.1	9
400	342.3	10	56.0	7
484	284.7	10	56.4	8
900	151.7	10	63.0	9
1936	101.0	10	96.5	10
2025	90.8	10	98.4	9
$t = 200\tau_A$				
225	965.5	11	65.6	12
324	754.3	12	53.6	12
400	465.7	10	68.7	13
484	427.8	10	74.0	15
900	276.8	12	67.7	16
1936	218.6	14	162.4	13
2025	189.9	13	149.5	13
$t = 280\tau_A$				
225	2473.1	24	113.5	12
324	1691.9	24	127.7	12
400	1359.6	24	135.1	12
484	1185.0	25	141.3	11
900	742.8	25	181.0	13
1936	514.8	27	226.6	13
2025	504.8	26	244.3	13

For every global domain partition the inter-domain overlap $\delta \in [7, 16]$, such that the compute time is minimized over the range of test cases, where the $\delta \in [5, 20]$. Physical parameters are given as follows: $\eta = v = 10^{-3}$, $ = 0.08$ and $\rho = 0.24$. The measurements are conducted with the number of processors $n = 225, 324, 400, 484, 900, 1936, 2025$ and each processor is assigned one subdomain.

out that the problems sizes about $15M$ unknowns are not large enough for $n > 1000$ and the communication cost takes its toll on the overall performance. Larger problem sizes may sufficiently improve speedup values.

7. Conclusions and future work

A parallel, nonlinearly implicit method is used for solving a model MHD problem. The proposed fully coupled implicit scheme with the third-order temporal discretization allows much larger time steps than the explicit method, while still preserving the solution accuracy. Moreover, the fully coupled implicit scheme demonstrates superior compute time performance when compared to the explicit scheme. This performance of the implicit method is attained on fine meshes with a large number of processors. Even though the numerical experiments are carried out on structured meshes, we do not use any special algorithms that take an advantage of the tensor product nature of the meshes for the explicit and the implicit methods. The extension of the algorithms to unstructured meshes and to three-dimensional space is straightforward. It is also observed that the one-level Schwarz preconditioner is sufficient for times up to $t = 150\tau_A$, where the execution time speedup is nearly perfect (Fig. 12a) and the deterioration of iteration numbers is mild. However, for the latter half of the simulation $(150\tau_A - 300\tau_A)$, the one-level method needs some improvement.

Future continuation of this work may include solutions of the MHD problem on finer meshes with a larger number of processors and with multilevel versions of the algorithm. Longer time integration with various η and v values, as well as higher ρ to ratios, may be helpful in the further understanding of the fully coupled implicit algorithm for the numerical solutions of MHD problems.
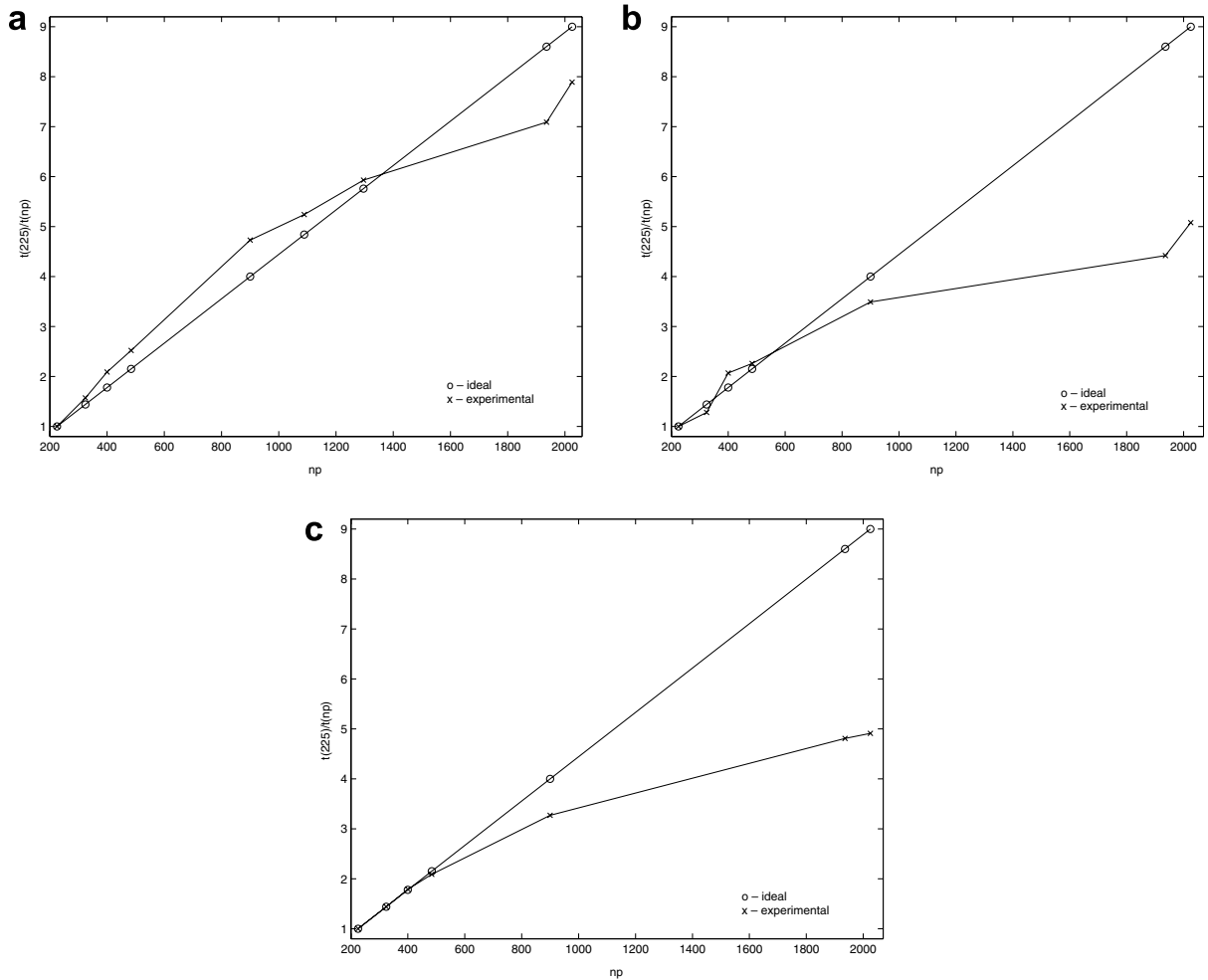
Fig. 12. Speedup curves $t(\eta_n = 225)/t(\eta_n)$ as compared to the ideal speedup for 1980×1980 mesh, *LU* factorization for all subproblems, $\Delta t = 1.0\tau_A$, 10 time steps at times $t = 100\tau_A$ (a), $t = 200\tau_A$ (b) and $t = 280\tau_A$ (c). Physical parameters are $\eta = v = 10^{-3}$, $= 0.08$ and $\rho = 0.24$. The straight line in the graphs shows the ideal speedup.

The importance of mesh adaptivity in resolving the near singular behavior of the current density in the final stages of the simulation has been quantified in [17] for the case of *h*-type Cartesian adaptive mesh refinement. We plan to extend our fully implicit parallel approach to *r*-type adaptive mesh refinement, by drawing points into the near singular layer and introducing additional metric terms into the governing equations, without changing the logical organization of the data or load balancing.

Acknowledgements

References

[1] S. Balay, K. Buschelman, V. Eijkhout, W.D. Gropp, D. Kaushik, M. Knepley, L.C. McInnes, B.F. Smith, H. Zhang, PETSc Users Manual, ANL-95/11 – Revision 2.3.1, Argonne National Laboratory, 2006.
[2] D. Biskamp, Nonlinear Magnetohydrodynamics, Cambridge University Press, New York, 1993.
[3] S.I. Braginskii, Reviews of Plasma Physics, vol.1, Consultants Bureau, New York, 1965.

 [4] E. Cafaro, D. Grasso, F. Pegoraro, F. Porcelli, A. Saluzzi, Invariants and geometric structures in nonlinear Hamiltonian magnetic reconnection, Phys. Rev. Lett. 80 (20) (1998) 4430–4433.
 [5] X.-C. Cai, Additive Schwarz algorithms for parabolic convection-diffusion equations, Numer. Math. 60 (1990) 41–62.
 [6] X.-C. Cai, W.D. Gropp, D.E. Keyes, R.G. Melvin, D.P. Young, Parallel Newton–Krylov–Schwarz algorithms for the transonic full potential equation, SIAM J. Sci. Comput. 19 (1998) 246–265.
 [7] X.-C. Cai, W.D. Gropp, D.E. Keyes, M.D. Tidriri, Newton–Krylov–Schwarz methods in CFD, in: R. Rannacher (Ed.), Proceedings of the International Workshop on the Navier–Stokes Equations, Notes in Numerical Fluid Mechanics, Vieweg Verlag, Braunschweig, 1994.
 [8] X.-C. Cai, D.E. Keyes, Nonlinearly preconditioned inexact Newton algorithms, SIAM J. Sci. Comput. 24 (2002) 183–200.
 [9] X.-C. Cai, M. Sarkis, A restricted additive Schwarz preconditioner for general sparse linear systems, SIAM J. Sci. Comput. 21 (1999) 792–797.
[10] L. Chacón, D.A. Knoll, A 2D high-β Hall MHD implicit nonlinear solver, J. Comput. Phys. 188 (2003) 573–592.
[11] L. Chacón, D.A. Knoll, J.M. Finn, An implicit, nonlinear reduced resistive MHD solver, J. Comput. Phys. 178 (2002) 15–36.
[12] T.F. Coleman, J.J. Moré, Estimation of sparse Jacobian matrices and graph coloring problems, SIAM J. Numer. Anal. 20 (1983) 187–209.
[13] J.E. Dennis, R.B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, SIAM, Philadelphia, 1996.
[14] A. Edwards, D. Campbell, W. Engelhardt, H. Fahrbach, R. Gill, R. Granetz, S. Tsuji, B. Tubbing, A. Weller, J. Wesson, D. Zasche, Rapid collapse of a plasma sawtooth oscillation in the JET tokamak, Phys. Rev. Lett. 57 (2) (1986) 210–213.
[15] R. Fitzpatrick, Scaling of forced magnetic reconnection in the Hall-magnetohydrodynamical Taylor problem with arbitrary guide field, Phys. Plasmas 11 (8) (2004) 3961–3968.
[16] R. Fitzpatrick, Scaling of forced magnetic reconnection in the Hall-magnetohydrodynamic Taylor problem, Phys. Plasmas 11 (3) (2004) 937–946.
[17] K. Germaschewski, A. Bhattacharjee, R. Grauer, D. Keyes, B. Smith, Using Krylov–Schwarz methods in an adaptive mesh refinement environment, in: Proceedings of the Chicago Workshop on adaptive mesh refinement methods, Lecture Notes in Computational Science and Engineering, vol. 41, Springer, 2003, pp. 115–124.
[18] R.J. Goldston, P.H. Rutherford, Introduction to Plasmas Physics, IOP Publishing, 1995.
[19] D. Grasso, F. Pegoraro, F. Porcelli, F. Califano, Hamiltonian magnetic reconnection, Plasma Phys. Control Fusion 41 (1999) 1497–1515.
[20] W. Gropp, E. Lusk, A. Skjellum, Using MPI: Portable Parallel Programming with the Message-passing Interface, The MIT Press, Cambridge, 1994.
[21] E. Hairer, S. Nørsett, G. Wanner, Solving Ordinary Differential Equations I, Springer-Verlag, New York, 1993.
[22] F.-N. Hwang, X.-C. Cai, A parallel nonlinear additive Schwarz preconditioned inexact Newton algorithm for incompressible Navier–Stokes equations, J. Comput. Phys. 204 (2005) 666–691.
[23] G. Jothiprasad, D. Mavriplis, D. Caughey, Higher-order time integration schemes for the unsteady Navier–Stokes equations on unstructured meshes, J. Comput. Phys. 191 (2003) 542–566.
[24] O.A. Ladyzhenskaya, The Boundary Value Problems of Mathematical Physics, Springer-Verlag, New York, 1985.
[25] J. Lewis, R. Rehm, The numerical solution of a nonseparable elliptic partial differential equation by preconditioned conjugate gradients, J. Res. Nat. Bur. Standards 85 (5) (1980) 367–390.
[26] M. Ottaviani, F. Porcelli, Nonlinear collisionless magnetic reconnection, Phys. Rev. Lett. 71 (23) (1993) 3802–3805.
[27] M. Ottaviani, F. Porcelli, Fast nonlinear magnetic reconnection, Phys. Plasmas 2 (11) (1995) 4104–4117.
[28] D.R. Reynolds, R. Samtaney, C.S. Woodward, A fully implicit numerical method for single-fluid resistive magnetohydrodynamics, J. Comput. Phys. 219 (2006) 144–162.
[29] Y. Saad, Iterative Methods for Sparse Linear Systems, SIAM, 2003.
[30] B. Smith, P. Bjørstad, W. Gropp, Domain Decomposition. Parallel Multilevel Methods for Elliptic Partial Differential Equations, Cambridge University Press, New York, 1996.
[31] H.R. Strauss, D.W. Longcope, An adaptive finite element method for magnetohydrodynamics, J. Comput. Phys. 147 (1998) 318–336.
[32] A. Toselli, O. Widlund, Domain Decomposition Methods – Algorithms and Theory, Springer-Verlag, Berlin, 2005.