Journal of Computational Physics 229 (2010) 8130–8143

Contents lists available at ScienceDirect

Journal of Computational Physics

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

XTOR-2F: A fully implicit Newton–Krylov solver applied to nonlinear 3D extended MHD in tokamaks

Hinrich Lütjens [*], Jean-François Luciani

Centre de Physique Théorique, Ecole Polytechnique, CNRS, F-91128 Palaiseau Cedex, France

ARTICLE INFO

Article history:
Received 7 July 2009
Received in revised form 15 June 2010
Accepted 9 July 2010
Available online 18 July 2010

Keywords:
Plasma
Tokamak
Magnetohydrodynamics
Nonlinear simulations
Fully implicit
Newton–Krylov
Internal kink
Thermal transport
Bifluid
Diamagnetic rotations

ABSTRACT

XTOR-2F solves a set of extended magnetohydrodynamic (MHD) equations in toroidal tokamak geometry. In the original XTOR code, the time stepping is handled by a semi-implicit method [1–3]. Moderate changes were necessary to transform it into a fully implicit one using the NITSOL library with Newton–Krylov methods of solution for nonlinear system of equations [4]. After addressing the sensitive issue of preconditioning and time step tuning, the performances of the semi-implicit and the implicit methods are compared for the nonlinear simulation of an internal kink mode test case within the framework of resistive MHD including anisotropic thermal transport. A convergence study comparing the semi-implicit and the implicit schemes is presented. Our main conclusion is that on one hand the Newton–Krylov implicit method, when applied to basic one fluid MHD is more computationally costly than the semi-implicit one by a factor 3 for a given numerical accuracy. But on the other hand, the implicit method allows to address challenging issues beyond MHD. By testing the Newton–Krylov method with diamagnetic modifications on the dynamics of the internal kink, some numerical issues, to be addressed further, are emphasized.

1. Introduction

Fusion plasma devices such as tokamaks manifest roughly two kind of behavior in the plasma core. One turbulent at small scale determining the transport coefficients and one at large scale usually referred as "magnetohydrodynamic (MHD) activity". The XTOR-2F code addresses only the second aspect.

The original semi-implicit XTOR [1–3] was developed for the study of resistive MHD activity including anisotropic thermal transport in tokamaks. But now, the challenge in the studies of global instabilities is to understand physics beyond MHD, i.e. including effects ranging from fluid corrections to MHD, known as bifluid effects, to kinetic effects due to hot ion populations such as α particles generated by fusion reactions. To progress towards that goal, the first necessary step was the transformation of the semi-implicit XTOR code into a fully implicit version able to include such improvements of the model beyond MHD. The main purpose of the present paper is to report on the performance of such a fully implicit method within the context of 3D tokamak MHD as a first step.

The MHD equations applied to tokamaks is well known to be a very stiff problem. First, there is coexistence on the same scale of very fast (compressible Alfvén) and very slow (resonant shear Alfvén) modes. Secondly, the relevant physics is slow compared to the basic time scale (the Alfvén time). The last point reflects the fact that the confinement is rather good in tokamaks. Therefore relevant studies must address long time behavior.

To address stiff problems, the implicit time advance is generally the best choice. But the solution of the tridimensional (3D) MHD equations by a fully implicit time advance method was a rather formidable task on computers of up to the

[*] Corresponding author. Tel.: +33 169334236; fax: +33 169334949.
E-mail address: Hinrich.Lutjens@cpht.polytechnique.fr (H. Lütjens).

0021-9991/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2010.07.013

2000s in full toroidal geometry. The semi-implicit XTOR code [1–3], developed from the end of 1980s, was thus optimized both for CPU time, and for memory saving. It has been extensively used both for parameter studies [5–10] and the simulation of MHD activity in experimental tokamak discharges [11–13]. But the development of this semi-implicit version is now terminated, and the semi-implicit XTOR is available for open collaborations.

The semi-implicit algorithm is well suited for second order, self adjoint problems, as turns out to be the ideal MHD. It basically adds a selective inertia, by application of a "semi-implicit operator" to each eigenvector, ensuring linear numerical stability. The "semi-implicit operator" is also chosen to dominate the nonlinear dynamics, in a sense detailed in Refs. [1,3], ensuring also the nonlinear stability. The semi-implicit algorithm appears to be far less efficient and well behaved for more general problem such as two-fluid MHD, where, for example, the diamagnetic rotations break the self adjointness. So we have concluded that even if a semi-implicit scheme could be used in some specific cases beyond MHD, it would always be prone to a lack of confidence, and that a long term investment in this direction would be questionable.

The progress both in computer power and in numerical methods now make the fully implicit approach tractable. For an implicit solution, a straightforward choice would be the use of a direct method, which requires the construction and the solution of a large matrix system corresponding to the 3D MHD problem linearized about its current position in time. Although this is probably the most robust way of solution, it remains a rather formidable task on today's computers in 3D geometries.

An alternative way is investigated since the beginning of the 2000s in 2D and reduced MHD [14–16] and in 3D solving the full resistive MHD equations [17,18]. It consists of solving every implicit time step by an iterative Newton–Krylov method. Historically, in the 1990s, this method first was applied to compressible fluid problems and Fokker–Planck equations before the first tries to solve MHD problems. A very detailed review about the development of this method and its applications to fluid and Fokker–Planck equations was published in 2004 by Knoll and Keyes [19].

Two important issues required special attention during the application of the Newton–Krylov method to 3D full MHD equations in tokamaks. The first one is to avoid the exact evaluation of the Jacobian matrix required at every implicit time advance step. For a detailed description of MHD activity in experimental tokamak plasmas, more refined models going beyond the resistive MHD model are necessary. For these models the evaluation of the Jacobian matrix can be very difficult and is circumvented by the use of so-called "matrix-free" methods [19]. In these methods the Jacobian matrix is evaluated by numerical differentiation.

The second issue is the preconditioning method in order to reduce as much as possible the number of iterations of the Newton–Krylov solvers at every time step. With an efficient pre-conditioner, the number of operations is much reduced compared to a direct iterative method, which appears generally to be roughly equivalent in CPU cost to an explicit time stepping. In the present method, we made the choice of a physical pre-conditioner, i.e. an operator based on the linearized problem to be solved. This preconditioning strategy is close to the one in Ref. [18]. The main difference is that in Ref. [18] the preconditioner is inverted by an iterative multi-grid method which is efficient in large 3D problems in Cartesian geometry, whereas in XTOR, due to basic choices, it is inverted by an exact LU method in the Fourier representation used that numerical code. These are closely related to the particularities of MHD in tokamak toroidal geometry. On one hand, large scale instabilities require only a moderate content in harmonics for a good resolution. On the other hand, the problem is particularly stiff, leading to ill conditioned matrices. In this context, the choice of the LU decomposition is recommended. This approach limits the overall time spent in preconditioning to between approximatively 1/3 and 1/2 of the total CPU time of the simulation (see Section 4.3 for more details).

In the present work, the semi-implicit full MHD code XTOR [3] was used as a toolbox to build the fully implicit code XTOR-2F. For the Newton–Krylov method, we use the NITSOL package [4]. The linearized MHD operator which was used before as the "semi-implicit operator" in Ref. [3] is used now to construct the pre-conditioner for the implicit solver.

The present paper is articulated in the following way. In Section 2, the equations solved by the present implicit version of XTOR-2F are given. The implicit numerical method is described in Section 3. The strategy used for the time stepping is presented in Section 4, together with first benchmarks. Then a comprehensive comparative convergence study between the semi-implicit and the Newton–Krylov methods is presented in Section 5, leading to an accuracy cost picture (Fig. 7). Conclusions and issues concerning bifluid simulations are presented in Section 6.

2. Physical model, geometry and discretization

2.1. Equations

At present, XTOR-2F solves the full resistive MHD equations, including anisotropic thermal transport and some neoclassical effects [3] extended by a part of the full bifluid equations that would result from the full Braginskii equations [20]:

$$\begin{aligned}
&\rho\partial_t\mathbf{v} = -\rho(\mathbf{v}\cdot\nabla)\mathbf{v} - \rho\left(\mathbf{v}_\mathbf{i}^*\cdot\nabla\right)\mathbf{v}_\perp + \mathbf{J}\times\mathbf{B} - \nabla p + (\nabla\nu\nabla)\mathbf{v},\\
&\partial_t\mathbf{B} = \nabla\times(\mathbf{v}\times\mathbf{B}) + \alpha\nabla\times\frac{\nabla_\parallel p_e}{\rho} - \nabla\times\eta(\mathbf{J}-\mathbf{J}_\mathbf{boot}),\\
&\partial_t p = -\Gamma p\nabla\cdot\mathbf{v} - \mathbf{v}\cdot\nabla p - \alpha\Gamma\frac{p_i}{\rho}\nabla p\cdot\nabla\times\frac{\mathbf{B}}{B^2} + \nabla\cdot\chi_\perp\nabla p + \nabla\cdot\left[\mathbf{B}\left(\frac{\chi_\parallel}{B^2}(\mathbf{B}\cdot\nabla)p\right)\right] + H,
\end{aligned} \tag{1}$$

with

$$\rho = Cp^{-\Gamma}, \tag{2}$$

or

$$\partial_t \rho = -\rho \nabla \cdot \mathbf{v} - \mathbf{v} \cdot \nabla \rho - \alpha \nabla p_i \cdot \nabla \times \frac{\mathbf{B}}{B^2} + \nabla \cdot D_\perp \nabla \rho + S, \tag{3}$$

The normalizations leading to Eqs. (1)–(3) are given in Ref. [3], the only new non-dimensional number is α, which basically measures the diamagnetic effects. In Eqs. (1)–(3), $\alpha = (\omega_{ci}\tau_a)^{-1}$ where ω_{ci} is the ion cyclotron frequency in the unit of magnetic field [3]. τ_a is the toroidal Alfvén time ($\tau_a = R_0/v_a$, where R_0 is the major axis of the torus and $v_a = B/\rho^{1/2}$ is the Alfvén velocity), and

$$\mathbf{v_i^*} = \alpha \frac{\mathbf{B} \times \nabla p_i}{\rho B^2}, \tag{4}$$

the ion diamagnetic velocity.

These equations are obtained from Braginskii [20], e.g. in Refs. [21–23]. The ion velocity is expanded from Braginskii's equations as $\mathbf{v_i} = \mathbf{v} + \mathbf{v}_i^* + \cdots$. The lowest order velocity $\mathbf{v}$ is defined with the perpendicular electrical drift and the parallel ion velocities, i.e. $\mathbf{v} = \mathbf{E} \times \mathbf{B}/B^2 + \mathbf{v}_{i//}$. This choice results in a Faraday equation involving $\nabla_{//}p_e$. In these equations, $\mathbf{v}_i^*$ and the cross-field heat flux contributions can be rearranged in the form given by Eqs. (2) and (3), where $p_i = p_e = p/2$ is assumed. We have not included for these first tests the electron inertia and the polarization terms which come from a higher order in the ion velocity, i.e. $\mathbf{v_i} = \mathbf{v} + \mathbf{v}_i^* + \mathbf{v_p}$, where $\mathbf{v}_p$ is the polarization drift velocity. This higher order approximation which contains e.g. effects relevant for magnetic reconnection processes (electron inertia, Hall term) will be included in the final version of XTOR-2F.

In Eqs. (1)–(4), $\mathbf{J} = \nabla \times \mathbf{B}$ is the current density field. ν and η are plasma viscosity and resistivity fields, respectively. In the transport terms of the evolution of p, T and ρ, $\chi_\perp$, $D_\perp$ and $\chi_\parallel$ are the perpendicular and parallel diffusion coefficients. H and S are the heat and density sources, respectively. The implemented choices for H are $H = -\nabla \cdot \chi_\perp \nabla p_0$ or $H = \eta J^2$, i.e. a source restoring the initial equilibrium pressure profile or an Ohmic heat source, respectively, and $S = -\nabla \cdot D_\perp \nabla \rho_0$. The subscript 0 indicates equilibrium quantities. In general, there is no parallel diffusion density coefficient in transport models. It could be included into Eq. (3) if necessary. Γ is the ratio of specific heats. XTOR-2F uses a simplified version of the bootstrap current density given by $\mathbf{J_{boot}}(t) = f_{bs}\|\mathbf{J_{boot,0}}\|(\nabla p(t))_r/p_0'\mathbf{B}(t)/\|\mathbf{B}(t)\|$ [24].

In the simulations, η varies in time according to Spitzer's resistivity law $\eta \propto T^{-3/2}$. This evolution is initialized by a mapping between equilibrium resistivity and temperature (or pressure if ρ is not evolved separately). This mapping is then used at every time step to evaluate $\eta(T(t))$. The initial η profile is obtained by the resistive equilibrium constraint demanding $\eta_0 < (J_{\Phi,0} - J_{\Phi,boot,0}) > = E_{\Phi,0}$ to be constant. $E_{\Phi,0}$ is then imposed as a boundary condition on the toroidal electric field. The transport coefficients $\chi_\perp$, $D_\perp$ and $\chi_\parallel$ can be non-constant provided that the diffusive terms in Eqs. (2), (3) are self adjoint. In the benchmarks to follow, $\chi_\perp$ is proportional to η, $D_\perp = 0$ and $\chi_\parallel$ is constant.

2.2. Basic geometrical choices used in XTOR-2F

The basic choice used in XTOR and XTOR-2F is a semi-spectral representation with linear staggered finite differences in the radial coordinate which labels the equilibrium poloidal magnetic flux surfaces, and Fourier representations both in the poloidal and the toroidal directions. For details concerning spatial discretization, we refer to Ref. [3], as no change were necessary in the basic variables between XTOR and XTOR-2F.

An important remark is that along with the Fourier representation, the modes are projected onto a restricted subset of modes. These modes are chosen to ensure a good spectral resolution of the MHD activity, and usually do not exceed several tens of poloidal modes for a given toroidal mode, and at most several tens of toroidal modes. Thus, the corresponding matrix system remains small compared to the one of the same problem but solved for instance by 2D finite elements in the poloidal plane as in Ref. [23]. Thus using direct LU decompositions for all the linear operators involved in XTOR and XTOR-2F, as described in Section 3.3, is a reasonable choice in terms of CPU performances.

The geometry of the plasma is toroidal with non-circular poloidal cross sections. It is provided trough the metric coefficients by the CHEASE code [25,26]. At present, the code has fixed boundary conditions. It requires a perfectly conducting shell at the plasma surface. Thus it cannot handle equilibria with a plasma separatrix inside the domain of solution.

3. Numerical scheme

3.1. The implicit scheme

There is a large choice of multi-step implicit numerical schemes which could be used to solve the nonlinear problem:

$$\dot{x} = F(x), \tag{5}$$

which symbolizes the set of Eqs. (1)–(3). Here we made the choice of a very simple one:

$$x_{n+1} - x_n = \Delta t F\left[\frac{x_{n+1} + x_n}{2} + \Theta(x_{n+1} - 2x_n + x_{n-1})\right], \tag{6}$$

n indexes the time step and Θ is a numerical constant. If $\Theta = 0$, the numerical time advance is a time centered scheme. With a finite value of Θ, the scheme remains second order in time and damps unresolved high frequencies. The default value in XTOR-2F is $\Theta = 1$ (as for all the benchmarks proposed in the following).

In Eq. (6), the abstract array x denotes all the variables present in Eqs. (1)–(3). These variables are treated in the same way by the implicit scheme, which results in a major simplification in the implementation of modular physical models.

As all variables are evolved in the same way, the linear transfer matrix is diagonal for every eigenvector of the linearized Eqs. (1)–(3), with eigenvalue $i\omega$. These eigenvalues can be readily determined by the following advance, where $q = \omega\Delta t$:

$$x_{n+1} - x_n = iq\left(\frac{x_{n+1} + x_n}{2} + \Theta(x_{n+1} - 2x_n + x_{n-1})\right). \tag{7}$$

The characteristic equation is

$$\lambda(\lambda - 1) = iq\left(\frac{1}{2}\lambda(\lambda + 1) + \Theta(\lambda - 1)^2\right), \tag{8}$$

where $\lambda = x_{n+1}/x_n$ is the amplification factor of the eigenmode. The scheme is linearly stable numerically if for a given value of Θ the modulus of the roots of Eq. (8) are less than 1 for every q. This is satisfied in the case $\Theta > 0$ for every q, i.e. the numerical scheme is linearly unconditionally stable. The particular case with $\Theta = 0$, i.e. the time centered scheme:

$$(\lambda - 1) = iq\frac{1}{2}(\lambda + 1). \tag{9}$$

In that case, $|\lambda| = 1$ for every q, i.e. the scheme is marginally stable.

3.2. Full MHD Newton–Krylov method of solution at every time step

The implicit method in Section 3.1 is now adapted to the system of MHD Eqs. (1)–(3).

The velocity, the magnetic, pressure fields and density fields $\mathbf{v}$, $\mathbf{B}$, p, ρ, respectively, are simultaneously advanced in time. F in Eq. (5) is the right-hand-side (RHS) of the system (1)–(3) and $x_n = (v^r_n, v^\theta_n/D, v^\phi_n, B^r_n, B^\theta_n/D, B^\phi_n, p_n)$ or $(v^r_n, v^\theta_n/D, v^\phi_n, B^r_n, B^\theta_n/D, B^\phi_n, p_n, \rho_n)$ depending on the choice of variables in Eqs. (2) and (3). $D = [\nabla r, \nabla\theta, \nabla\phi]$ is used to regularize the contra-variant θ components of the fields at the mesh axis. At every time step the system:

$$x_{n+1} - x_n - \Delta t F\left[\frac{x_{n+1} + x_n}{2} + \Theta(x_{n+1} - 2x_n + x_{n-1})\right] = 0, \tag{10}$$

must be inverted for the evaluation of x_{n+1}. Re-arranging this equation with $\Delta_n = x_{n+1} - x_n$ this can be rewritten:

$$G(\Delta_n, \bar{x}) \equiv \Delta_n - \Delta t F\left[\left\{\frac{1}{2} + \Theta\right\}\Delta_n + \bar{x}\right] = 0, \tag{11}$$

with

$$\bar{x} = (1 - \Theta)x_n + \Theta x_{n-1}. \tag{12}$$

Eq. (11) is solved for Δ_n by means of a preconditioned Newton–Krylov method, i.e.

$$M^{-1}G'\left(\Delta^k_n, \bar{x}\right)\left(\Delta^{k+1}_n - \Delta^k_n\right) + M^{-1}G\left(\Delta^k_n, \bar{x}\right) = 0, \tag{13}$$

where M is the pre-conditioner, k indexes the Newton steps in the Newton–Krylov method and prime denotes the derivative with respect to Δ_n. At every Newton step, a linear system of equation is solved by a Krylov iterative method. The Newton Iterative SOLver (NITSOL) library [4] is used for this purpose. It gives the choice between three transpose free Krylov methods, generalized minimum residual (GMRES (m)), bi-conjugate gradient stabilized (BiCGSTAB) and transpose free quasi-minimum residual (TFQMR). All three were tested with XTOR-2F, and using the default settings in NITSOL, a twice as fast convergence is observed with GMRES (20) as with the two other methods, BiCGSTAB and TFQMR.

$G'\left(\Delta^k_n, \bar{x}\right)$ is the MHD operator linearized about $\bar{x}$. In NITSOL, this operator can be user provided or interpolated numerically. The latter case is the so-called "matrix-free" method. A comparison was performed between both approaches in ideal MHD with XTOR-2F, showing that both methods required quite comparable number of times the evaluation of the RHS in Eqs. (1)–(3) for the same nonlinear simulation. After this successful test, we have decided to use exclusively the matrix-free method, thus avoiding explicit linearization of non-ideal terms in Eqs. (1)–(3).

3.3. Choice of the pre-conditioner in XTOR-2F

The pre-conditioner used in XTOR-2F is based on the equations solved by the code, Eqs. (1)–(4). M^{-1} is the product:

$$M^{-1} = M^{-1}_\eta M^{-1}_D M^{-1}_\chi M^{-1}_{\mathrm{MHD}}, \tag{14}$$

with

$$\begin{aligned} M_{\mathrm{MHD}} &= (L_{\mathrm{MHD}}, 1), \\ M_{\eta} &= (1, L_{\eta}, 1, 1), \\ M_{\chi} &= (1, 1, L_{\chi}, 1), \\ M_D &= (1, 1, 1, L_D). \end{aligned} \tag{15}$$

Here, with $\widetilde{\Theta} = 1/2 + \Theta$ and 0 indexing equilibrium quantities:

$$\begin{aligned} L_{\chi} &= 1 - \widetilde{\Theta}\Delta t \left\{ \nabla \cdot \chi_{\perp}^{0} \nabla + \nabla \cdot \left[\mathbf{B_0} \left(\frac{\chi_{\parallel}^{0}}{B_0^2} (\mathbf{B_0} \cdot \nabla) \right) \right] \right\}, \\ L_D &= 1 - \widetilde{\Theta}\Delta t \nabla \cdot D_{\perp}^{0} \nabla, \\ L_{\eta} &= 1 - \widetilde{\Theta}\Delta t \nabla \times \eta^0 \nabla \times, \end{aligned} \tag{16}$$

and L_{MHD} is the ideal (one fluid) linear MHD stability operator including a viscosity term, i.e.

$$L_{\mathrm{MHD}} = \begin{pmatrix} g_{ij}^0 - \widetilde{\Theta}\Delta t (\nabla v^0 \nabla) & -\widetilde{\Theta}\Delta t F_v^{0B} & -\widetilde{\Theta}\Delta t F_v^{0p} \\ -\widetilde{\Theta}\Delta t F_B^{0v} & 1 & 0 \\ -\widetilde{\Theta}\Delta t F_p^{0v} & 0 & 1, \end{pmatrix}, \tag{17}$$

where

$$\begin{aligned} F_v^{0B} \boldsymbol{\delta}\mathbf{b} &= (\nabla \times \mathbf{B_0}) \times \boldsymbol{\delta}\mathbf{b} + (\nabla \times \boldsymbol{\delta}\mathbf{b}) \times \mathbf{B_0}, \\ F_v^{0p} \delta p &= -\nabla(\delta p), \\ F_B^{0v} \boldsymbol{\delta}\mathbf{v} &= \nabla \times (\boldsymbol{\delta}\mathbf{v} \times \mathbf{B_0}), \\ F_p^{0v} \boldsymbol{\delta}\mathbf{v} &= -\boldsymbol{\delta}\mathbf{v} \cdot \nabla p_0 - \Gamma p_0 \nabla \cdot (\boldsymbol{\delta}\mathbf{v}) \cdot \end{aligned} \tag{18}$$

$\left(g_{ij}^0\right)$ is the equilibrium metric tensor. L_{MHD} is triangular in Faraday's and the pressure equation. Thus, inverting:

$$L_{\mathrm{MHD}} \begin{pmatrix} \boldsymbol{\delta}\mathbf{v} \\ \boldsymbol{\delta}\mathbf{B} \\ \delta p \end{pmatrix} = \begin{pmatrix} \mathbf{F_v} \\ \mathbf{F_B} \\ F_p \end{pmatrix}, \tag{19}$$

where $(\mathbf{F_v}, \mathbf{F_B}, F_p)^{\top}$ is the RHS of Eqs. (1)–(3), one obtains:

$$\begin{aligned} \boldsymbol{\delta}\mathbf{v} &= L_0^{-1} \left\{ \mathbf{F_v} + \widetilde{\Theta}\Delta t \left[F_v^{0B}(\mathbf{F_B}) + F_v^{0p}(F_p) \right] \right\}, \\ \boldsymbol{\delta}\mathbf{b} &= \mathbf{F_B} + \widetilde{\Theta}\Delta t F_B^{0v}(\boldsymbol{\delta}\mathbf{v}), \\ \delta p &= F_p + \widetilde{\Theta}\Delta t F_p^{0v}(\boldsymbol{\delta}\mathbf{v}), \end{aligned} \tag{20}$$

with

$$L_0 = g_{ij}^0 - \widetilde{\Theta}\Delta t (\nabla v^0 \nabla) - \left(\widetilde{\Theta}\Delta t\right)^2 \left(F_v^{0B} F_B^{0v} + F_v^{0p} F_p^{0v} \right), \tag{21}$$

and $F_v^{0B}, F_v^{0p}, F_B^{0v}$ and F_p^{0v} defined in Eq. (18). In Eq. (21), one recognizes an operator which only departs from the numerically stabilizing operator used in the semi-implicit method of solution in Ref. [3] by the numerical factors multiplying its different terms. The semi-implicit operator in Ref. [3] indeed reads :

$$L_0^{SI} = g_{ij}^0 - c\nabla^2 - \Delta t_0^2 \left(F_v^{0B} F_B^{0v} + F_v^{0p} F_p^{0v} \right), \tag{22}$$

where the numerical constant c is required for the nonlinear stability of the semi-implicit scheme (see Ref. [3], Section 3.4) and Δt_0 is the fixed time step chosen to evaluate and LU decompose $L_0^{SI} \cdot L_0$ was therefore readily implemented in the present method as a part of the pre-conditioner M. We recall that L_{MHD} (as L_0), L_{η}, L_{χ} and L_D are inverted by a standard block LU decomposition in Ref. [3]. This is unchanged in XTOR-2F at moment.

Contrary to Ref. [3] where these terms are advanced in time by a split method, the thermal conduction, the density transport and the resistive diffusion parts of the equations are integrated in the Newton–Krylov process. This appears to be necessary for numerical stability. To avoid any ambiguity, we emphasize that the operator splitting in Eq. (14) concerns only the pre-conditioner.

4. Tuning the Newton–Krylov method

First, the rules used in XTOR-2F to control the iteration numbers of the Newton–Krylov method is described in Section 4.1. These rules are then applied to the nonlinear evolution of an internal kink in ideal MHD in Section 4.2. For this purpose, we use as initial conditions an equilibrium provided by the CHEASE code [25,26] which is unstable towards such an instability (the low shear q-profile in Ref. [27,28]). The internal kink mode is a good test case because it shifts quickly the plasma core away from the discretization mesh center, which is rather demanding numerically.

In the studies to follow, the numerical cost of both the semi-implicit and the implicit method are compared by means of the number of operations. A good estimation is obtained by the number of evaluations of $M^{-1}F$ where F is the RHS of Eqs. (1)–(3) and M is the pre-conditioner given by Eq. (14) (in ideal MHD, the resistive and the thermal transport contributions to M are set to identity). The amount of extra operation due to the construction and the LU decomposition of the pre-conditioner or the semi-implicit operator indeed remains small, and will be discussed in Section 4.3.

To complete this study, CPU consumption are presented in Section 4.3 with a 3.2 Ghz bi-quadricore Intel Xeon 5100 "Woodcrest" processor.

We emphasize that for all the cases in Section 4.1 and later in Section 5, $\alpha = 0$ since the diamagnetic terms could not be handled in a satisfactory way by the semi-implicit XTOR, unless allowing intractable small time steps.

In the following benchmarks, two different resolutions are used.

- *Low resolution case:* mesh of 150 radial, 12 toroidal and 32 poloidal points. After de-aliasing and projection on selected modes, the spectral content is 4 toroidal modes including $n = 0$ and a band of 12 poloidal modes for every toroidal mode.
- *High resolution case:* mesh of 300 radial, 24 toroidal and 64 poloidal grid points. After de-aliasing and projection on selected modes, the spectral content is 8 toroidal modes including $n = 0$ and a band of 24 poloidal modes for every toroidal mode.

4.1. Optimizing time step and Newton–Krylov iterations

The pre-conditioner presented in Section 3.3 is very efficient in the linear phase of the evolutions. But in regimes where the dynamics departs from the linear one, the Newton–Krylov solver iterates significantly more both on the Newton and the Krylov method. It appears that an optimum is obtained when the product of Newton and Krylov iterations is about 10. Indeed, we observed that the solver converges quickly in the first 10 iterations. Then, the full convergence to the "exit convergence criterion" is slower, and can even be oscillatory. In order to keep the solver in this optimal range, the time step is modified automatically, thus inducing a reevaluation of the pre-conditioner with the new time step. This behavior will be illustrated in Section 4.2. After a number of tests with different kind of instabilities, the default setup in the code is the following:

- If the number of RHS evaluations of Eqs. (1)–(4) exceeds 15 the time step is reduced by 3/4.
- If the number of RHS evaluations of Eqs. (1)–(4) is 3 the time step is multiplied by 6/5.

A range of iteration numbers is kept between increasing and decreasing the time step (here 3 and 15, respectively) to avoid a back and forth switching behavior since each time step change requires the construction and LU decomposition of the pre-conditioner. Note that these numbers are user provided and can be tuned for a given problem. The exit convergence criterion for the Newton–Krylov method in XTOR-2F is identical to the one chosen by the NITSOL algorithm [4], i.e. it acts simultaneously on the inexact Newton and the Krylov method. In XTOR-2F, the convergence tolerance with respect to the function G in Eq. (11) given by

$$\mathrm{tol} = \epsilon \max\left(10^{-3},\ \|\Delta_n\|\right), \tag{23}$$

where ϵ is a user provided constant and $\Delta_n = x_{n+1} - x_n$ is the converged field increment obtained by the Newton–Krylov method. $\|\ldots\|$ is the Euclidean norm (ℓ_2 norm) of a vector. An intensive use of the code has shown that a good choice for XTOR-2F simulations is $\epsilon = 10^{-4}$. When $\|\Delta_n\|$ is larger than 10^{-3} this ensures that the relative error on the time evolution is smaller than ϵ. For example the relative error on a growth rate is smaller than ϵ. Furthermore the cut-off 10^{-3} in Eq. (23) was added to avoid a useless number of iterations during the beginning of the simulation which starts from numerical noise. The effect of ϵ on the convergence of the solution will be described later in Section 5.1 within the more general framework of resistive MHD including thermal transport.

4.2. Application to an ideal internal kink

As a first benchmark, the efficiency of the method in Section 4.1 used to control the iterations of the Newton–Krylov solver by a modification of the time step is tested.

The initial equilibrium profiles have the same shape as shown in Fig. 2 in Ref. [3], except that the pressure has been rescaled to destabilize the ideal MHD internal kink (the poloidal β at the $q = 1$ surface is $\beta_{p,\,q=1} = 0.21$ [29]). For these benchmarks, the density is not evolved separately. $\eta = 0$ and there are no diamagnetic rotations in Eqs. (1)–(4), i.e. $\alpha = 0$.

Fig. 1 shows the evolution of the kinetic energies of the toroidal modes both for the low and the high resolution discretization grids for an ideal internal kink instability. Note that at every time step the energy of every toroidal mode number n is the Euclidean norm of the energies of all poloidal (m) modes and radial nodes in the simulation. In Fig. 2(a) the accumulated

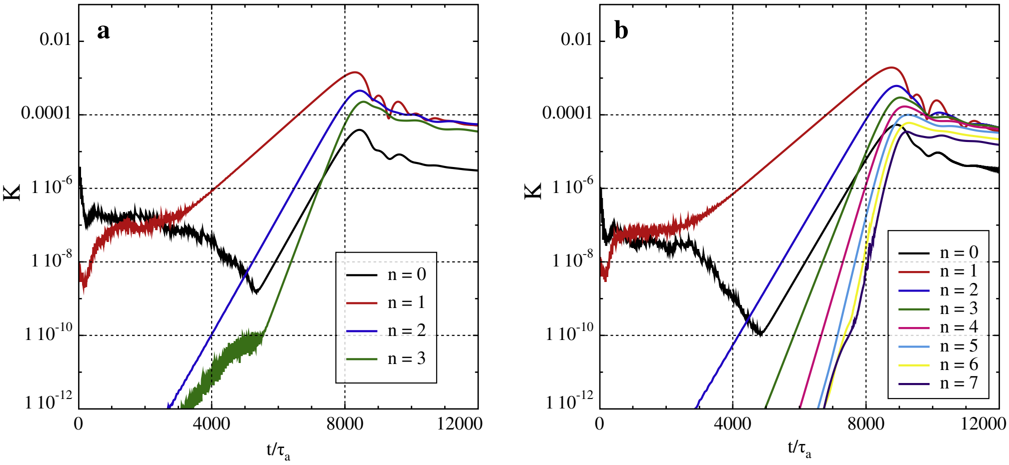

Fig. 1. Kinetic Energies of the toroidal modes for the ideal internal kink test case. (a) Low resolution mesh. (b) High resolution mesh.

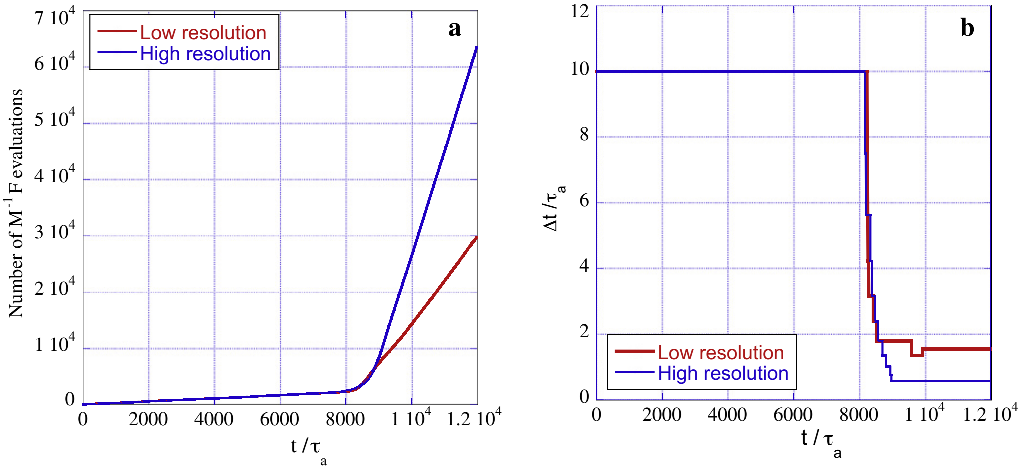

Fig. 2. (a) Accumulated number of $M^{-1}F$ evaluations for the ideal internal kink with the small and the large discretization grid. (b) Evolution of the time step for the same two simulations.

number of $M^{-1}F$ evaluations are presented for these two test cases. For both cases, the tuning rules given in Section 4.1 are applied with $\epsilon = 3 \times 10^{-3}$ in the Newton–Krylov exit tolerance Eq. (23).

Fig. 2(a) shows that with both spatial resolutions, the accumulated number of $M^{-1}F$ evaluations are comparable in the linear growth phase of the simulation until approximatively $t = 8000\tau_a$. During this phase, the time step is not allowed to exceed $\Delta t = 10\tau_a$ (this is a user choice). As can be observed in Fig. 2(b), it remains unchanged. In the nonlinear phase after $t = 8000\tau_a$, the number of accumulated $M^{-1}F$ evaluations increases consequently because on one hand the dynamics of the internal kink departs significantly from the linear one, or equivalently the RHS of Eqs. (1)–(3) departs significantly from its linear counterpart, which is used for preconditioning via M^{-1} in Eq. (14). On the other hand, the time step is reduced significantly by the rules in Section 4.1, as can be observed in Fig. 2(b). The number of $M^{-1}F$ increases even more when the discretization mesh is refined, because details in the magnetic topology appear with higher harmonics in m and n which are increasingly more difficult to resolve. Fig. 2(a) shows indeed that the accumulated number of $M^{-1}F$ evaluations grows about twice as fast for the high resolution case compared to the low one in the nonlinear phase of the simulation, reaching a total of about 6.5×10^4 at the end of the simulation.

One can observe that with both discretization meshes the numerical noise coming from the residuals of the Newton–Krylov solver in the first $5000\tau_a$. This noise is confined to a low value for every toroidal harmonic, without spreading, or accumulation in time.

In order to determine an optimum time step for the fully implicit method of solution, a series of simulations are shown in Fig. 3 with the small discretization mesh, keeping the time step Δt and the pre-conditioner M fixed throughout the entire

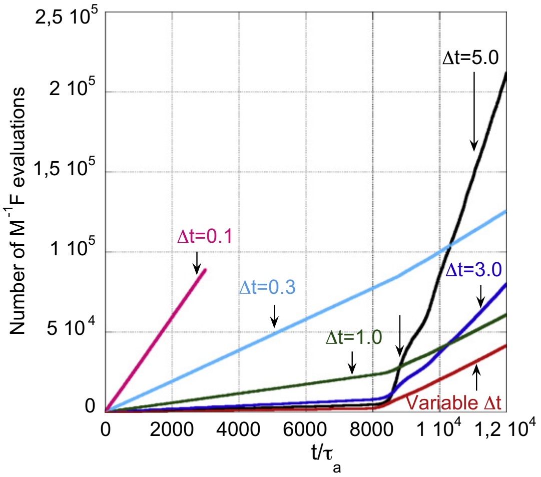

Fig. 3. Accumulated number of $M^{-1}F$ evaluations for the ideal internal kink with the small discretization grid and fixed time steps in the Newton–Krylov solver.

run. These simulations are compared with a run where the time step is variable and controlled by the rules in Section 4 (red curve). An inspection of Fig. 3 shows that for $\Delta t \leqslant \tau_a$, the method becomes already inefficient in the linear phase of the simulation (i.e. the method needs more evaluations of the RHS of Eqs. (1)–(3) for one Alfvén time than required). On the other hand, with $\Delta t > \tau_a$ the method is inefficient in the nonlinear part of the simulation. It can be observed that with the step adjustment rules in Section 4, the behavior of the Newton–Krylov solver is close to its optimum in both the linear and the nonlinear part of the dynamics. An extensive use of the code for production purposes also supports this observation [30].

4.3. Some remarks about CPU consumption

The CPU cost with the small and the large discretization meshes are detailed here for both the ideal kink simulations in Section 4.2 and the resistive internal kink simulations including thermal transport used later for the convergence studies in Section 5. The CPU times given here are of course processor depending (here we use one core of a 3.2 Ghz bi-quadricore Intel Xeon 5100 "Woodcrest" processor). But, these times indicate the fractions of time spent in the different parts of the code during a nonlinear simulation, which are relatively insensitive to the kind of processor used for the calculation.

In Table 1, the time spent in the construction and the LU decomposition of the different matrices involved in the pre-conditioner in Section 3.3 are shown. The pre-conditioner is reevaluated and stored at every change in the time step according to the rules in Section 4. These operations only occur a moderate number of times during a nonlinear simulation, e.g. 40 times for the large grid resistive benchmark in Section 5, and thus only cause a small overhead in CPU consumption.

Table 2 shows the time spent in the different operations required for the evaluation of Eq. (13) for both discretization mesh sizes. The "Initialization" row is the CPU time used for the initialization of the nonlinear solver. It occurs only once

Table 1
CPU seconds used for the pre-conditioner matrix construction and LU decomposition.

Grid size	L_{MHD}		L_η		L_χ	
	Small	Large	Small	Large	Small	Large
Construction	13.82	518	2.82	117	0.76	31
LU	0.09	2.32	0.055	1.29	0.007	0.11

Table 2
CPU seconds used for the different step in the evaluation of Eq. (13).

Grid size	Ideal kink		Resistive kink with transport	
	Small	Large	Small	Large
Initialization	0.088	0.59	0.115	0.782
F = RHS of in Eqs. (1)–(3)	0.061	1.89	0.174	2.19
M^{-1} multiplication	0.064	1.27	0.101	1.29
NITSOL offset	0.019	0.25	0.033	0.28

per XTOR-2F time step. The next two rows indicate the CPU time spent first in the evaluation of the RHS of Eqs. (1)–(3), F, and second in the multiplication by M^{-1}. The last row, "NITSOL offset" gives the CPU time used internally by the nonlinear solver without the time used to evaluate $M^{-1}F$. The 3 last operations are performed at every Krylov iteration.

Combining the CPUS times in Tables 1 and 2 shows that for simulations as in Sections 4.2 and 5.1, the overall time spent in preconditioning is between approximatively 1/3 and 1/2 of the total CPU time of the simulation. This is a good ratio, which is obtained because of the direct LU decomposition method used to invert the pre-conditioner. This would be different if iterative methods were used for preconditioning, because in that case, to our knowledge the preconditioning would dominate the CPU cost of the simulation.

5. Convergence studies of the implicit the semi-implicit schemes

For the purpose of this study, plasma resistivity and thermal conduction are included in the model, and the same initial equilibrium is used as in Section 4.2, but with a rescaled plasma pressure so that $\beta_{p,\ q=1} = 0.14$ (this equilibrium is ideal MHD stable). Two series of tests are provided. First, the convergence of the implicit solver is tested by changing the numerical tolerance on the convergence criterion of the Newton–Krylov method, Eq. (23). Secondly, this study is completed by a comparative convergence test between the implicit and the semi-implicit solution by means of the parameters Δt_0 and c in the semi-implicit operator (22). This work allows a numerical cost/accuracy analysis comparing the semi-implicit and the implicit schemes.

It is important to remark here that reducing ϵ in Eq. (23) for the implicit scheme and reducing Δt_0 and c in Eq. (22) amounts to reduce the time step in both methods. For the former method, this occurs because of the tuning rules in Section 4.1 and for the latter because of the numerical stability criterion Eq. (18) in Ref.[3]. But it is important to note that the time step varies in time for both numerical schemes, and that in the internal kink simulations the strongest time step reduction occurs when the instability saturates.

It is also emphasized that solving the resistive MHD equation with thermal transport is a much more challenging problem numerically than the ideal MHD simulations as shown in Section 4.2, since the plasma core is entirely turned over.

In the simulations presented in here, the initial resistivity profile is specified by $\eta_0 J^0_\Phi = E_{//}$ where $E_{//}$ is the constant parallel electric field (see also Eq. (28) in Ref. [3]). Here, $\eta_0 = 10^{-6}$ at the magnetic axis. The resistivity profile evolves in time during the simulation with the temperature profile according to Spitzer's law. The parallel thermal conductivity profile is constant in space and time, $\chi_{//} = 100$. The perpendicular thermal conductivity profile is proportional to the initial resistivity profile, $\chi_\perp \propto \eta_0$ and at the magnetic axis, $\chi_\perp = 10^{-5}$. It is also kept constant in time.

All the implicit and semi-implicit simulations performed for our convergence study used the high resolution grid detailed in Section 4. This is necessary for the resolution of the current sheet driving the reconnection.

5.1. Convergence of the implicit scheme versus convergence tolerance of the Newton–Krylov solver

Fig. 4(a) shows the time evolution of the kinetic energies of the toroidal modes for the implicit scheme. This case ran with a tolerance $\epsilon = 10^{-5}$ in the convergence criterion (23). In the convergence test to follow, simulations were done with four different values of ϵ, i.e. $\epsilon = 10^{-5}$, 10^{-4}, 3×10^{-4} and 10^{-3}.

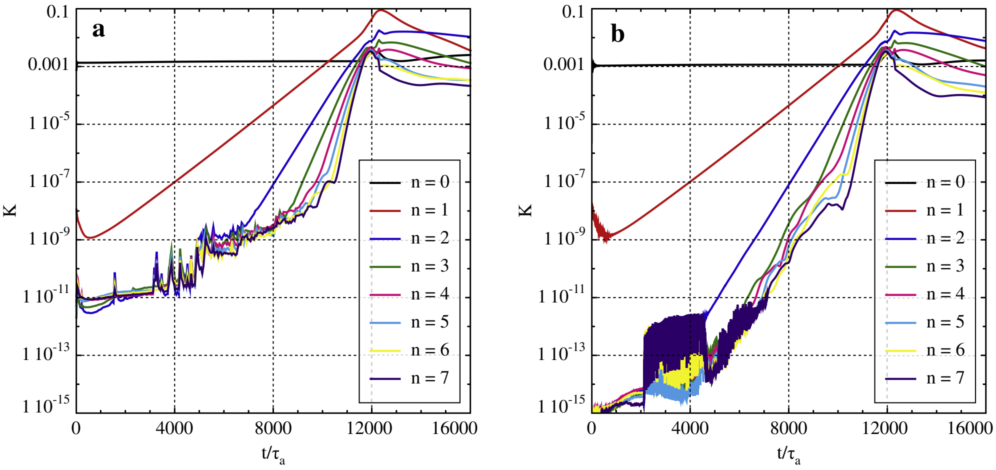

Fig. 4. Kinetic Energies of the toroidal modes for the resistive internal kink test case including thermal transport with the large discretization mesh, (a) the implicit scheme and $\epsilon = 10^{-5}$, (b) the semi-implicit scheme with $\Delta t_0 = 0.3$ and $c = 2 \times 10^{-6}$ in Eq. (22).

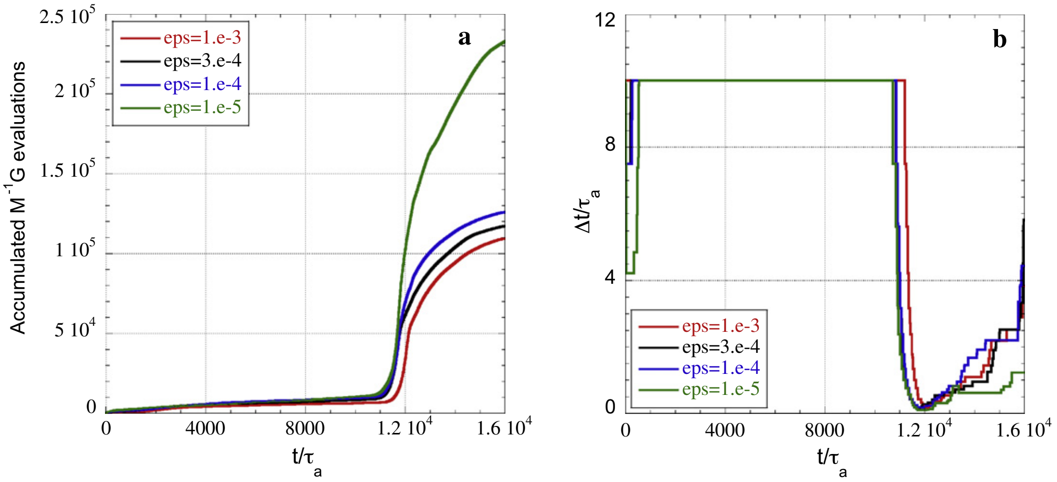

Fig. 5. (a) Accumulated number of $M^{-1}F$ evaluations for the resistive internal kink with thermal transport and the fully implicit Newton–Krylov solver, the large discretization mesh. (b) Evolution of the time step Δt according to the rules in Section 4 for the cases in Fig. 5(a).

The time evolution of the accumulated number of $M^{-1}F$ evaluations is presented in Fig. 5(a) for the 4 values of ϵ chosen in our study. The time coordinate is the same as in Fig. 4. It can be observed that the number of $M^{-1}F$ evaluations for the fully implicit method is only about doubled when compared to the ideal simulations in Fig. 2(a) with the high resolution mesh. This shows the good behavior of the implicit solver, even when the thermal transport is included in the dynamics. The fast parallel thermal diffusion penalizes only moderately the implicit scheme. It can be observed that the evolutions remain close for the three higher values of ϵ, whereas the curve with $\epsilon = 10^{-5}$ breaks away from the other ones. The number of $M^{-1}F$ evaluations increases indeed by about 50% in the latter case.

Fig. 5(b) shows the time evolutions of the time step piloted by the rules in Section 4.1 for the same 4 cases as in Fig. 5(a). The maximum time step of $10\tau_a$ allowed during the simulation is user specified. It is reached during the linear growth phase of the instability. The variation at the beginning of the run is due to the set up of the $n = 0$, $m = 1$ Pfirsch–Schlüter convection cells. At the sawtooth crash at about $t = 11{,}500\tau_a$, the time step drops significantly and eventually recovers slowly. The latter behavior is due to the perpendicular thermal source, which allows the equilibrium core plasma pressure to recover after the sawtooth crash and thus the pre-conditioner to become more efficient again.

For a quantitative comparison between these 4 simulations, the numerical error is measured by means of the relative error on the magnetic field:

$$\delta B = \frac{\|\mathbf{B}(t) - \mathbf{B}_{\rm ref}(t)\|}{\|\mathbf{B}_{\rm ref}(t)\|}. \qquad (24)$$

$\mathbf{B}_{\rm ref}(t)$ is the magnetic field at time t of a reference case and $\|\ldots\|$ is the Euclidean norm. In the convergence studies to follow, the ***reference case*** is the implicit high resolution case solved with a Newton–Krylov tolerance $\epsilon = 10^{-5}$ in Eq. (23) shown in Fig. 4(a). This choice is due to the fact that the Newton–Krylov method with $\epsilon = 0$ indeed solves the physical problem on a given discretization mesh.

Fig. 6(a) shows the time evolution of $\delta\, B$ for $\epsilon = 10^{-3}$,3×10^{-4} and 10^{-4}. It can be observed that δB is significantly reduced between the two highest values of ϵ (red and black curves), and that the gain is moderate between $\epsilon = 3 \times 10^{-4}$ and 10^{-4} (black and blue curves). This explains both the evolution of the accumulated number of $M^{-1}F$ evaluations and of the time step in Fig. 5(a) and (b), respectively. For $\epsilon = 10^{-4}$, the solution is already well converged, and decreasing that value does only slightly reduce the numerical noise level in the solution, but significantly increases the number of iterations of the Newton–Krylov solver.

This relative insensitiveness of the result with respect to ϵ (of course, its value must be reasonably small) is very important for the robustness of the Newton–Krylov method, because it shows that in all the phases of the evolution, there is only little interference between the numerical noise of the solver and the physical modes. Indeed, the physical high m and n dynamics is not polluted by the numerical high mode oscillations of the solver. This was a rather surprising result for us.

5.2. Comparative convergence study between the semi-implicit and the implicit scheme

Fig. 4(b) shows the time evolution of the kinetic energies of the toroidal modes for the same case as in Fig. 4(a), but solved by the semi-implicit method in Ref. [3]. This simulation was performed with $\Delta t_0 = 0.3$ and $c = 2 \times 10^{-6}$ in Eq. (22). Except for the low magnitude numerical noise in the initial phase of the simulation, the semi- and the implicit results look similar. In the following, this observation will be quantified by a convergence study.

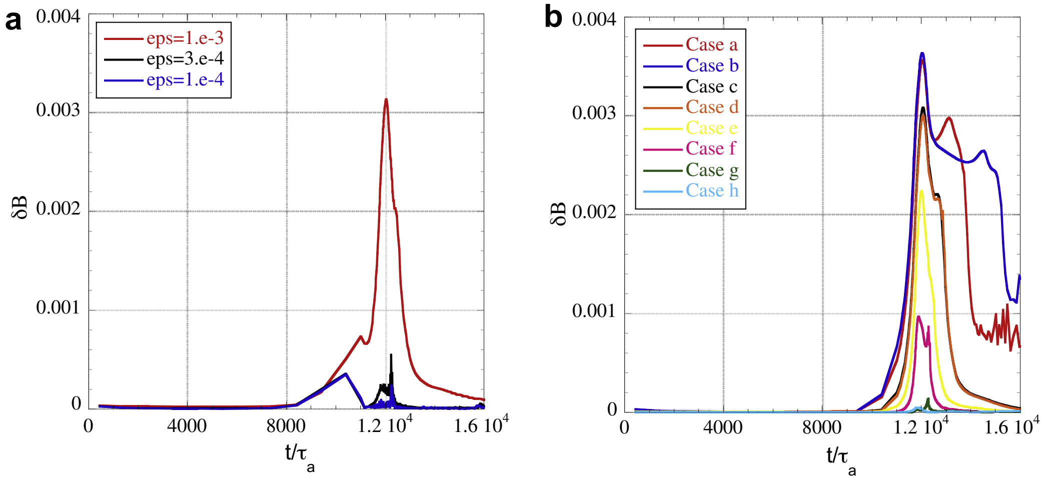

Fig. 6. Time evolution of the numerical error on the magnetic field (a) between the reference case $\epsilon = 10^{-5}$ and different values of ϵ, (b) between the implicit reference case $\epsilon = 10^{-5}$ and the semi-implicit cases in Table 3.

The semi-implicit method is known to be very sensitive to the shape of the semi-implicit operator used for the numerical stabilization of the scheme. This comes from the impact of this operator on the mode dynamics. In Eq. (22), the ideal MHD part induces a second order in time correction, and the $c\nabla$ term even a zeroth order correction. As mentioned in Ref. [3], c must be small compared to the ideal MHD part of the operator, but is required for numerical stability in the nonlinear phase of the simulation.

In Table 3, the values of Δt_0 and c in the semi-implicit operator (22) used for the convergence study are summarized. All these simulations will be compared with the same reference case as in Section 5.1 by means of δB in Eq. (24), i.e. the implicit simulation with $\epsilon = 10^{-5}$ in Eq. (23).

The semi-implicit method requires one $M^{-1}F$ evaluation per time step. We remind that the time step of this method is significantly reduced during the nonlinear phase of the simulation because of the CFL conditions at the magnetic axis (Eqs. (16) and (18) in Ref. [3]).

A problematic issue for a quantitative comparison of the implicit and the semi-implicit scheme arises from the effects of c and Δt_0 in Eq. (22) mainly in the fast linear phase of the simulations. Without any correction, one would only measure the numerical drift between the two methods of solutions. To avoid this problem, it was decided to calibrate the time evolutions to the end of the linear growth phase of the internal kink.

In Fig. 6(b), the time evolutions of δB defined in Eq. (24) are presented for the semi-implicit cases in Table 3. In all the simulations, the maximum error occurs when the mode saturates. For a large Δt_0, the error remains large in the last phase of the simulation (blue and red curves, cases a and b in Table 3, respectively). The case shown in Fig. 4(b) corresponds to the smallest error in Fig. 6(b), $\delta B_{\max} \approx 5 \times 10^{-5}$ (light blue curve, case h in Table 3).

To complete this convergence study, the numerical cost versus accuracy for all the simulations in Sections 5.1 and 5.2 are evaluated. Here, the numerical cost is the accumulated number of $M^{-1}F$ evaluations at $t = 16{,}000\tau_a$, the end of the simulation. The maximum errors, $\delta B_{\max}$, are evaluated for every simulation from Fig. 6(a) and (b). The results are presented in Fig. 7. It can be observed that with a small error (say $\delta B_{\max} \leqslant 10^{-3}$) the numerical cost is increased by about a factor 3 for the implicit scheme compared to the semi-implicit one. In Fig. 7, one also observes that the semi-implicit method allows quick computations with large errors, useful for preliminary studies.

Table 3
Semi-implicit resistive internal kink test cases including thermal transport with the large discretization mesh used in the convergence study. Δt_0 and c are the coefficients in the semi-implicit operator, Eq. (22).

Runs	Δt_0 in Eq. (22)	c in Eq. (22)
Case a	10	3×10^{-4}
Case b	10	3×10^{-3}
Case c	3	3×10^{-5}
Case d	2	3×10^{-5}
Case e	1	3×10^{-5}
Case f	0.5	3×10^{-5}
Case g	0.3	3×10^{-5}
Case h	0.3	2×10^{-6}

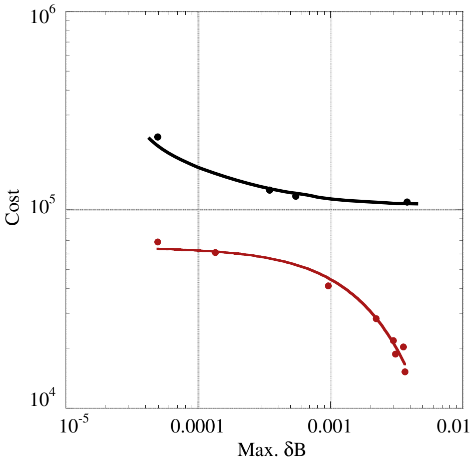

Fig. 7. Cost versus maximum error on δB for the semi-implicit cases in Fig. 6(b) (red curve) and the implicit cases in Fig. 6(a) (black curve). $\delta B_{\max}$ are obtained from Fig. 6(a) and (b). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

From both the ideal and the fully resistive MHD simulations, our conclusion is that the Newton–Krylov implicit method, when applied to basic one fluid MHD is about a factor 3 more costly than the semi-implicit one for the same numerical accuracy. This confirms that semi-implicit methods were well suited in the one fluid MHD case. But we emphasize again that the current state of research is to develop numerical simulations beyond MHD, which in our opinion can not rely on semi-implicit like methods.

6. Discussion and perspectives

The original semi-implicit version of the 3D MHD XTOR code presented in Ref. [3] was successfully transformed into a fully implicit one for the extended MHD code XTOR-2F. For the solution of the implicit problem required at every time step, a preconditioned Newton–Krylov method was used. In the version of the code presented here, the pre-conditioner is a product of the linearized ideal (one fluid) MHD, resistive diffusion and thermal diffusion operators.

The method was applied to internal kink simulations with increasing complexity of the physical model. A comparative study between the fully implicit method presented in this work and the semi-implicit method used in Ref.[3] shows a good agreement between both methods of solution. The number of operations (and therefore the CPU time) is increased by a factor of about 3 for a good numerical accuracy with the fully implicit method compared to the semi-implicit one for problems which can be solved by both methods.

We were surprised by the good behavior of the fully implicit solver concerning two very important issues. The first is the numerical noise. It appears that, even in deeply nonlinear phases, the large m or n harmonics not involved in the dynamics stay at a very low level, well under the one that would arise from an uniform distribution of the Newton–Krylov residuals. The second point is the good behavior of the ”matrix-free” option. This allows to implement physical models involving complicated algebra without the need of providing the Jacobian, or even to implement physical constraints which are intrinsically implicit.

These results confirm a posteriori the optimal character of the semi-implicit schemes for second order, self adjoint problems. But it is presently most important to go beyond MHD, by the implementation of more and more refined physical models. This needs a robust and unquestionable numerical method, and this cannot be done, in our opinion, with a semi-implicit method. For instance, the XTOR-2F code has been conceived to be ready for the inclusion of kinetic or fast ions module, resulting in an hybrid fluid/particles code.

For typical grid sizes as the one used for the large test cases in the present work, the sizes of the matrices remain reasonably small because of the Fourier representations used in both the angular directions of the torus. For this reason, not much time was spent in the parallelization of the code. However, if in the future, for some reason substantially larger grid sizes are required, as a first step the code can straightforwardly be parallelized on the toroidal mode number n for all the algebra and the number of multiple FFT’s done in 1D or in 2D. Efficient parallelization on a massively parallel machine requires a lot more work, which is let for future at moment.

As stated above, the principal goal in the development of XTOR-2F is the inclusion of physical effects beyond MHD. The code is currently used for comprehensive studies including bifluid effects both for internal kink and tearing modes. Here, only some issues to be solved are addressed.

We show by preliminary results that the Newton–Krylov method is indeed able to handle bifluid MHD. It is emphasized that we never managed to include successfully these effects in the semi-implicit XTOR code for two reasons. First ion diamagnetic rotations were strongly affected by the scheme, unless working with a detrimental small time step. Secondly

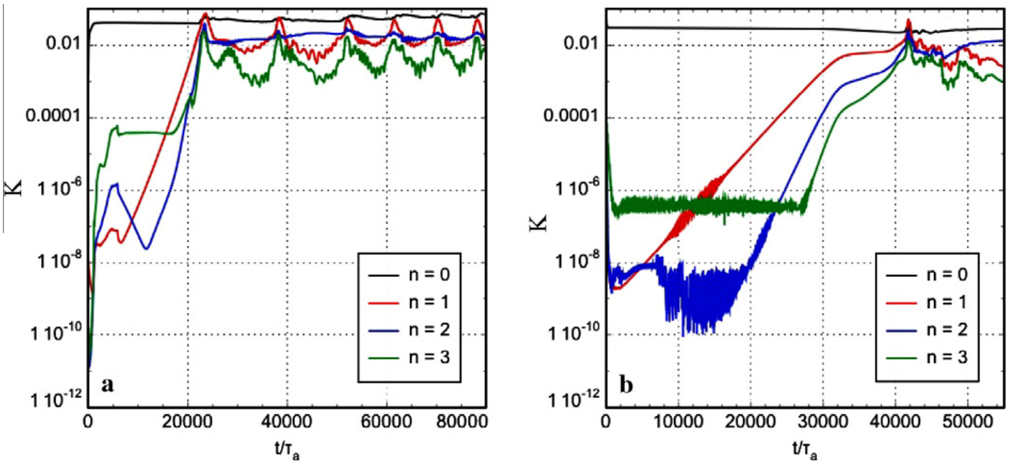

Fig. 8. Kinetic Energies of the toroidal modes for the small discretization mesh for the ideal internal kink test case the fully implicit Newton–Krylov. (a) Resistive MHD with pressure evolution and ion diamagnetic rotation. (b) Same case as (a) but supplemented with electron diamagnetic rotation.

the electron diamagnetic modifications to Faraday's equation led to new stiffness which could not be taken into account by the semi-implicit treatment.

Two internal kink test cases are used here with the small discretization mesh. In the first one, the internal kink simulation including plasma resistivity and thermal transport of Section 5 is supplemented by ion diamagnetic rotations. The electron diamagnetic rotations ($\nabla_{//} p_e = 0$) are artificially set to zero in Faraday's Eq. (2). Only the pressure is evolved in time. The second one is the same case, but it moreover includes these electron diamagnetic rotations and separate pressure and the density time evolutions.

In the first test case, Fig. 8(a) shows kinetic energies for a simulation of several sawtooth cycles. Inspecting the kinetic energy evolution of the first growth phase shows that the internal kink is significantly stabilized compared to the ideal and resistive cases. Here, normalized to the toroidal Alfvén time the growth rate $\gamma = 2 \times 10^{-3}$ and $\omega^{*} i = 7.8 \times 10^{-3}$ at the $q = 1$ surface.

In Fig. 8(b), time evolutions of the kinetic energies are presented for the second test case. As expected, the ω_e^{*} stabilization is strong on a resistive mode. It results in a plateau phase in Fig. 8(b) where the core pressure slowly spirals outward. This is followed by another phase with a much faster reconnection process. This behavior, similar to the sawtooth ramp and crash, is coherent with the expected diamagnetic physics.

These two computations were selected from a general study of sawtooth oscillations, which will be published elsewhere and which goes beyond the scope of the present paper.

In Fig. 9, the blue curve shows the number of $M^{-1}F$ evaluations where F is the RHS in Eqs. (1)–(3) for the first test case (ω_i^{*} only). A comparison of the number of $M^{-1}F$ evaluations for the first sawtooth cycle for $t \leqslant 3 \times 10^4\ \tau_a$ and the ones observed

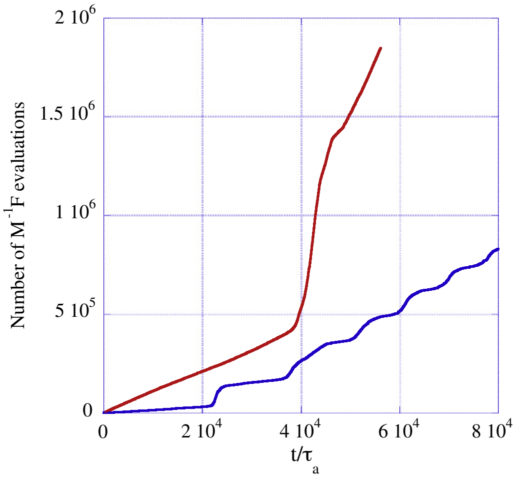

Fig. 9. Accumulated number of $M^{-1}F$ evaluations for the internal kink with the small discretization grid and the Newton–Krylov solver. (a) Resistive MHD with pressure evolution and ion diamagnetic rotation (blue curve). (b) Same case as (a) but supplemented with electron diamagnetic rotation (red curve). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

for the resistive kink evolution in Section 5 reveals no significant increase between the two cases. Note that the non-uniform growth of the blue curve in Fig. 9 is due to an increase of the time step between two sawtooth crashes, see rules in Section 4.1. The red curve in Fig. 9 shows the second test case with both ω_i^* and ω_e^*. In that case, the number of $M^{-1}F$ evaluations increases significantly. This issue reveals that even though electronic diamagnetic corrections are subdominant, they result in a sufficient modification of the operators to weaken the effect of the preconditioning. It is therefore necessary to extend the physical pre-conditioner used here to include bifluid terms. This will break the symplectic structure displayed in Eq. (17) and involves substantial changes in the linear algebra tools. This is currently under development, and numerical results on the subject will be reported after completion of this work.

Acknowledgements

The authors thank C. Nguyen, D. Leblond, F. Halpern, P. Maget and D. Meshcheriakov for carefully reading this manuscript. This work was granted access to the HPC resources of IDRIS under the allocation 2010-i2010050198 made by Grand Equipement National de Calcul Intensif (GENCI). It was carried out within the framework the European Fusion Development Agreement and the French Research Federation for Fusion Studies (FR-FCM). It is supported by the European Communities under the contract of Association between Euratom and CEA. The views and opinions expressed herein do not necessarily reflect those of the European Commission. Part of the graphics were realized with the SmileLab software, see http://www.satimage.fr/software/en/index.html.

References

[1] K. Lerbinger, J.F. Luciani, J. Comput. Phys. 97 (1991) 444.
[2] H. Lütjens, Comput. Phys. Commun. 164 (2004) 301.
[3] H. Lütjens, J.F. Luciani, J. Comput. Phys. 227 (2008) 6944.
[4] M. Pernice, H.F. Walker, Special issue on iterative methods, SIAM J. Sci. Comput. 19 (1998) 302. <http://users.wpi.edu/~walker/NITSOL/>. The NITSOL package can be downloaded at:.
[5] H. Lütjens, J.F. Luciani, Phys. Plasmas 4 (1997) 4192.
[6] H. Lütjens, J.F. Luciani, X. Garbet, Phys. Plasmas 8 (2001) 4267.
[7] H. Lütjens, J.F. Luciani, X. Garbet, Plasma Phys. Controlled Fusion 43 (Suppl. 12A) (2001) A339.
[8] H. Lütjens, J.F. Luciani, Phys. Plasmas 9 (2002) 4837.
[9] H. Lütjens, J.F. Luciani, Phys. Plasmas 12 (2005). Article No. 080703.
[10] H. Lütjens, J.F. Luciani, Phys. Plasmas 13 (2006). Article No. 112501.
[11] P. Maget, H. Lütjens, G.T.A. Huysmans, P. Moreau, B. Schunke, J.L. Segui, X. Garbet, E. Joffrin, J.F. Luciani, Nucl. Fusion 47 (2007) 233.
[12] P. Maget, G.T.A. Huysmans, X. Garbet, M. Ottaviani, H. Lütjens, J.F. Luciani, Phys. Plasmas 14 (2007). Article No. 052509.
[13] P. Maget, G.T.A. Huysmans, H. Lütjens, M. Ottaviani, Ph. Moreau, J.L. Segui, Plasma Phys. Controlled Fusion 51 (2009). Article No. 065005.
[14] L. Chacón, D.A. Knoll, J.M. Finn, J. Comput. Phys. 178 (2002) 15.
[15] L. Chacón, D.A. Knoll, J. Comput. Phys. 188 (2003) 573.
[16] D.A. Knoll, V.A. Mousseau, L. Chacón, J. Reisner, J. Sci. Comput. 25 (2005) 213.
[17] L. Chacón, Comput. Phys. Commun. 163 (2004) 143.
[18] L. Chacón, Phys. Plasmas 15 (2008). Article No. 056103.
[19] D.A. Knoll, D.E. Keyes, J. Comput. Phys. 193 (2004) 357.
[20] S.I. Braginskii, in: M.A. Leontovich (Ed.), Reviews of Plasma Physics, vol. 1, Consultants Bureau, New York, 1965, p. 205.
[21] S.-T. Tsai, F.W. Perkins, T.H. Stix, Phys. Fluids 13 (1970) 2108.
[22] L.E. Sugiyama, W. Park, Phys. Plasmas 7 (2000) 4644.
[23] D.D. Schnack, D.C. Barnes, D.P. Brennan, C.C. Hegna, E. Held, C.C. Kim, S.E. Kruger, A.Y. Pankin, C.R. Sovinec, Phys. Plasmas 13 (2006). Article No. 058103.
[24] O. Sauter, R.J. La Haye, Z. Chang, et al, Phys. Plasmas 4 (1997) 1654.
[25] H. Lütjens, A. Bondeson, A. Roy, Comput. Phys. Commun. 69 (1992) 287.
[26] H. Lütjens, A. Bondeson, O. Sauter, Comput. Phys. Commun. 97 (1996) 219.
[27] H. Lütjens, A. Bondeson, G. Vlad, Nucl. Fusion 32 (1992) 1625.
[28] A. Bondeson, G. Vlad, H. Lütjens, Phys. Fluids B4 (1992) 1889.
[29] M.N. Bussac, R. Pellat, D. Edery, J.L. Soulé, Phys. Rev. Lett. 35 (1975) 1638.
[30] P. Maget, H. Lütjens, R. Coelho, B. Alper, M. Brix, P. Buratti, R.J. Buttery, H. De la Luna, N. Hawkes, I. Jenkins, C. Challis, C. Giroud, X. Litaudon, J. Mailloux, M. Ottaviani, EFDA Contributors, Nucl. Fusion 50 (2010). Article No. 045004.