THE NIMROD CODE DEVELOPMENT PROJECT

Executive Summary

The NIMROD project was commissioned by the Office of Fusion Energy in February, 1996, to address several perceived needs of the large scale computations that are crucial to the Magnetic Fusion Program. These needs arose as a result of four factors:

1. The ascendancy of massively parallel computer architectures over the serial vector architectures for which most large fusion codes have been optimized, and the high costs and inefficiencies associated with the conversion of existing codes to the new architectures;
2. The general uncertainty in the configuration of future generations of supercomputers, and the anticipated costs of yet another massive code conversion;
3. The high cost and long time line associated with individual code development, and the general inaccessibility of the resulting codes to the general fusion community; and,
4. The increasing programmatic importance of physics problems that require extensions of the capabilities of the existing suite of computational tools.

The identified needs can be summarized as:

1. The need to lower the future cost of application development, maintenance, and use;
2. The need to implement computational and programming methods that isolate architecture dependencies so that future codes will be more efficiently adapted to evolving new architectures;
3. The need to make applications codes more readily available to, and usable by, the overall fusion science community.
4. The need for advanced simulation codes to address physics problems of immediate programmatic interest.

NIMROD is an acronym for Non-Ideal Magnetohydrodynamics with Rotation, Open Discussion. It refers to the technical goals of the project, and to the process around which the project is structured. The themes of the NIMROD project are:

- The rapid development of a state-of-the-art, user friendly code needed to address issues that are central and key to the rapid advance of fusion physics
- Designing and developing a modern fusion physics code that will be easy to maintain in the face of uncertainty in future computer architectures
- Testing new paradigms for code development by geographically distributed and technically diverse teams

Over the past seventeen months the NIMROD project has received $145,000 of new funding from OFES. About half of that has gone for computer software and hardware, travel, and to support our colleagues in Russia. Approximately 4.8 full-time

equivalents (FTEs) have been spent on the NIMROD project over the past seventeen months. Most of this labor effort has been redirected from other projects.

Background

Modern tokamak plasmas are hot enough to be virtually collisionless. Analysis of the nonlinear dynamics of these plasmas for long time scales, for example the evolution of neoclassical magnetic islands and their effect of β-limits, therefore requires extension of the usual resistive MHD model to include additional "non-ideal" effects such as the Hall term, anisotropic thermal transport, different ion and electron temperatures, finite electron inertia, neo-classical contributions to the stress tensor, pressure contributions from energetic species, etc. The details of the non-circular geometry of these devices are also essential for accurate modeling.

One way to formulate such a model is to add additional terms to the usual resistive MHD equations. Another approach is to directly solve the nonlinear, two-fluid dynamical equations simultaneously with Maxwell's equations. In this case the only assumption is that the ions and electrons have near Maxwellian distribution functions and can thus be described as interacting fluids. This approach misses purely kinetic effects such as those related to Landau damping and finite Larmor radius, but includes all the cold plasma wave branches such as cyclotron, hybrid, and whistler waves, as well as low frequency Alfvén and sound waves[1]. Collisionless, as well as resistive, reconnection is also described by this model. The NIMROD code is being developed to solve the equations of such a model in three-dimensional toroidal geometry and time.

Quality Function Deployment

The NIMROD project is structured around a process called Integrated Product Development (IPD), which is an aspect of Concurrent Engineering[2]. One aspect of IPD is Quality Function Deployment (QFD), which is a methodology for reaching and documenting sound decisions. It is a well defined method for developing a quality product aimed at satisfying the customer by translating the customer's demands into design targets and major quality assurance points to be used throughout the development phase, to assure that quality is built into the product while it is still in the design stage. A goal of the QFD methodology is to avoid costly re-engineering after product deployment. Re-engineering often involves more cost than the initial design. An IPD project is based on a team that is multi-discplinary and self-directed. The customer is a crucial member of the team, and participates throughout the design and development process. IPD and QFD are widely used in manufacturing industries. At its heart, QFD is a well defined algorithm for making common sense decisions.

One goal of the NIMROD project is to test if the IPD/QFD process can be useful in addressing the programmatic needs enumerated above. In this case, the customer is the potential code user, a fusion experimentalist or theorist, and the product is a computer code that can be easily accessed and used by the fusion community to solve problems of programmatic interest. Re-engineering, which the QFD methodology aims to avoid, here refers to the need to make massive and costly code modifications in response to evolving computer architectures or programmatic needs. Building a product that successfully avoids re-engineering may require a somewhat longer development time, but will pay dividends in the future.

At this time, the NIMROD team consists of physicists and computer scientists from ten institutions in two countries. National labs, universities, and industry are represented. Potential users (customers or their representatives) have been involved in the code design process from the beginning. The team holds bi-monthly face-to-face meetings, and meets more frequently with telephone and video conferencing, and daily with e-mail. A dedicated NIMROD web page is also used to record and exchange information. Decisions are made by team consensus. When difficult choices must be made, the formal QFD methodology has proven useful in making the proper decisions. At other times, the proper course seems clear and common sense prevails easily.

The NIMROD Code System

The NIMROD code system consists of a physics kernel (`nimrod`), a pre-processor (`nimset`), a grid generator (`fluxgrid`), and a post-processor. All components of the code system interact with the user, and with each other, through a graphical user interface (GUI). The NIMROD GUI can also access user defined equilibrium and stability codes that can be used to generate input for the NIMROD physics kernel. The interaction is through standard fusion file formats, such as `eqdisk`, which contains the information necessary to describe an axisymmetric poloidal equilibrium.

The role of the grid generator is to read a file containing the equilibrium fields (such as `eqdisk`) and generate a grid to be used by the physics kernel. NIMROD uses a grid in the poloidal plane, and an expansion in finite Fourier series in the periodic toroidal direction The poloidal grid is non-orthogonal, and is based flux coordinates inside the separatrix. These coordinates conform to the axisymmetric ($n = 0$) component of the poloidal magnetic field. (Occasional regridding may be required as the axisymmetric component evolves.) This choice of grid naturally resolves the extreme localization of dynamical behavior near mode rational surfaces that is characteristic of low frequency phenomena in magnetized toroidal plasmas. Since one coordinate direction is aligned (or nearly aligned) with the local magnetic field direction inside the separatrix, it also simplifies the difficult problem computing of highly anisotropic transport. The NIMROD grid is decomposed as an unstructured collection of blocks. There are two kinds of blocks: RBLOCKs and TBLOCKs. RBLOCKs contain a structured grid of logically rectangular cells. TBLOCKs contain an unstructured grid of triangles. TBLOCKs are used to join non-conforming RBLOCKs, and also in the vicinity of the magnetic axis. The combination of RBLOCKs and TBLOCKs allows NIMROD to simultaneously and accurately represent both highly localized activity near rational surfaces and real machine geometry. In addition, flux conforming elements can be restricted to the core plasma region. The grid is easily extended beyond the separatrix into the vacuum region.

The pre-processor establishes the initial conditions on the NIMROD grid, and generates an input file for the NIMROD physics kernel. The initial conditions generally consist of an equilibrium distribution of magnetic field, pressure, and density (obtained from the `eqdisk` or other equilibrium file), along with small, perturbed values of the magnetic field and/or velocity. These can be generated as a random perturbation, or an eigenfunction can be read from files produced by a linear stability code. The input file contains run control parameters, and is generated from values supplied interactively by the user through the GUI or as modifications to a pre-existing input file. The initial conditions

and the grid are written to a NIMROD restart file, which contains the instantaneous state of the system, and is read by the NIMROD physics kernel.

The NIMROD physics kernel solves the nonlinear, time dependent, three dimensional fluid plasma equations in axisymmetric toroidal geometry. The physics kernel always starts from a NIMROD restart file, and periodically produces new restart files. These files can be used to restart NIMROD from an intermediate point in the calculation, and can be read by post-processing codes. For the NIMROD physics kernel, there is no distinction between starting and restarting; every case is a restart. The kernel also produces other intermediate files that can be used by graphical post-processors to produce static or animated visualization.

NIMROD primarily uses two graphical post-processors. XDRAW, which is used for "quick-look", or diagnostic graphics, was written by a NIMROD team member and has found independent use throughout the fusion community. The kernel also generates files that can be used by IBM DataExplorer (DX), a commercially available product that can produce publication quality graphics and state-of-the-art animation. Data compatible with other user defined or commercially available visualization products can also be produced.

The NIMROD GUI

Users interact with the NIMROD code system through a graphical user interface (GUI). Through it, the user can completely control the solution of a particular problem, beginning with equilibrium generation and linear stability analysis, through grid generation and input parameter specification, to code execution, run-time job monitoring, quick-look and presentation quality graphics, animation, and file storage and management. A help package is included to guide the user through the GUI functionality. This functionality was designed with input from customer representatives.

The NIMROD GUI is written in the scripting language tcl/tk. It can control and interact with jobs that are running on different computers. The ideal configuration is for the user to run the GUI, the pre-processor, and the post-processor/graphics package on a local workstation, and to run the physics kernel remotely, eg., on the T3E at NERSC.

The GUI is very general, and it is presently being used in other applications within the fusion program.

Programming Standards

NIMROD is designed to be modular, portable, parallelizable, and maintainable across evolving computer architectures and changing individual code developers and users. A set of programming standards has been developed with QFD methodology, and these standards have been rigorously applied by the development team. The components of the NIMROD code system are written using the object-based capabilities and data structures of Fortran90. Standard IEEE binary format is used for all data files. The code is well commented, and the coding style is uniform. Machine dependent libraries and functions are avoided. MPI is used for parallel computations.

Because of the use of these programming standards, NIMROD can be run on a large variety of computer platforms, including the CRAY T3E, CRAY T3D, IBM SP2, CRAY C90, workstation clusters, and serial Unix workstations.

The NIMROD Physics Kernel

The NIMROD physics kernel solves the coupled fluid equations for electrons and ions simultaneously with Maxwell's equations (ignoring the displacement current) in three space dimensions and time. These equations are solved as a single fluid system. The equations for this model are obtained by adding and subtracting appropriate multiples of the ion and electron equations of motion, and contain all the physics of the original two-fluid system[3]. The resulting single fluid equations are a momentum equation

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot \rho \mathbf{v}\mathbf{v} = -\nabla \cdot \mathbf{P}' + \frac{1}{c}\mathbf{J} \times \mathbf{B} \quad , \tag{1}$$

and a generalized Ohm's law

$$\mathbf{E} = -\frac{1}{c}\mathbf{v} \times \mathbf{B} + \frac{4\pi}{\omega_{pe}^2 (1+\nu)}\left[\frac{\partial \mathbf{J}}{\partial t} + \nabla \cdot (\mathbf{v}\mathbf{J} + \mathbf{J}\mathbf{v})\right] - \frac{1}{ne(1+\nu)}\nabla \cdot \left(\mathbf{P}'_e - \nu \mathbf{P}'_i\right) + \frac{1}{nec}\frac{1-\nu}{1+\nu}\mathbf{J} \times \mathbf{B} + \eta \mathbf{J} \tag{2}$$

where $\mathbf{v} = \left(\mathbf{v}_i + \nu \mathbf{v}_e\right)/(1+\nu)$ is the center of mass velocity, $\mathbf{J} = \mathbf{J}_e + \mathbf{J}_i$ is the total current density, $\nu = Zm_e / m_i$ is the mass ratio, Z is the ionization state, $\rho = mn$ is the mass density, $m = m_i(1+\nu)/Z$, $n = n_e = Zn_i$ is the number density, and $\mathbf{P}' = \mathbf{P}'_e + \mathbf{P}'$ is the total pressure tensor. The center of mass pressure tensors, denoted by the prime, are related to the individual species pressure tensors by

$$\mathbf{P}'_{e,i} = \mathbf{P}_{e,i} + n_{e,i} m_{e,i}\left(\mathbf{v}_{e,i} - \mathbf{v}\right)\left(\mathbf{v}_{e,i} - \mathbf{v}\right) \quad , \tag{3}$$

where $\mathbf{P}_{e,i} = p_{e,i}\mathbf{I} + \Pi_{e,i}$, $p_{e,i}$ is the species scalar pressure, and $\Pi_{e,i}$ is the species stress tensor.

The first term on the right hand side of Equation (2) represents ideal MHD, the second term represents electron inertia, the third term represents pressure effects, and fourth term is the Hall term, and the last term represents resistivity. In the usual resistive MHD model, only the first and last terms in Equation (2) are retained. Advection is not included in the present version of NIMROD. (Note that Equation (2) does not assume that the perpendicular electron pressure gradient is exactly balanced by the Lorentz force.)

The combined system of Equations (1) and (2), along with Maxwell's equations and a thermodynamic model, are very stiff. They contain responses ranging from electron cyclotron frequencies ($\Omega_{ce} \sim 1.7 \times 10^{11}\,\mathrm{sec}^{-1}$ in a proposed tokamak) to resistive diffusion ($\tau_R^{-1} \sim 1.4 \times 10^{-5}\,\mathrm{sec}^{-1}$). The low frequency modes of interest to the fusion program evolve on the order of seconds, and in a hot plasma the behavior of these modes can be affected by all the terms in Equation (2). Thus some sort of implicit method must be used to solve these equations.

We first note that all the terms in the generalized Ohm's law are symmetric except the Hall term, so that the complete system cannot be solved with standard iterative conjugate gradient (CG) methods. In NIMROD, the symmetric and anti-symmetric (Hall) plasma responses are treated separately by operator splitting, with CG iterations used to invert the symmetric part of the operator. Further, since NIMROD uses a finite Fourier series for the toroidal direction it is desirable to invert the system of equations mode by

mode. However, the nonlinear terms in Equation (2) cause all toroidal modes to be coupled, so that a fully implicit treatment is a formidable task. Instead, a semi-implicit[4-6] approach is used in the present version of NIMROD. A fully implicit treatment remains a goal, but requires advances in linear solver software.

The semi-implicit method is really a simple means of achieving the decomposition of a complex operator. One is often confronted with a situation, such as the present case with NIMROD, in which an implicit treatment of a complex problem is required but the resulting linear system is very difficult to solve. For example, consider the equation

$$\frac{\partial \mathbf{u}}{\partial t} = i\Omega \cdot \mathbf{u} \quad . \tag{4}$$

Now suppose that the operator $i\Omega$ can be decomposed as $i\Omega = i\omega_1 + i\omega_2$, where $i\omega_1$ contains the parts of $i\Omega$ that demand implicit treatment, and $i\omega_2$ contains everything else, and can be treated explicitly. For example, $i\omega_1$ may contain fast time scale physics and $i\omega_2$ may only describe slow motions. Or, $i\omega_1$ may contain only linear or axisymmetric terms. Then, we would like to solve the related problem

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = i\omega_1 \cdot \mathbf{u}^{n+1} + i\omega_2 \cdot \mathbf{u}^n \quad . \tag{5}$$

Solution of Equation (5) requires inversion of the operator $\mathbf{I} - i\omega_1 \Delta t$, which contains the essential implicit physics.

Often the explicit decomposition of $i\Omega$ into $i\omega_1$ and $i\omega_2$ is difficult or impossible to obtain, so that Equation (5) cannot be realized directly. The semi-implicit method provides an alternate decomposition of $i\Omega$. In the semi-implict method, the original problem defined by Equation (4) is modified as

$$\frac{\partial \mathbf{u}}{\partial t} = i\Omega \cdot \mathbf{u} + i\sigma \Delta t \cdot \frac{\partial \mathbf{u}}{\partial t} \tag{6}$$

where $i\sigma$ is the an arbitrary operator called the *semi-implicit operator*. Explicit time differencing is then applied to Equation (6). The result is

$$\begin{aligned} \frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} &= i\Omega \cdot \mathbf{u}^n + i\sigma \cdot \left(\mathbf{u}^{n+1} - \mathbf{u}^n \right) \\ &= i\sigma \cdot \mathbf{u}^{n+1} + i\omega_2 \cdot \mathbf{u}^n + \left(i\omega_1 - i\sigma \right) \cdot \mathbf{u}^n \quad . \end{aligned} \tag{7}$$

Solution of Equation (7) requires inversion of the operator $\mathbf{I} - i\sigma \Delta t$. Since $i\sigma$ is arbitrary, we can choose it so that it closely approximates $i\omega_1$ in some sense, but is easier to invert. Then the last term in Equation (7) becomes small and Equation (7) closely approximates Equation (5), but is easier to solve. A particularly useful situation occurs when both $i\Omega$ and $i\omega_1$ are precisely known, but $i\omega_2$ is tedious, impossible, or otherwise difficult to express explicitly. In that case, choosing $i\sigma = i\omega_1$ accomplishes the desired decomposition of $i\Omega$ in a simple and efficient way. It can be shown that proper choices of the semi-implicit operator lead to simple, unconditionally stable algorithms[6].

Without the semi-implicit term Equation (7) is simply an explicit equation, and is relatively easy to implement. The simple addition of the semi-implict term results in an unconditionally stable algorithm that is much simpler to solve than the original fully implicit

formulation. In fact, any explicit, and therefore time-step limited, algorithm can be efficiently stabilized by the addition of the proper semi-implicit operator. This is the power of the method.

Note that Equation (7) can be rewritten as

$$(\mathbf{I} - i\sigma\Delta t)\cdot\frac{\mathbf{u}^{n+1}-\mathbf{u}^{n}}{\Delta t} = i\Omega\cdot\mathbf{u}^{n} \quad . \tag{8}$$

Hence, the effect of the semi-implicit operator is to alter the time scale by a factor of $\|\mathbf{I} - i\sigma\Delta t\|^{-1}$. Since $i\sigma$ is generally a monotonic function of the wave number k, the modes with larger k (shorter wavelength) are most affected. The semi-implicit method alters the phase velocity of these modes (affects their dispersion) to the point where they satisfy the explicit Courant condition at the chosen (large) time step. Thus the semi-implicit method is sometimes described as introducing a k-dependent inertia. (Time centered implicit methods accomplish stability by the same means.)

To begin the time advance in NIMROD, the velocity at the advanced time step, $\mathbf{v}^{n+1}$, is obtained explicitly from Equation (1). Introducing time differencing into Equation (2) (without the anti-symmetric Hall term), the electric field can be divided into implicit and explicit parts, $\mathbf{E} = \mathbf{E}_{im} + \mathbf{E}_{ex}$. The ideal MHD contribution is written as $\mathbf{v}^{n+1}\times\mathbf{B}^{n}$, and is contained in the explicit electric field. Using Ampere's law without the displacement current, the implicit electric field can be written as $\mathbf{E}_{im} = \mathbf{Z}\cdot\nabla\times\Delta\mathbf{B}$, where the impedance tensor $\mathbf{Z}$ is given by

$$\mathbf{Z} = \frac{c}{4\pi}\left(\frac{4\pi}{\omega_p^2\Delta t} + \frac{1}{2}\eta\right)\mathbf{I} \quad , \tag{9}$$

and $\Delta\mathbf{B} = \mathbf{B}^{n+1} - \mathbf{B}^{n}$. Thus, electron inertia and resistivity are treated fully implicitly.

In the semi-implicit method, a new term is added to the original equation to obtain a modified equation that is unconditionally stable at arbitrary time step. The new term contains an arbitrary operator. If chosen properly, the solution of the modified equation closely approximates the solution of the original equation, but the new equation is easier solve[6]. Introducing this *ansatz* into Faraday's law, we have

$$\frac{1}{c}\frac{\Delta\mathbf{B}}{\Delta t} = -\nabla\times\left(\mathbf{E}_{im} + \mathbf{E}_{ex} + \mathbf{E}_{SI}\right) \quad , \tag{10}$$

where $\mathbf{E}_{SI}$ is the "semi-implicit electric field". Since $\mathbf{E}_{SI}$ is arbitrary, we choose it to be

$$\mathbf{E}_{SI} = \mathbf{Z}_{SI}\cdot\nabla\times\Delta\mathbf{B} \quad , \tag{11}$$

where

$$\mathbf{Z}_{SI} = \alpha\frac{c}{4\pi}\left(\frac{V_{A0}}{c}\right)^2\left(\mathbf{I} - \hat{\mathbf{b}}_0\hat{\mathbf{b}}_0\right) \tag{12}$$

and the subscript 0 refers to quantities based on the axisymmetric (n = 0) component. The resulting system to be solved it

$$\left[c\Delta t\nabla\times\left(\mathbf{Z} + \mathbf{Z}_{SI}\right)\cdot\nabla\times\mathbf{I} + \mathbf{I}\right]\cdot\Delta\mathbf{B} = -c\Delta t\nabla\times\mathbf{E}_{ex} \quad . \tag{13}$$

It can be shown that a proper choice of the coefficient α leads to an unconditionally stable system.

The operator on the left hand side of Equation (14) is axisymmetric and does not couple toroidal modes, so Equation (13) can be inverted mode by mode. Further, the choice of the form of the semi-implicit electric field, Equations (11) and (12), makes the solution of Equation (13) closely approximate the solution of a fully implicit system. For a linear system, the fully implicit and semi-implicit formulations are virtually equivalent. This has been confirmed by numerical experiments on NIMROD.

The anti-symmetric Hall term introduces whistler waves that propagate parallel to the magnetic field, so that an implicit treatment is also required for its advance. We again use a semi-implicit method to avoid inverting an anti-symmetric operator. Using operator splitting, we write

$$\frac{\mathbf{B}^{n+1}-\mathbf{B}^{*}}{c\Delta t} = -\nabla\times\left(\mathbf{E}_{Hall}+\mathbf{E}_{SI-Hall}\right) \quad , \tag{14}$$

where $\mathbf{B}^{*}$ is the solution of Equation (13), $\mathbf{E}_{Hall}$ is the Hall term on the right hand side of Equation (2), and

$$\mathbf{E}_{SI-Hall} = \alpha_H \nabla\times\Delta\mathbf{B} \quad . \tag{15}$$

A proper choice of the semi-implicit coefficient α_H leads to an unconditionally stable algorithm. The system defined by Equations (14) and (15) is symmetric and can be inverted with standard CG methods.

The pressure is also advanced by a semi-implicit method[4], so that the overall time advance is unconditionally stable. The present version of NIMROD contains a single adiabatic energy equation with the assumption $p_e = p_i = p/2$. Separate energy equations for electrons and ions will be added in a later version.

We again stress that our goal is to attain a fully implicit time advance. The semi-implicit method will only be used until suitable linear solver software becomes available that can solve for all toroidal modes simultaneously as an anti-symmetric system.

In a toroidal plasma, neoclassical transport occurs in the regime where the bounce frequency of trapped particles is greater than the effective collision frequency. The following expression for the divergence of the stress tensor are valid in this regime[7,8]:

$$\nabla\cdot\Pi_\alpha^{\text{neo}} = \mathbf{BB}\cdot\nabla\left(\frac{f_\alpha}{B^2}\right)+\frac{f_\alpha}{B^2}\nabla\left(\frac{B^2}{2}\right)-\frac{f_\alpha}{B^2}\mathbf{B}\times\nabla\times\mathbf{B}-\frac{1}{3}\nabla f_\alpha \tag{16}$$

where

$$f_\alpha = \left(p_\parallel - p_\perp\right)_\alpha = -m_\alpha n\mu_\alpha \frac{\left\langle B^2\right\rangle}{\left\langle\left(\hat{\mathbf{b}}\cdot\nabla B\right)^2\right\rangle}\mathbf{v}_\alpha\cdot\nabla\ln B \tag{17}$$

and μ_α is the viscous damping frequency for species α. During the nonlinear evolution of a plasma, or when flux surfaces do not exist, the flux surface averages in Equation (17) are taken over the surfaces defined by the axisymmetric (toroidal mode number $n = 0$) magnetic field. Note that only the velocity, and not its spatial derivatives, enters into

Equation (17). This formulation thus introduces phase shift and damping, but no diffusion. Explicit advection should suffice for numerical stability.

NIMROD uses finite elements for spatial discretization in the poloidal plane and finite Fourier series (Fourier collocation) in the periodic toroidal direction. The metric elements that describe the non-orthogonal grid are fitted with bicubic splines in order to capture the actual curvature of the grid lines. The dependent variables are expanded in bilinear finite elements. Finite elements were chosen for the spatial representation because they offer a consistent and well defined discretization of the equations on a non-orthogonal grid. Further, because the discrete equations are derived from a variational principle the resulting discrete operators are guaranteed to be symmetric positive definite. (The exception, of course, is the Hall term which is inherently antisymmetic.)

Linear Solver

The present version of NIMROD (Version 2.0) uses a standard iterative preconditioned conjugate gradient (CG) solver that is limited to symmetric positive definite systems. Two types of preconditioner are used: diagonal, and block direct. The diagonal preconditioner is standard. In the block direct preconditioner, a direct solve is done independently within each of the blocks of the NIMROD grid (RBLOCKs or TBLOCKs). Message passing (MPI) has been implemented in both preconditioners, and in the CG solver. Use of the block direct preconditioner results in fewer CG iterations than is required when the diagonal preconditioner is used, but at the present stage of optimization the preconditioner takes longer to execute. Obtaining better preconditioners is clearly a key to optimizing the performance of NIMROD.

The overall execution time of the code is primarily determined by the performance of the linear solver. We expect this issue will attract more of our attention in the next year.

Parallel Performance

One of the critical needs for fusion computing is to take maximum advantage of the new generation of massively parallel computer architectures. Most fusion codes have been optimized for serial vector processors. Conversion of these codes to perform optimally on parallel machines is a long and costly task.

From its design inception, the computationally intensive NIMROD physics kernel has been structured so as to enable a robust parallel implementation of the code. In fact the NIMROD kernel is a single source code; it runs without modification on single processor machines or any parallel platform that supports a message-passing style of programming. This includes workstations, traditional vector supercomputers, and all current-generation massively parallel machines such as the Cray T3E at NERSC. The key design decision that enabled this was the block-structure of the grid. As discussed above, NIMROD represents the poloidal simulation plane of a tokamak as a collection of adjoining grid blocks; the toroidal discretization is pseudo-spectral. Individual grid blocks are composed of regular rectangular grid cells or arbitrary collections of triangular elements. The chief motivation for this was to enable complex geometries to be gridded in a computationally efficient manner. However, it has also allowed us to rapidly implement NIMROD on parallel machines in a message-passing paradigm.

Individual processors are assigned one or more blocks (and their associated toroidal modes). All of the computational kernels in NIMROD perform operations on a block-by-block basis; in parallel these operations remain fast on-processor computation. The only inter-processor communication required is the exchange of values for block-edge grid points shared with other processors. To allow for arbitrary block connectivity, we have implemented this communication kernel with asynchronous, point-to-point message passing. A single, bundled message is exchanged between two processors who share two (or more) adjoining grid blocks. This block-connection operation is called from within NIMROD's implicit conjugate gradient solver and effectively parallelizes both the Jacobi and block-inversion preconditioners currently implemented in the code. We have successfully benchmarked NIMROD on the NERSC T3E and found that it runs scalably on large numbers of processors with only a minor communication cost overhead. Further work remains to be done to test the performance and convergence of the implicit solver and its preconditioners as a function of (a) the number of blocks used to grid the poloidal plane and (b) the number of processors used.

Validation

A detailed validation plan has been prepared for NIMROD. To date, NIMROD has been validated against shear and compressional Alfvén waves (including two-fluid effects) in straight and toroidal geometry, sound waves, magnetoacoustic waves, whistler waves, and linear instability and nonlinear saturation of an ideal internal kink mode in a straight cylindrical RFP. Tests of the linear MHD stability of ideal internal kink modes in non-circular, toroidal tokamak equilibria at both zero- and finite-β are underway.

Status

The status of NIMROD Version 2.0 is as follows:

1. The NIMROD physics kernel solves the nonlinear, three-dimensional two-fluid equations in toroidal geometry with arbitrary poloidal cross-sectional shape. The outer boundary must be a perfectly conducting wall. The equations are formulated as a single fluid model. The cold plasma response is complete (except for advection). The warm plasma response consists of a single adiabatic energy equation. A semi-implicit time advance is used. The physics kernel has been validated in shear and compressional Alfvén waves, sound waves, magnetoacoustic waves, whistler waves, and linear instability and nonlinear saturation of an ideal internal kink mode in a straight cylindrical RFP. Tests of the linear MHD stability of ideal internal kink modes in non-circular, toroidal tokamak equilibria at both zero- and finite-β are underway.
2. The grid generator and pre-processor can interact with files generated by a number of standard equilibrium codes. Grids with arbitrary configurations of RBLOCKs and TBLOCKs can be generated.
3. The post-processor consists of XDRAW for quick-look and diagnostic graphics, and IBM DataExplorer for animation and publication quality graphics. The physics kernel can generate files for either of these packages.
4. The GUI is functional. It can run and interact with jobs on remote computers. It controls the execution of the NIMROD Code System.

NIMROD Goals

The short term goals of the NIMROD team are:

1. Deliver a β-test version of NIMROD to General Atomics by late summer or early fall of 1997. This version will have undergone preliminary validation and be ready for more extensive validation tests in the customer's working environment.
2. Have a validated, nonlinear, parallel code with full 2-fluid effects complete and ready for further extensions (separatrix, vacuum region, heat transport, neoclassical effects, etc.) by July 1998.
3. Integrate more users into the team as NIMROD becomes a production tool.

Future Plans

During the next year we plan to confront the following technical issues:

1. Validation: We will continue to carry out our validation plan. This will require an intensive labor effort by the entire team. We hope to have a completely validated core code by July 1998.
2. Parallel Linear Solver: The performance of the linear solver will determine the performance of the overall code. It is of utmost importance to obtain an optimized linear equation solver for NIMROD. The solver must be parallel and should be capable of solving anti-symmetric systems that couple all toroidal modes. We hope to secure half of a post-doctoral scientist at Lawrence Livermore Laboratory to work on this problem.
3. Vacuum Region//Moving Separatrix: There is no restriction in the NIMROD grid that excludes the separatrix. Thus, external kinks and vertical motions of the plasma column can, in principle, be studied with NIMROD. There problems require a self-consistent treatment of the vacuum region, and an implementation of the matching conditions at the moving separatrix. One solution is to treat the vacuum as a highly resistive, highly viscous fluid. Another is to formulate the problem using a Green's function approach. We plan to begin to address this problem in the next year.
4. Resistive Wall: The present version of NIMROD assume a perfectly conducting boundary. Nonideal boundary conditions, such as those imposed by a resistive wall, are required to address such important problems as mode locking and the effect of field errors. These are easy to formulate in straight cylinder, where analytic external solutions are known. In a torus, numerical procedures, such as Green's function techniques, must be applied. We plan to begin to address the formulation of this problem in the next year.
5. Neoclassical effects and transport: Neoclassical effects and anisotropic thermal conduction must be included to study the problem of soft β-limits in advanced tokamaks. We hope to secure half of a post-doctoral scientist at the University of Wisconsin to begin to formulate this problem for NIMROD.
6. Documentation and Maintenance: In order to be a useful tool for the community the documentation for NIMROD must be complete and current. This is a major job.

We also need to implement a code management system, such as CVS, to be under the control of a code librarian.

---

1. T. H. Stix, *Waves in Plasmas*, American Institute of Physics, New York (1992).
2. D. Clausing, *Total Quality Development*, ASME Press, New York (1994).
3. N. A. Krall and A. W. Trivelpiece, *Principles of Plasma Physics*, p. 89ff, McGraw-Hill, New York (1973).
4. D. H. Harned and W. Kerner, J. Comp. Phys. **25**, 1855 (1985).
5. D. H. Harned and D. D. Schnack, J. Comp. Phys. **65**, 57 (1986).
6 D. D. Schnack, D. C. Barnes, Z. Mikić, D. H. Harned, and E. J. Caramana, J. Comp. Phys. **43**, 17 (1987).
7. F. L. Hinton and R. D. Hazeltine Rev. Mod. Phys **48**, 239 (1976).
8. S. P. Hirschman and D. J. Sigmar, Nucl. Fusion **9**, 1079 (1981).

PROPOSED BUDGET 7/97-7/98

Additional Expenses Only

ITEM	COST ($1000)
1. Support for Carl Sovinec (LANL)	$125
2. Parallel linear solver (LLNL)	50
3. GUI development, validation, documentation, coordination (SAIC)	50
4. Neoclassical effects, computational support (U. Wisc.)	50
5. Computer hardware (LANL)	15
6. Computer hardware (GA)	15
7. Validation, graphics (Keldish Institute, Moscow State University)	50
8. Documentation and maintenance	20
TOTAL	**$375**

PROPOSED MANPOWER ALLOCATION 7/97-7/98

Man-Years (FTE)

PERSONNEL	EFFORT	TASKS
A. Glasser (LANL)	0.75	Graphics, animation, formulation
D. Schnack (SAIC)	0.6	Coordination, formulation, validation, documentation
C. Sovinec (LANL)	0.75	Development, formulation, validation
H. Karandikar (SAIC)	0.1	Coordination, facilitation
A. Tarditi (SAIC)	0.5	GUI, validation, documentation
M. Chu (GA)	0.15	customer representative, validation
S. Plimpton (SNL/ABQ)	0.2	code optimization, parallel linear solver, parallelization
A. Koniges (LLNL)	0.1	Parallelization, linear solver
R. Nebel (LANL)	0.5	Validation, formulation
T. Gianakon (Cardarache)	0.1	GUI, neoclassical effects, validation
S. Kruger (U. Wisc.)	0.1	GUI, neoclassical effects, librarian
Post-doc (U. Wisc.)	0.5	Neoclassical effects, transport
Post-doc (LLNL)	0.5	Parallel linear solver
TOTAL	**4.85**	

We propose to maintain this inflation adjusted level of support for the 2-3 year time frame.

Total manpower effort 2/96-7/97: 4.8 FTE, or 3.2 FTE/year