Spectral Element Implementation of the M3D Extended MHD Code

H.R. Strauss
B. Hientzsch
New York University,
251 Mercer Street, New York, New York 10012
J. Chen
Princeton Plasma Physics Laboratory, Princeton, NJ 08570

Abstract

A spectral element library has been developed and applied to the M3D extended MHD code. Spectral element methods offer several possible advantages for MHD simulations. They are high order discretizations and offer the possibility of exponential decrease of the error with increasing degree. Since spectral element methods can be implemented with discrete operators that are combinations of tensor product matrices and point-wise operations, they can be implemented efficiently at close-to-peak on modern computer architectures. The resulting global stiffness and system matrices are sparse block matrices in which the blocks are dense and of a special structure. Direct solvers can use static condensation and sparse solvers for the much smaller Schur complement system which leads to a fast and efficient solution algorithm.

M3D [1] is a highly modular code for extended MHD problems. Its modularity allows the implementation of several discretizations and the change from one to another for the same problem. Originally, M3D used a spectral discretization in the toroidal and poloidal angles combined with finite differences in the radial direction. While leading to fast solvers and accurate solutions, that discretization did not allow for complicated geometries of the cross sections such as needed to model stellarators and divertor tokamaks. To handle such geometries, a discretization with linear finite elements [2] was introduced, which is still the standard version at present. The current implementation of a spectral element discretization, as described in this presentation, should allow for complex geometries while at the same time recovering the high accuracy that the original version achieved on simple geometries. A comparison of results from the linear finite element and the spectral element discretizations will be presented.

Spectral element (SEL) methods [3, 4] have been recently intoduced in MHD simulations [5, 6]. The SEL approach offers several possible benefits. The discretization can be made accurate to high order, with exponential decrease of error as the order is increased. Static condensation provides an efficient solution method for elliptic problems, in which the Schur complement matrix to be solved is orders of magnitude smaller than the original matrix. Curved isoparametric elements allow alignment with realtively complicated boundary shapes encountered in simulation of magnetic fusion experiments. Spectral elements give a diagonal mass matrix, which is big advantage for the partially implicit M3D time stepping scheme.

Against these benefits, there is a concern that high order methods are only good for smooth problems and will not work for highly nonlinear turbulent MHD flows, such as can occur in magnetic fusion disruptions and Edge Localized Modes (ELMs).

M3D has been parallelized using Openmp for shared memory computers, and MPI / Petsci for distributed memory computers. The Openmp version is more restricted in problem size, mainly because no domain decomposition in poloidal planes is employed. Each poloidal plane is assigned to a separate processor. The MPI / Petsci distributed memory version is being developed. For now, the standard Petsci solvers are used, without taking advantage of static decomposition. This will be addressed in the future.

Spectral Elements

The SEL elements have C^0 continuity, which is the standard approach. There exist C^1 SEL methods, but they are more complicated to implement and have certain restrictions to their applicability.

The SEL method consists of a spectral discretization inside of discrete spatial regions. The spectral representation allows for an efficient high order discretization. The use of finite elements allows the mesh to be aligned with more or less arbitrary boundaries.

The solution is expanded in terms of Lagrange basis functions, and evaluated by collocation on the Gauss Lobatto Legendre (GLL) nodal points. The discretization is a tensor product of one dimensional representations. This makes the method highly efficient. Mappings and curved elements bring in geometrical factors inside of diagnonal matrices.

Here a nodal representation is used. Each basis function is zero except a single point. Hence the product of two basis functions, evaluated by collocation, is zero if the basis functions are different. This gives a diagonal mass matrix which is trivial to invert, and allows a large speedup in a partially explicit method such as M3D. The nodal representation can easily be transformed into a modal representation, which might lend itself to easier spatial filtering, or truncation error analysis. In this implementation, no special filtering is used. Instead we rely on adequate physically based dissipation by viscosity, thermal condunctivity, and resistivity to obtain a suffiently smooth solution.

The curved elements are applied by an isoparametric representation of the element edges. These are blended by a linear weighting inside the elements.

Coupling of Spectral Elements to M3D

M3D is a highly modular code, which makes it possible to change the underlying discretization of the solution. The right hand side of the equations is built up by calling a set of subroutines which implement various operators, such as derivatives in the three coordinate directions, Poisson brackets, inner product of derivatives, and Laplacian. The left hand side of the equations involves solving a set of two dimensional elliptic operators, such as Poisson and Helmholtz equations. The high level driver part of the M3D code, which is for the most part Fortran legacy code, does not explicitly contain mesh information. It simply calls functions from the SEL library of solvers and differential functions. The physical variables, as well as many auxillary variables, are stored in common blocks in the driver code. It is assumed that the mesh is unstructured, and the variables are simply listed in two index arrays. The first index refers to the location in the poloidal plane, and the second refers to toroidal angle. When an operator is called, the variables in its arguments are passed to functions in the SEL library, which is written in C. The variables must be mapped to temporary C arrays which have a different layout than the “flat” Fortan arrays. The C

arrays are organized by elements, and include storage for the nodal values interior to each element, and the boundary values of each element. This arrangement is convenient for the solvers, which use static condensation. Although it seems redundant to move data between different arrays which represent the same data in different formats, the movement of data consumes a generally negligible amount of computer time. The same strategy was used the couple M3D to linear and low order finite elements, while leaving the Fortran code essentially intact.

Mesh generation is initiated in the driver code. A skeleton mesh is generated, which consists of the corner vertices of quadrilateral elements. In addition, the GLL points lying on the curved element boundaries are generated. This information is passed to the SEL library, which generates the full mesh, which consists of the GLL points on lines connecting the boundary points. The interior points lie on curves which are linearly blended from the element boundary points. It would be desirable, instead, to have the interior element points generated by calling back the driver program. This would ensure that the points were aligned with flux surfaces. On the other hand, higher order discretations should be less sensistive to the mesh than low order.

Equilibrium and Linear Stability Comparison of Spectral Element and FEM Implementations

Some preliminary testing is presented. Some initial tests were done of a tokamak equilibrium. The equilibria were produced by choosing the same nonequilibrium initial state for both versions and relaxing to a two dimensional equilibrium. A simple case was chose which was used to benchmark the FEM version to the initial spectral M3D. The equilibria are in excellent agreement for sufficiently high resolution. The SEL run used the skeleton mesh shown in Fig.1, with 8th order polynomials. Fig.2,Fig.3 show the equilibrium pressure in (a) and the toroidal current density, which somewhat more sensitive to discretization, in (b).

These equilibria were perturbed with toroidal mode number $n = 3$ perturbations and were advanced in time until the solution was dominated by an unstable eigenfunctions. The eigenfunctions show some small diffences in detail. This can be seen in Fig.4, where (a) is the FEM and (b) the SEL solution. The convergence of growth rates with number of meshpoints in shown in Fig.5. The solid curve is growth rate as a function of meshpoints for the FEM, and the dashed curve is the growth rate as a function of meshpoints for the SEL. The SEL calculations all used the skeleton mesh of Fig.1, but the polynomial order was varied from 2 to 8. Except for order 2, the growth rate seems to not vary very much with order. This is an example of the improved convergence expected of SEL methods.

Nonlinear Simulations

In general, the low order elements are more robust in nonlinear simulations, because they seem to require less smoothness. An important lesson seems to be that adequate smoothing must be included, by means of viscous dissipation coefficents such as viscosity and resistivity, to keep the solution sufficiently smooth for spectral elements. The results indicate that high order elements can be robust. Looking at Fig.6(a), and Fig.7(a) the

pressure in the FEM seems to have extra smoothing. This is also evident in Fig.6(b), Fig.7(b), the toroidal current density.

References

[1] W. Park, E.V. Belova, G.Y. Fu, X. Tang, H.R. Strauss, L.E. Sugiyama, Plasma Simulation Studies using Multilevel Physics Models Phys. Plasmas **6** 1796 (1999).

[2] H. R. Strauss and W. Longcope, An Adaptive Finite Element Method for Magnetohydrodynamics, J. Comput. Phys. 147, 318 - 336 (1998).

[3] A. T. Patera, A spectral element method for fluid dynamics - laminar flow in a channel expansion, J. Comp. Phys.**54**, 463 (1984).

[4] P. F. Fischer and A. T. Patera, Parallel spectral element solution of the Stokes problem, Journal of Computational Physics, 1991, 92(2), p380-421.

[5] A. H. Glasser and X. Z. Tang, The SEL macroscopic modeling code, Computer Physics Communications **164**,237 (2004)

[6] C. R. Sovinec, A. H. Glasser, T. A. Gianakon, D. C. Barnes, R. A. Nebel, S. E. Kruger, D. D. Schnack, S. J. Plimpton, A. Tarditi, and M. S. Chu, Nonlinear magnetohydrodynamics simulataion using high - order finite elements, J. Comp. Phys. **195**, 355 (2004).

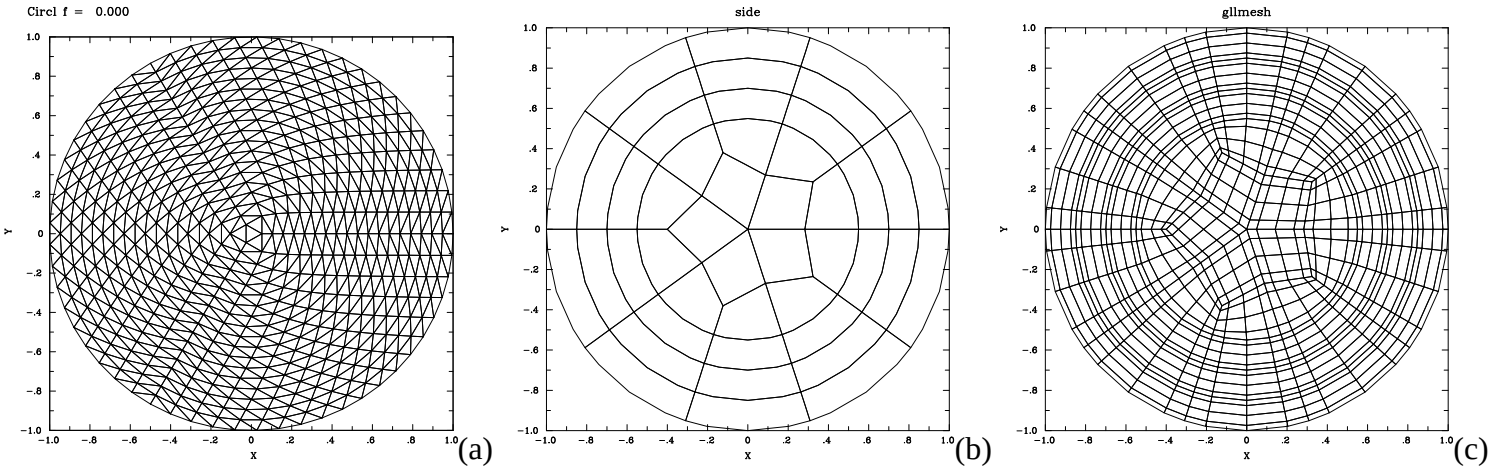

Figure 1: (a) Linear finite element mesh (b) skeleton mesh for spectral elements (c) spectral element mesh

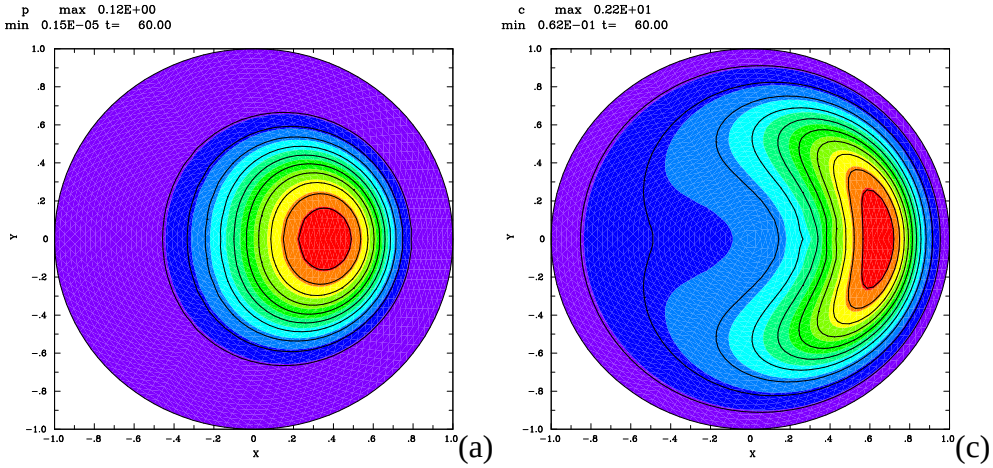

Figure 2: (a) Equilibrium (a) pressure (b) toroidal current density calculated with linear finite elements

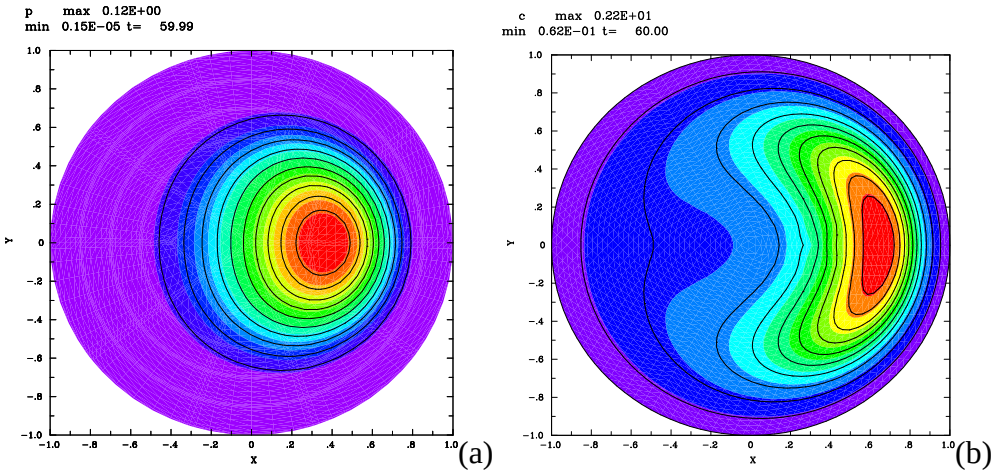

Figure 3: Equilibrium (a) pressure (b) toroidal current density calculated with 8th order spectral elements

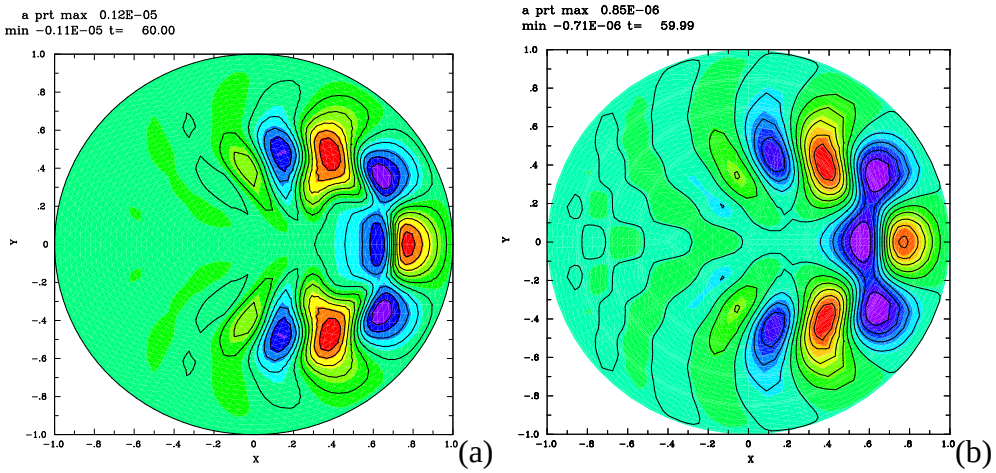

Figure 4: (a) perturbed magnetic flux function ψ (a) calculated with linear finite elements (b) calculated with 8th order spectral elements

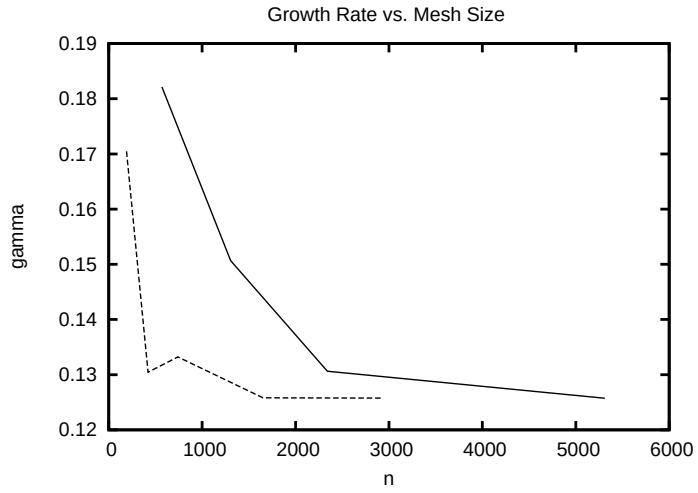

Figure 5: Growth rate as a function of number of mesh points. Solid curve: Finite elements; Dotted curve, spectral elements. Note the expected improved convergence of the growth rate with spectral elements. The number of mesh points in the SEL mesh was changed by varying the polynomial order from 2 to 8.

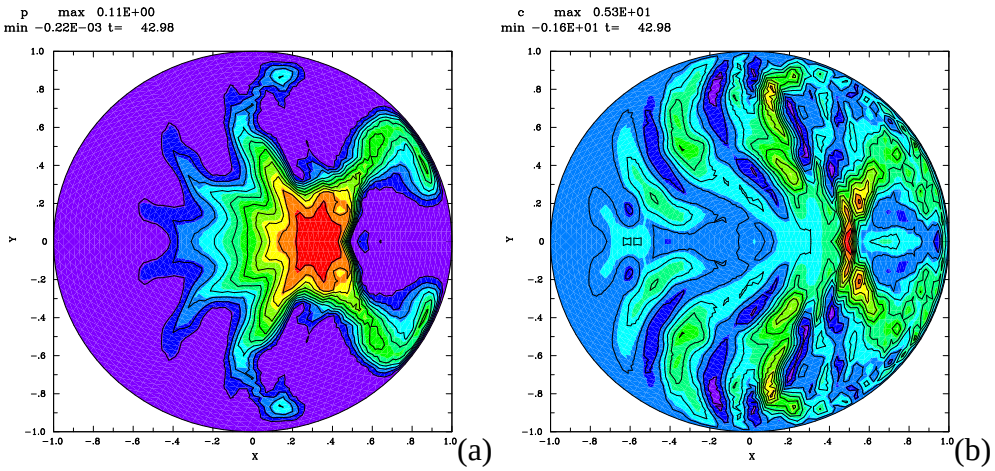

Figure 6: (a) nonlinear pressure (b) nonlinear current density calculated with linear finite elements

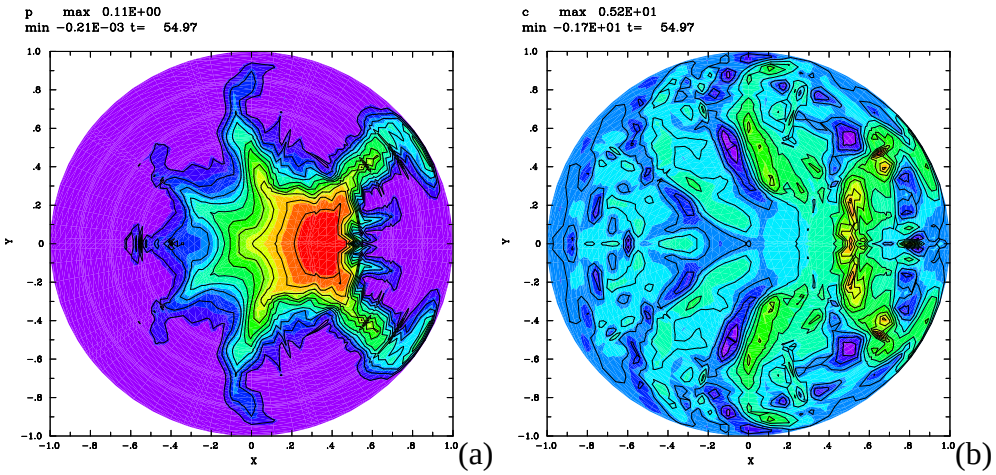

Figure 7: (a) nonlinear pressure (b) nonlinear current density calculated with 8th order spectral elements