Technical report, Project ASTER
(ANR-CIS-2006-001-02)

Implementation of an iterative solver in the non-linear MHD code JOREK

G.T.A. Huysmans

Association Euratom-CEA, CEA/DSM/DRFC, Centre de Cadarache,
13108 St. Paul lez Durance, France

This work, as part of the project ANR-CIS.2006.001, has benefited from financial support from the French ‘Agence National de la Recherche’

Ce travail a bénéficié d'une aide de l'Agence Nationale de la Recherche portant (référence ANR-06-CIS6-001.

Introduction

The non-linear MHD code JOREK has previously been applied to the simulation of Edge Localised Modes (ELMs) in tokamaks [1] in the magnetic geometry with an x-point. These initial simulations of the non-linear evolution of medium-n ballooning modes successfully described the non-linear phase and the decay of the instability. However, the simulations were limited to the equilibrium n=0 toroidal harmonic and one additional (n=6) harmonic.

The JOREK code uses a fully implicit scheme for the time evolution of the MHD equations. This requires a solution of one large matrix containing all the degrees of freedom. This system of equations was solved using a direct method, using one of the libraries Pastix, WSMP, or MUMPS. The limitation in the toroidal resolution is mostly due to the memory requirements of these direct methods.

In order to reduce the memory requirements and possibly improve the scalability of the parallisation, iterative solvers are being considered. The implementation and performance of a suitably preconditioned GMRES solver in JOREK is described below. The new solver is used in the simulations of ballooning modes with an increased toroidal resolution.

The hybrid version of the Pastix library using a combination of a direct and an indirect method could also be a viable option. This option will be investigated and compared to the currently solver in the near future.

Iterative solver, preconditioner

In the present version JOREK code, the variables, $\vec{U}$, are represented using finite elements in the poloidal plane and Fourier harmonics in the direction of symmetry, the toroidal direction:

$$\vec{U}(s,t,\phi) = \mathrm{Re}(\sum_{n=0}^{Ntor-1} \sum_{i=1}^{Nnodes} \sum_{j=1}^{4} u_{ijn} H_j\left(s,t\right) e^{in\phi})$$

where $\left(s,t\right)$ are the local variables in each finite element, φ is the toroidal angle, u_{ijn} are the expansion coefficients. The functions $H_j\left(s,t\right)$ describe the newly implemented cubic Bezier finite elements [2]. The Bezier elements are a generalisation of the standard cubic Hermite finite elements which have a continuous function value and a continuous first derivative. Each variable has 4 degrees of freedom per node. The total number of degrees of freedom is therefore $4N_{tor}N_{nodes}N_{\mathrm{var}}$. A typical number (for 16 harmonics on a grid of 101 by 128 finite elements) is of the order of 5 million.

The JOREK code uses a fully implicit time evolution scheme without any splitting of the equations. At every time step the equations are linearised around the current state (a linearised Crank Nicholson scheme):

$$\frac{\partial A(\vec{u})}{\partial t} = B(\vec{u}) \;\Rightarrow\; \left(\frac{\partial A(\vec{u}_n)}{\partial u} - \tfrac{1}{2}\delta t \frac{\partial B(\vec{u}_n)}{\partial u}\right)\delta\vec{u} = B(\vec{u}_n)\,\delta t$$

Where the operators A and B describe the MHD equations, $\vec{u}_n$ the variables at time step n and $\delta\vec{u} = \vec{u}_{n+1} - \vec{u}_n$.

The resulting complex sparse matrix of order 5×10^6 has to be solved at every time step.

The sparsity structure of the matrix is determined by the unstructured 2D finite element grid but with a relatively large block size of $4N_{tor}N_{\mathrm{var}}$ due to the number of unknowns at each node of the grid.

The effectiveness of an iterative solver is determined mostly by the quality of the preconditioner. Our choice of the preconditioner is motivated by physics arguments specific for our application, i.e. MHD instabilities in tokamak plasmas.

The MHD instabilities are usually a relatively small magnetic perturbation to a large 'equilibrium' magnetic field (the n=0 component). The velocity perturbations can be larger than the equilibrium flow but the amplitude is of the same order as the magnetic perturbation.

In the toroidally symmetric tokamak, the toroidal mode number (i.e. of the Fourier expansion) is a 'quantum' number. This means that for small perturbations, the evolution of each toroidal harmonic is independent from the evolution of the other harmonics.

When the matrix is reordered such that all the contributions from one harmonic appear in the same (now sparse) block, the matrix becomes diagonally dominant. The interaction between the harmonics is small, at least when the instabilities are small. The amplitude of the off-diagonal terms will become larger in the strongly non-linear phase. In other words, for small amplitudes the evolution of the system is determined by the interaction of each toroidal mode with the n=0 equilibrium mode.

Thus, it appears logical to choose the sub-matrices of each toroidal harmonic as the preconditioning matrix. This gives N_{tor} independent matrices of manageable size. This approach can be seen as a form of a block-Jacobi preconditioner but with the blocks from the matrix reordered by toroidal mode number.

Implementation

The GMRES solver made available by CERFACS [3] has been implemented in the JOREK code. The GMRES solver requires an external routine for the matrix vector multiplication of the preconditioning matrix and an external routine for the matrix vector multiplication of the full matrix. In JOREK, the left side preconditioning is used.

The resulting sub-matrices are factorised and solved using a direct solver (Pastix [4]). Each matrix is assigned to an MPI group of processors; the total number of processors is divided into N_{tor} groups. This choice of preconditioner leads to a very good scalability but does require a global communication to redistribute the matrix from the original element based distribution to a toroidal mode number based distribution. Of course, only the block diagonals are redistributed. The size of the matrix for each harmonic is $4N_{\mathrm{var}}N_{nodes}$, with the same sparsity structure due to the finite elements but with a smaller blocksize $(4N_{\mathrm{var}})$.

The most time consuming part is the factorization of the sub-matrices. Therefore the preconditioning is only recalculated (and factorised) when the number of GMRES iterations (of the previous time step) exceeds a certain number. This number (typically 20-50 iterations) is chosen such that the time spend on GMRES iterations remains below or of the same order as the time for the factorization.

Figure 1 shows an example of the number of iterations as a function of time for the case of an internal kink instability in a circular plasma. In the first 100 time steps the amplitude of the instability is still small and only a small number of GMRES iterations is required. The non-linear phase of the internal kink instability is characterized by a relatively large perturbation. This leads to an increase in the number of iterations. In this phase, the preconditioning matrix is updated and factorised at every time step. The figure shows the number of GMRES iterations for two values of the time step ($\delta t = 20$ and $\delta t = 10$ Alfven times). It is clear that the largest value of the time step does not necessarily lead to the shortest computing time. In this case the reduction in the number of iterations more than compensates for the reduction in the time step.

The number of GMRES iterations can be used to adapt the time step dynamically during the simulation. An example is shown in Figure 2. In this case, when the number of iterations exceeds 200, the time step is reduced by a factor of two. This scheme allows one to take very large time steps in the linear phase (up to 1000 Alfven times) and automatically adapt the time step in the non-linear phase.

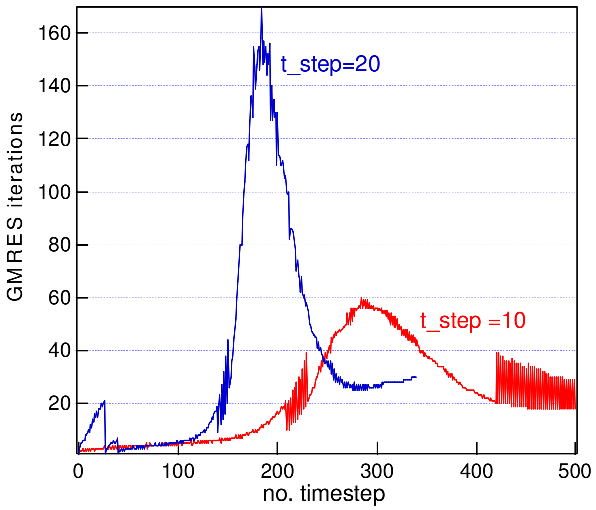


***Figure 1** The number of GMRES iterations during a simulation of an internal kink crash for two values of the time step. (resistivity and viscosity are 10^{-5}, the GMRES residue 10^{-8})*

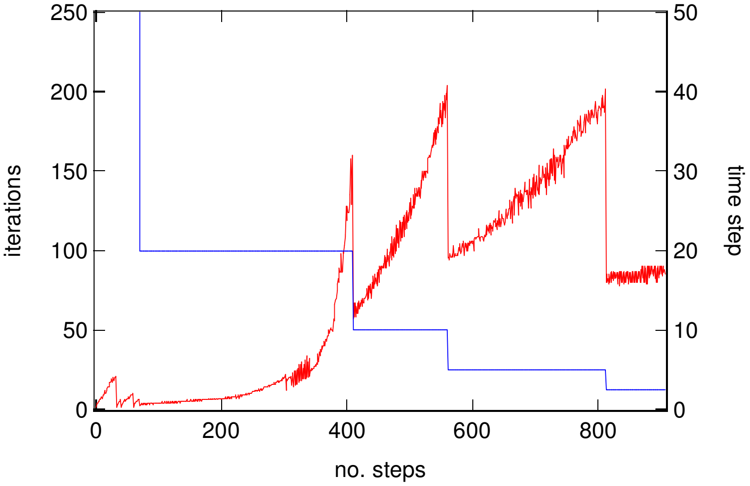


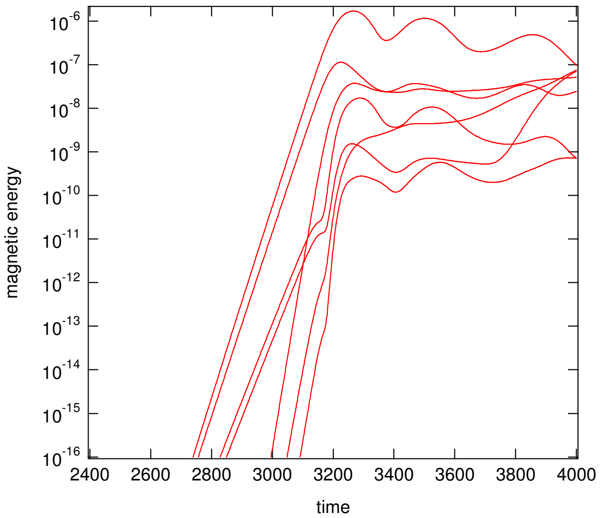
***Figure 2** The number of GMRES iterations (red) and the dynamically adapted time step (blue) as a function of time for a simulation of an internal kink mode. (resistivity and viscosity are 10^{-6}, GMRES residue 10^{-8})*

Application

For the ASTER project the main application is the simulation of ELMs in tokamak plasmas. It is therefore necessary to verify that the new iterative solution scheme also works in the case of ballooning modes. The JOREK code has been applied to evolve an unstable equilibrium in the relevant magnetic geometry with an x-point and with a large pressure gradient just inside the separatrix. The toroidal mode numbers used are n=0, 3, 6, 9, 12, 15, 18 and 21, i.e. 8 Fourier harmonics. The high toroidal mode numbers require a fine resolution in the poloidal plane. Here an (unstructured) grid of 56 radial and 96 poloidal cubic Bezier elements has been used.

The new solver successfully completed the simulation using a relatively low number of iterations. Figure 3 shows the evolution of the magnetic energy of all the Fourier harmonics (except n=0). The dominant harmonic is an n=12 mode. Figure 4 shows the density (in color) in the poloidal plane at t=3360 Alfven times. Also shown are the contours of the flux surfaces. The expulsion of density filaments into the low density region outside the main plasma is consistent with the previous simulations using a single harmonic.

Figure 5 shows the number of iterations as a function of the time step. The pre-conditioning matrix is recalculated when the number of iterations exceeds 50. This occurs only in a short phase of the simulation at the time of maximum amplitude of the perturbation.



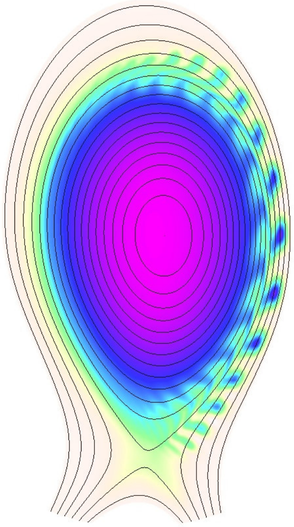


Figure 3 *The time evolution of the magnetic energy of the toroidal harmonics as a function of time (in units of Alfven time). The dominant harmonic is an n=12 mode. (central resistivity* 10^{-6}*, viscosity* $5x10^{-6}$)

Figure 4 *(right) The perturbed density profile in the poloidal plane at t=3360 Alfven times, showing the expulsion of density filaments into the vacuum (a low density region). Also shown are the flux surfaces with the x-point.*

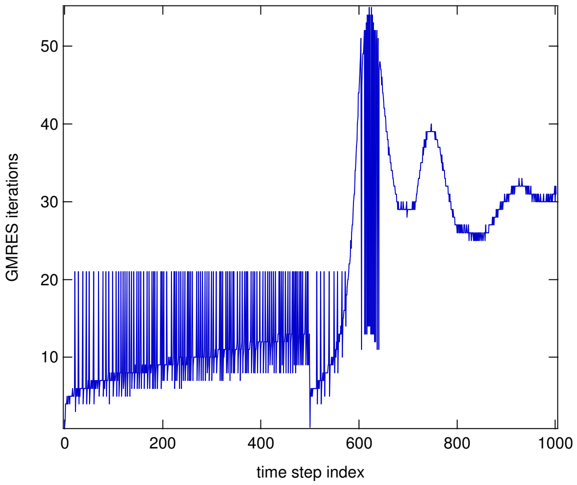


Figure 5 *The number of GMRES iterations for a ballooning mode simulation using 8 toroidal harmonics. The preconditioner is recalculated when the number of iterations exceeds 50.*

Conclusion

The GMRES solver implemented in JOREK using the sub-matrices of each toroidal harmonic as preconditioner appears to work very well. The iterative solver changes the scaling of the cpu time and the memory requirements from a quadratic scaling in the number of toroidal harmonics to a linear scaling. This is true as long as the factorisation of the preconditioner is the dominant factor. At higher number toroidal mode numbers, the GMRES iterations will become more costly due to the quadratic increase of the matrix size. However, the parallelisation of the GMRES matrix vector product scales well with the number of processors.

Further work is required to establish the limits of the iterative method. With decreasing dissipative effects (resistivity viscosity and heat/particle diffusion) the condition number of the system of equations increases and the iterative methods become more difficult.

Next, the performance of the current iterative method using an ‘physics’ based preconditioner will be compared with the hybrid version of the Pastix [4] library which uses a mathematics based preconditioner (incomplete factorisations).

References

[1] G.T.A. Huysmans and O. Czarny, *Nucl. Fusion* **47** No 7 (July 2007) 659-666
[2] O. Czarny and G.T.A. Huysmans, accepted for publication in *Journal of Comp. Physics* (2008)
[3] V. Frayssé, L. Giraud, S. Gratton, and J. Langou, A set of GMRES routines for real and complex arithmetics on high performance computers, CERFACS Technical Report TR/PA/03/3, public domain software available on www.cerfacs/algor/Softs, 2003
[4] P. Hénon, P. Ramet and J. Roman, Parallel Computing, 2001, Vol. 28, No.2, p.301-321