DOE/ER/54297-- T1

Progress report for

Neoclassical Transport of Energetic Particles in Asymmetric Toroidal Plasma

John R. Cary, Principal Investigator
Center for Integrated Plasma Studies and Department of Physics
University of Colorado
Boulder, CO 80309-0390

October, 1997

I. Introduction

During the most recent funding period we obtained results important for helical confinement systems and in the use of modern computational methods for modeling of fusion systems. Our most recent results include showing that the set of magnetic field functions that are omnigenous (i.e., the bounce-average drift lies within the flux surface) and, therefore, have good transport properties, is much larger than the set of quasihelical systems. This is important as quasihelical systems exist only for large aspect ratio. This was work was published as a Physical Review Letter[1] and is in press in the Physics of Plasmas. We have also carried out extensive earlier work on developing integrable three-dimensional magnetic fields,[2] on trajectories in three-dimensional configurations,[3,4] and on the existence[5] of three-dimensional MHD equilibria close to vacuum integrable fields. At the same time we have been investigating the use of object oriented methods for scientific computing. Our recent paper[6] comparing Fortran 90 and C++ for object oriented scientific computing has been accepted for publication at Computer Physics Communications and is currently being downloaded 30 times per week over the web.

II. Transport in helical plasma confinement systems

In helical plasma confinement systems, an overriding issue has been neoclassical transport. The reason is that neoclassical transport is known to be particularly large due to the presence of particles with large orbits, i.e., whose deviation off the flux surface due to guiding-center drifts is large. Indeed, in the absence of strong plasma potential, particles can drift directly to the walls. For traditional machines, neoclassical transport is so large that it is dominant - experiments have found that neoclassical transport explains loss data for helical devices while tokamaks remain dominated by turbulent transport. Thus, for the international program, which includes the new large helical machines at Toki (LHD) and at Griswald, developing an understanding of neoclassical transport is critical.

DISCLAIMER

This is now becoming critically important to the US program also. First off, for the US to keep abreast of international experiments, we must retain significant expertise in the US. Secondly, there are a large number of proposed or design studies of new helical experiments at many US institutions. Examples are the quasihelical stellarator HSX at Wisconsin, the studies of spherical tokamak/stellarator hybrids at Texas and ORNL, and the quasitoroidal stellarator (MHH2) studies at NYU that are being further investigated for possible devices at Auburn and PPPL. With all of this proposal activity and the additional activity on the international front, it is important that the US rebuild its theoretical expertise in helical systems, expertise that was lost in the narrowing of the program in the last decade.

In the last year the PI and co-I of this grant showed that helical systems with good transport can be obtained more easily than previously believed. Their work showed that there exist omnigenous magnetic functions, for which the bounce-averaged drift across the flux surfaces vanishes, and that the requirements for obtaining such systems are much less restrictive than those for obtaining quasihelical systems.

In the next funding period we propose to continue our investigations into helical systems with improved confinement. Our research will include investigations into whether the requirements can be satisfied in a near-axis expansion of the MHD equilibrium, and into the transport theory for omnigenous systems. This transport theory is not simply obtainable by taking limits of existing multiple-helicity theories, as they are dominated by transitioning particles, which do not exist in omnigenous configurations. The development of transport theory for omnigenous systems should answer fundamental questions, such as whether transport in omnigenous systems is ambipolar.

A. Background: transport in helical systems

Insight into the nature of guiding-center motion can be obtained by first looking at the parallel motion, i.e., by neglecting the drifts. Thus, as is well known, the motion is that in the one-dimensional effective potential $e\Phi^* \equiv e\Phi + \mu B$. Furthermore, the electrostatic potential is found to be a function of only the magnetic surface variable ψ to lowest order, and so one need only study the variation of the magnetic strength along the field line to determine the basic types of orbits.

In general, the magnetic field strength has a rich Fourier representation. However, for toroidal stellarators, there are usually two harmonics that dominate. One is that due to toroidicity, which introduces a $\cos(\theta)$ variation. The other is due to the helical field and has a cos($\ell\theta$-Nφ) variation. In sum, these give a magnetic strength that dominantly varies according to

$$B = B_0(\psi)\left[1 - \varepsilon_t \cos(\theta) + \varepsilon_r \cos(\ell\theta - N\varphi)\right] \tag{1}$$

.

The flux increases as area near the axis. Hence, the flux radius,

$$r \equiv \sqrt{2\psi/B_0(0)}, \tag{2}$$

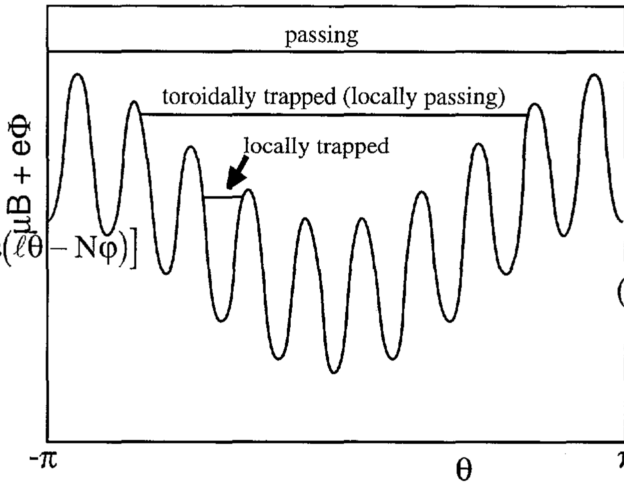

Fig. 1. Variation of the magnetic field along a field line for the two-helicity stellarator. Such systems have three types of particles: passing, toroidally trapped, and locally trapped.

also known as the circularized radius of the flux surfaces, scales as distance. From this and analyticity it follows that relative strength ε_t of the toroidicity harmonic varies near the magnetic axis as r, while the relative strength ε_r of the helical ripple amplitude varies as r^{ℓ}.

A plot of the field strength along a field line,

$$\varphi = \varphi_0 + q\theta, \tag{3}$$

is shown in Fig. 1. That is, Fig. 1 shows a plot of

$$B = B_0\left[1 - \varepsilon_t \cos(\theta) + \varepsilon_r \cos((\ell - Nq)\theta - N\varphi_0)\right]. \tag{4}$$

This plot shows that there are three types of particles, the locally trapping particles, which bounce back an forth between two local maxima produced by the helical variation of the field; the toroidally trapped particles, which pass over at least one local maximum and are, therefore, reflected by the toroidal variation of the field; and the passing particles, which are not reflected.

Drifts significantly complicate the situation. Locally trapped particles experience a bounce averaged drift that crosses flux surfaces. Thus, locally trapped particles can convect out of the plasma. This causes large transport, which we will discuss momentarily. In addition, drifts cause the field line label φ_0 to change. This changes the phase φ_0 of the ripple. Both cross-surface drifts and within-surface drifts lead to variation of the phase space area inside the separatrix[7] and, thus, to transitions, i.e., changes of state. Such transitions lead to chaotic motion and enhanced transport at low collision frequency. In addition, such transitions cause all toroidally trapped particles to eventually cycle through the locally trapped state. Thus, the fraction of particles participating in the convective transport mechanism is much larger than just the class of locally trapped particles.

These transport mechanisms are not ambipolar; the ions and electrons convect at different rates, and their trajectory sizes are not proportional to their momenta. Thus, a plasma potential is created. The plasma potential causes the trajectories of the bulk plasma particles to become closed, though not to be as small as they would be without the cross-surface drifts. Because the drift trajectory widths now depend on the potential, one can find, as does the plasma, an *ambipolar potential*, at which the outward flow of ions and electrons is equal.[8] (Of course, the ambipolar potential is largely irrelevant for the high-energy particles, such as fusion-product alphas, as their energy is much greater than the that of a bulk plasma particle.)

In any case, the particle flow can finally be found, and the resulting transport coefficients can be very large. Moreover, as the collisionality is reduced below the locally trapped particle bounce frequency (appropriately normalized for the smallness of collision needed to change state) the transport begins to increase, because the direct convection of particles then has a greater effect.

To address these problems, Nuhrenberg and Zille proposed that stellarators be made quasihelical. They showed numerical results for MHD equilibria for which the magnetic field strength B nearly was a function of only a single linear combination of the flux angles. That is,

$$B \approx B_0\left[1 - \varepsilon_r \cos(\ell\theta - N\varphi)\right]$$

.

(In fact, one can show that the poloidal wave number must be unity if one is to have quasihelicity near the magnetic axis.) For such systems one can show that there is a rigorous guiding-center invariant and, because of its form, that the particle trajectories upon bounce av-

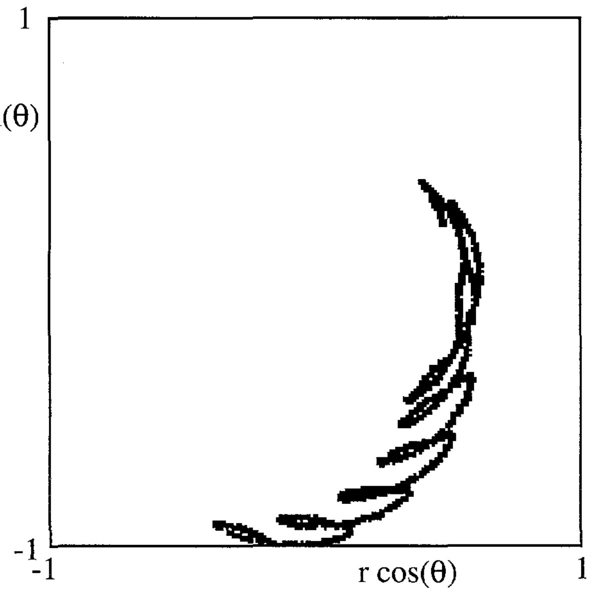

Fig. 2. Typical trapped particle trajectory in a two-helicity stellarator. Without a plasma potential, particles drift directly out of the plasma.

eraging do not drift across a flux surface. Thus, for such systems the neoclassical transport should be greatly reduced.

The original configurations found by Nuhrenberg and Zille were at quite large aspect ratios. It was hoped that systems having lower aspect ratios could be found, but the work of Garren and Boozer indicates that quasihelicity must worsen as the aspect ratio is reduced. The calculated MHD equilibria by expansion in distance from the magnetic axis attempting to impose the condition of quasihelicity at each order. They found that at third order quasihelicity cannot be imposed. Thus, exactly quasihelical systems cannot exist, and, practically speaking, at lower aspect ratios the degree of non quasihelicity must increase.

In addition, there has been activity on the front of finding *quasitoroidal* stellarators.[9] These are stellarators, but for which the coefficients of the magnetic strength harmonics for nonzero toroidal number N are small. In order to have nonzero rotational transform near the axis, some of the nonzero-N harmonics are finite, but over the entire radius of the plasma the nonzero-N harmonics are much smaller than the usual toroidal harmonics.

B. Recent results in obtaining systems with reduced neoclassical transport

The analysis of particle orbits in strong magnetic fields relies on the guiding-center approximation. A Hamiltonian description of guiding-center motion, needed for obtaining the drift surfaces, is given by the phase-space Lagrangian. Here we summarize the phase-space Lagrangian, how it can be used to analyze guiding-center trajectories, and how such analysis can be used to determine the structure of magnetic fields that have good transport properties.

In Boozer coordinates, a special set of flux coordinates for which the angular covariant components of the magnetic field are flux functions, the Lagrangian determining the motion of guiding-center is

$$L_{gc} = \left(muB_{\psi}/B\right)\dot{\psi} + \left(e\psi/c + muB_{\theta}/B\right)\dot{\theta} + \left(eA_{\varphi}/c + muB_{\varphi}/B\right)\dot{\varphi} - h, \tag{6}$$

where the Hamiltonian h is given by

$$\mathrm{h} = \tfrac{1}{2}\mathrm{mu}^2 + \mu\mathrm{B} + \mathrm{e}\Phi, \tag{7}$$

and B_i and A_i are the *covariant* components of these vectors. This is a phase-space Lagrangian of the four variables (ψ,θ,φ,u). One can show that the dynamics given by such a Lagrangian is analogous to that in usual Hamiltonian theory. There are a full set of Poincare invariants, and there is a Noether theorem.

One can prove that the character of the motion in this Lagrangian depends on only the magnetic strength function B(ψ,θ,φ). This follows because in vacuum fields B_ψ and B_θ vanish, while B_φ is constant, while for MHD equilibria, the helicities appearing in B_ψ are the same as those that appear in B_φ. For the case where B is a flux function, the system is said to be *isodynamic.*[10] For the case where B contains is a function of only one helicity, say $\ell\theta - N\varphi$, there is a corresponding helical invariant,

$$P_h = \frac{\partial L_{gc}}{\partial \dot{\theta}} = \frac{e}{c}\left(\psi + \frac{\ell A_\varphi}{N}\right) + \frac{mu}{B}\left(B_\theta + \frac{\ell B_\varphi}{N}\right). \tag{8}$$

The existence of this invariant implies that the trajectories do not deviate far from the invariant surface, and that the bounce averaged drift vanishes. The first fact follows because the first term on the right side of Eq. (8) can vary no more than the second term, which varies only due to the oscillations of the parallel velocity. The second fact holds because this invariant is a function of only the single helicity. Hence, after one bounce one must return to the same place. Systems having such symmetry are termed quasihelical or quasisymmetric. Zille and Nuhrenberg found approximately quasihelical systems for large aspect ratio.

However, one need not have this symmetry to still have very good trajectories. Hall and McNamara introduced the term *omnigenous* to describe systems for which the bounce averaged drift across flux surfaces vanishes. For such systems the characteristic step size for neoclassical transport remains small.

Our recent work[1,4] has illustrated that there exist mod-B functions having the omnigenity property, yet which are not quasihelical. Our analysis finds the mathematical restrictions on the function |B(ψ,θ,φ)|. A sample trajectory for a nontrivial magnetic strength function is shown in Fig. 3. One can see that such a trajectory does not drift directly out of the machine. We refer the reader to our paper, which is also in the appendices, for the details of the construction of this magnetic field. Here we elaborate on the conclusions.

Our derivation shows, first of all, that an omnigenous magnetic field must, on each surface, have all magnetic maxima of the same value and all magnetic minima of the same value. Beyond that there is a great deal of freedom. Any two dimensional function may be chosen for the magnetic field strength on one half of a flux surface, and then the values are determined on the other half.

These results show that there is significantly greater freedom in omnigenous systems than in the previously sought quasihelical systems. Quasihelical systems have, on each flux surface, the freedom of a function of only one variable, the magnetic field as a function of the helical angle. In contrast, omnigenous systems have the freedom of a two dimensional function subject to a parity-like constraint and the constraints associated with the maxima and minima. Thus, satisfying the omnigenity constraint should be much easier than attempting to satisfy the quasihelical constraint.

Our results also imply that optimizations should attempt to minimize different residuals. Previously, the amplitudes of all magnetic harmonics $B_{m,n}$ for $(m,n) \neq k(\ell, N)$ were minimized. Our results show that many of the harmonics do not damage the property of omnigenity. Indeed, in the omnigenous system analyzed in Figs. 2-3, the nonconforming harmonics have strengths that are 1/4-1/3 that of the main helical amplitude.

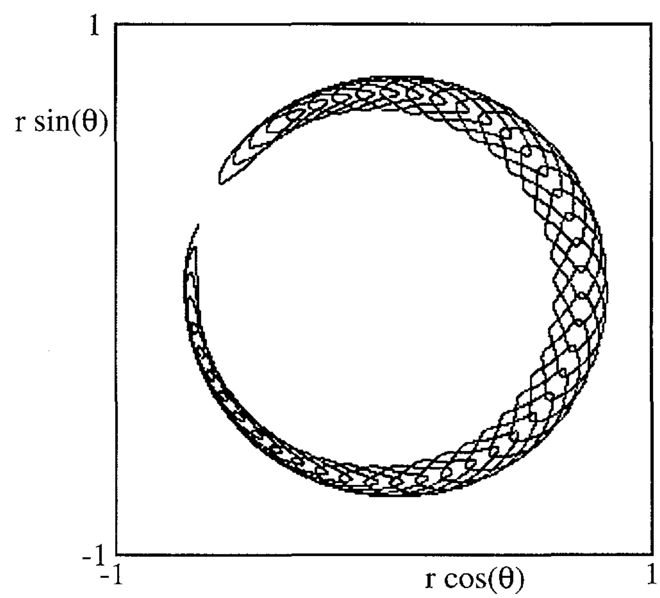

Fig. 3. Trajectory of a trapped particle in an omnigenous magnetic field. The trajectory remains near its original flux surface; thus the magnetic field is omnigenous. The lack of symmetry is manifest in the variation of the trajectory width as the particle drifts poloidally.

III. Advanced computational methods

There has been a wave of activity in the computational community of late with the investigation of and adoption of new methods of structuring analysis applications that have come out of the computer science community. Illustrative of this are the recent DOE programs, such as the ACTS toolkit[11] development program, which is geared towards using new technologies for software frameworks, numerical kernels, and runtime support. These developments are primarily being carried out using object oriented programming techniques, both in compiled languages like C++ and scripting systems like Python. The PI and co-I have been working along these lines for some time. Indeed, their discussion of scientific object oriented computing is in press[12] and will appear later this year.

Over the past year, the PI was involved in discussions around how the fusion community can most easily move in the direction of modern computing, not only per the items above but also to take advantage of the new computer architectures becoming available and to develop applications that can be more easily used through development of intuitive graphical user interfaces (GUI's). These discussions led to a white paper[13] on this subject, and further discussions resulted in a proposal[14] to the office of Mathematical, Information, and Computer Sciences of DOE. These discussion lay a groundwork for future developments in the field.

In the coming project period, we intend to continue our explorations into the use of object oriented methods for scientific computation. We are members of a collaboration with GA, LLNL, and possibly others to develop a transport modeling application based on modern scripting and object oriented methods. We also propose to develop a class library for Monte Carlo modeling of guiding centers in arbitrary magnetic fields. Such a library could be used for delta-f calculations of transport in asymmetric confinement devices.

A. Object oriented programming

The advantages of object oriented programming.[15-18] are widely described in the literature and have been acknowledged by the software and science community. At one level, object oriented programming consists of designing *classes*. A class specifies the structure of a data object and how other code can interact with the data. An *object* is a particular instantiation of a class; an object has its own data. A program is put together by first defining the classes. Next, the program instantiates the classes and by calling the *methods* of the classes, manipulates or displays the data. The set of public methods is called an *interface* of the class. In the C++ paradigm, user intervention can consist of commands that are passed to the classes through the interface. The user can use and reuse the classes in different combinations and also change them easily, employing *inheritance* and *overriding*. These methods allow one to add and override functionality of classes without writing new classes from scratch.

Many object oriented languages provide the capabilities mentioned above. For several reasons, our language of choice is C++. First of all, C++ provides the power to do scientific calculations and write large scale and structurally complicated applications. Second, only C++ has templates, which will allow one to optimize operations by fusing loops and eliminating creation of unnecessary temporaries. Third, of all OOP languages, C++ is the mostly widely accepted and used in the scientific community; it has been adopted for most of the DOE/ACTS projects. Other languages, auch as Eiffel or Smalltalk, are not used by many scientists. The new OOP language Java[19] is far from maturity, does not support operator overloading, and does not have the template mechanism necessary for efficient scientific object oriented programming. Fortran90[20] is not a fully object oriented language, since it does not have a single simple construct to support abstract data types and does not provide for inheritance.[12]

B. Transport modeling

In the next project period we will participate in a collaboration to develop a modern, object oriented transport modeling application. This is an initiative that GA and LLNL have committed resources to. There is a recognized need to develop modeling software that is more user friendly, more easily maintained, and that more easily interacts with existing data. Furthermore, such a collaboration could act as a test bed for remote collaboration technologies.

In the current case, the goal is to combine modern object oriented programming methods with scripting technologies, perhaps through use of Python.[21] Object oriented methods would be ideal for a transport application, as there is a great deal of complexity in such an application. Fueling can be through neutral beams or pellet injection, there are a multiplicity of species, ions, electron, fusion products, etc., that have differing dynamics, and there are a multiplicity of processes, e.g., neoclasssical transport and turbulent transport, that govern the dynamics. Scripting technologies can be used to steer an application. For example, by interacting through the command interface, the modeler can choose to stop the dynamics and insert a new transport model at the point where discrepancies are observed.

This collaboration is only at the early discussion stage at the writing of this proposal. We intend to be involved in this discussion to learn more about the range of possibilities in such applications. We will take on the task of writing one or more C++ classes for this application once the interfaces are specified.

C. Modeling of guiding centers

Our work on transport has led us to begin the creation of a class library for Monte Carlo studies of guiding centers. Our class library was used to perform the simulations whose results are shown in Figs. 2-3. We intend to make our class library publicly available, so that others developing OOP guiding-center simulations will be able to use our code in their work. The advantage of OOP in this context is that one defines a standard set of interfaces, and much of the coding is unchanged as one changes from one model to another. Moreover, other users may redefine methods but use much of the same structure to simulate different systems.

One aspect of our library is the simple hierarchy shown in Fig. 4. The base class, `MagFld`, actually is based on a much more complicated hierarchy for defining systems depending on a certain number of parameters. Thus, the base classes above `MagFld` define all I/O and descriptor data. To the higher base classes, `MagFld` adds the interfaces needed for describing the magnetic field: covariant, Clebsch, and contravariant components, Jacobian, magnetic field magnitude. Software reuses is obtained by defining many functions in the base class, such as the one for obtaining the magnitude of B from its covariant and contravariant components. Of course, users can override these methods to obtain faster or more specific methods.

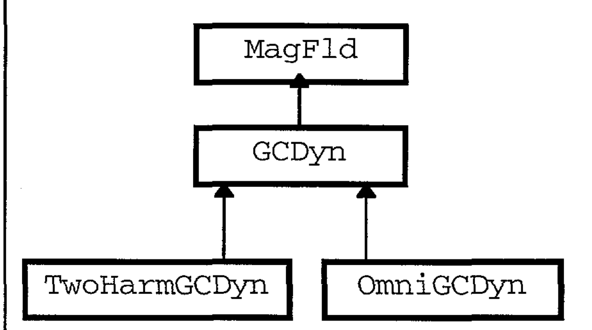

Fig. 4. Inheritance structure for guiding-center modeling.

From this base class we derive the `GCDyn` class, which defines the interface for what is needed to integrate guiding centers. Now one must add interfaces for obtaining the electric potential, the gradient of the magnetic strength, the current two-form, etc. These basic methods must be defined, then the class contains other universal methods for calculating other needed quantities. An obvious example is the calculation of the time derivatives of the guiding center variables.

The advantage of such a class library comes when one simulates specific systems. Two such systems are given at the bottom level of the hierarchy of Fig. 4. The first, `TwoHarmGCDyn`, provides the appropriate equations for the case of the standard stellarator model (containing two harmonics in the magnetic strength) in Boozer coordinates. The second, `OmniGCDyn`, provides the dynamics for omnigenous magnetic fields of the type we investigated in our last funding period. The OOP method was advantageous here, because we were able to set up the two separate cases yet use them interchangeably in our analysis application. The instantiation sets up the particular system, but after that all I/O, graphics, integrations, etc. proceed identically.

The hierarchy has been started in such a way that one can derive from these classes those needed for simulating, e.g., weighted particles. Such simulations will be needed for delta-f simulations to obtain transport coefficients.[22] Delta-f simulations permit one to represent only the deviation of a distribution from a Maxwellian by particles. In derived classes one would add data describing the background distribution so that one could give the equation for the weights, and one would add the interfaces for interacting with these data. However, much of the coding would remain the same (the particle equations of motion are unchanged in delta-f calculations.)

---

1 J. R. Cary and S. G. Shasharina, Phys. Rev. Lett.**78**, 674 (1997); Phys. Plasmas, In press (1997).

2 J. R. Cary and J. D. Hanson, Phys. Fluids **29**, 2464 (1986); J. D. Hanson and J. R. Cary, Phys. Fluids **27**, 767 (1984); J. R. Cary, Phys. Fluids **27**, 119 (1984).

3 J. R. Cary, C. L. Hedrick, and J. S. Tolliver, Phys Fluids **31**, 1586 (1988).

4 J. R. Cary and S. G. Shasharina, Phys. Fluids B**5** (7), 2098-2121 (1993).

5 J. R. Cary and M. Kotschenreuther, Phys. Fluids **28** (5), 1392-1401 (1985); J. D. Hanson and John R. Cary, Physics of Plasmas **1** (11), 3713-3715 (1994).

6 J. R. Cary, S. Shasharina, J. C. Cummings, J. V. W. Reynders, and P. J. Hinker, "Comparison of C++ and Fortran 90 for Object-Oriented Scientific Programming," Comp. Phys. Comm, in press (1997). Available on the web at http://beams.Colorado.EDU/~cary/papers/CompCPP_F90SciOOP.html.

7 D. Dobrott and J. M. Greene, Phys. Fluids **14**, 1525 (1971).

8 L. M. Kovrizhnykh, Nucl. Fusion **24**, 851 (1984).

9 P. R. Garabedian, Phys. Plasmas **3**, 2483 (1996).

[10] D. Palumbo, Nuovo Cimento B X53, 507 (1968).

[11] SciTL, The Scientific Template Library, available on the web at http://www.acl.lanl.gov/SciTL/.

[12] J. R. Cary, S. Shasharina, J. C. Cummings, J. V. W. Reynders, and P. J. Hinker, "Comparison of C++ and Fortran 90 for Object-Oriented Scientific Programming," Comp. Phys. Comm, in press (1997). Available on the web at http://beams.Colorado.EDU/~cary/papers/CompCPP_F90SciOOP.html.

[13] B. Braams, J. Cary, V. Chan, R. Cohen, W. Dorland, W. Houlberg, S. Jardin, D. Schnack, "Needs and Opportunities in Computational Fusion Science", on the web at http://beams.colorado.edu/~cary/NeedsOppsCompFusion.html.

[14] http://www.pppl.gov/nmfc/proposal.html.

[15] B. Stroustrup, *The C++ progamming lanquage, second edition* (Addison-Wesley, Reading, Massachusetts, 1991).

[16] S. Lippman, *C++ Primer, second edition* (Addison-Wesley, Reading, Massachusetts, 1991).

[17] E. Gamma, R. Helm et. al, *Design Patterns* (Addison-Wesley, Reading, Massachusetts, 1995); A. Eliëns, *Principles of Object-Oriented Software Development* (Addison-Wesley, Reading, Massachusetts, 1994).

[18] G. Booch, *Object Oriented Development*, IEEE Transactions on Software Engineering **12**, 211 (1986); Also discussed in Ref. 7, p. 93.

[19] S. R. Davis, *Learn Java Now*, Microsoft Press, 1996.

[20] T. M. Ellis, I. R. Philops,and T. M. Lahey, *Fortran 90 Programming* (Adison-Wesley Publishing Group, Reading, MA, 1995).

[21] http://www.python.org.

[22] Z. Lin, W. M. Tang, W. W. Lee, Phys. Plasmas **2**, 2975 (1995); M. Sasinowski and A. H. Boozer, Phys. Plasmas **2**, 610 (1995).