JOURNAL OF COMPUTATIONAL PHYSICS **142,** 473–488 (1998)
ARTICLE NO. CP985947

An Improved Convection Scheme Applied to Recombining Divertor Plasma Flows

D. A. Knoll

Applied Theoretical and Computational Physics Division, Los Alamos National Laboratory,
Los Alamos, New Mexico 87545
E-mail: nol@lanl.gov

Received July 21, 1997; revised February 12, 1998

A high-order finite volume convection scheme, in conjunction with a monotonicity preserving flux-limiter is applied to a combined tokamak edge plasma/Navier–Stokes neutral transport model. This is a highly nonlinear system of convection–diffusion–reaction equations which describe the partially ionized boundary layer plasma of a tokamak fusion reactor. The solutions of interest contain a sharp ionization front. The improved convective discretization is applied within the context of the existing matrix-free Newton–Krylov solution algorithm. More accurate convective differencing is shown to make a significant difference on a problem of current interest. It is demonstrated that a matrix-free Newton–Krylov implementation, where the preconditioner is derived using first-order upwind convective differencing, provides savings in both memory requirements and CPU time. © 1998 Academic Press

Key Words: high-order convective differencing, Newton–Krylov methods, tokamak divertor plasma.

1. INTRODUCTION

In this paper, we study the application of a high-order, monotone, convection scheme to an advanced fluid transport model of a partially ionized tokamak edge plasma. The tokamak is a toroidally shaped magnetic confinement fusion device [1], and the tokamak edge plasma fluid equations are a highly nonlinear set of convection–diffusion–reaction equations which describe the boundary layer plasma of this device. They contain widely varying time and spatial scales, and the transport coefficients and reaction rates are strong functions of density and temperature. These equations describe the flow of plasma particles and energy from the edge of the reactor core into what is called the divertor region, which serves as the charged particle exhaust system.

The original solution method applied to the two-dimensional edge plasma fluid equations was a pressure-correction, SIMPLE [2] based, segregated solution algorithm [3] with density, velocity, and temperature as dependent variables. A staggered grid and finite volume discretization were employed, with thermodynamic variables at cell centers and velocities at cell faces. Convective differencing was done with a first-order method such as pure upwind, the hybrid method, or the power-law method [2]. More recently a fully coupled Newton–Krylov algorithm has been developed for improved nonlinear convergence performance [4–7]. However, the dependent variables, the staggered grid, and the first-order convective differencing have been maintained. In this paper we study the application of a high-order, monotone, convection scheme within the context of the existing staggered grid and dependent variables, as well as the existing Newton–Krylov algorithm. The high-order convection scheme is applied to an advanced transport model for recombining divertor plasmas which includes a Navier–Stokes treatment of the neutral component [8].

For discretization of the convective terms, the so-called QUICK (quadratic upstream interpolation for convective kinematics) scheme is used [9]. This is an upwind biased, 3-point interpolation scheme for evaluating cell face values from cell centered values. In order to preserve monotonicity, the QUICK scheme is flux-limited. The flux-limiting strategy we employ is the SMART (sharp and monotonic algorithm for realistic transport) scheme of Gaskel and Lau [10]. Standard central differencing is used for the diffusive fluxes.

The implementation of higher-order, monotone, convection schemes within implicit solution algorithms has been met with some challenges [11, 12]. These challenges are a result of the increased storage due to a larger numerical stencil, lack of diagonal dominance in the solution matrix, and lack of “analytical differentiability” of some of the limiters. We will demonstrate that a matrix-free Newton–Krylov algorithm provides a natural frame work for the inclusion of higher order, monotone, convection schemes. It will also be demonstrated that including higher order convective differencing can make a significant impact on divertor simulations of current interest.

The remainder of this paper is outlined as follows. Section 2 discusses the convective discretization, the Newton–Krylov algorithm, and how they fit together. While the convective differencing algorithm used in this paper is not new, its use on the edge plasma fluid equations and within the edge plasma modeling community is new. For these reasons we present some details of the discretization. Section 3 describes the physical model and geometry. Section 4 contains computational results and looks at algorithm performance, and Section 5 provides a summary and conclusions.

2. CONVECTIVE DISCRETIZATION AND SOLUTION ALGORITHM

2.1. *Convective Discretization*

With the motivation to clearly illustrate the implementation of this discretization we go into detail on a one-dimensional (1D) steady-state convection–diffusion–reaction equation. This material is not new but is included for self-consistency. Consider finite volume discretization of

$$u\frac{\partial\phi}{\partial x} - D\frac{\partial\phi^2}{\partial x^2} = S(\phi), \tag{1}$$

with u the convecting velocity, D a diffusion coefficient, S a source/sink function, and ϕ the dependent variable. The finite volume method balances fluxes in and out of cell faces

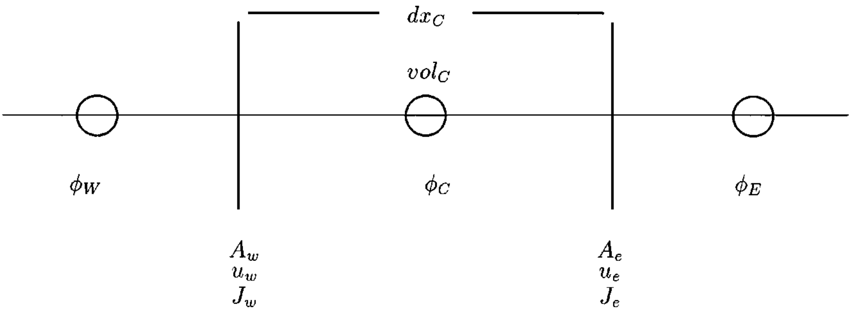

FIG. 1. Schematic representation of 1D grid.

with the sources or sinks resulting in

$$J_e - J_w = vol_c \cdot S(\phi_C), \tag{2}$$

where

$$J_e = A_e\left[u_e\phi_e^* - D \cdot (\phi_E - \phi_C)/dx\right] \tag{3}$$

and

$$J_w = A_w\left[u_w\phi_w^* - D \cdot (\phi_C - \phi_W)/dx\right]. \tag{4}$$

This can be seen schematically in Fig. 1. Upper case letters denote cell-centered quantities and lower case letters denote cell face quantities (in 1D $vol = dx$). The face values of the dependent variable ϕ must be determined for use in the convective operator. For flow directed to the right, QUICK [9] evaluates the east face value as

$$\phi_e^* = (\phi_C + \phi_E)/2 - (\phi_E - 2\phi_C + \phi_W)/8. \tag{5}$$

QUICK itself is not monotonic, which means that ϕ_e^* is not guaranteed to be bounded by ϕ_E and ϕ_C. Since monotonicity is a desired property, a flux-limiter must be applied to ensure monotonicity, and in this paper the so-called SMART scheme [10] is employed for this purpose. The SMART limiter is first demonstrated graphically, with the aid of Lenard's normalized variable diagram (NVD) [10]. Defining two nondimensional variables,

$$\tilde{\phi}_f = \frac{\phi_e^* + \phi_W}{\phi_E - \phi_W} \tag{6}$$

and

$$\tilde{\phi}_c = \frac{\phi_C + \phi_W}{\phi_E - \phi_W}, \tag{7}$$

the NVD, with the SMART limiter, is shown in Fig. 2. The line $\tilde{\phi}_f = \tilde{\phi}_c$ is first-order upwind differencing, the dashed line is QUICK differencing, and the dashed double dot line is the SMART limiter. Anywhere along the upwind line or within the triangle, ΔAED, ensures a choice of ϕ_e^* which is bounded, i.e. monotone [10]. Given $\tilde{\phi}_c$, and the piecewise linear SMART limiter we have a unique value for $\tilde{\phi}_f$, and thus ϕ_e^*.

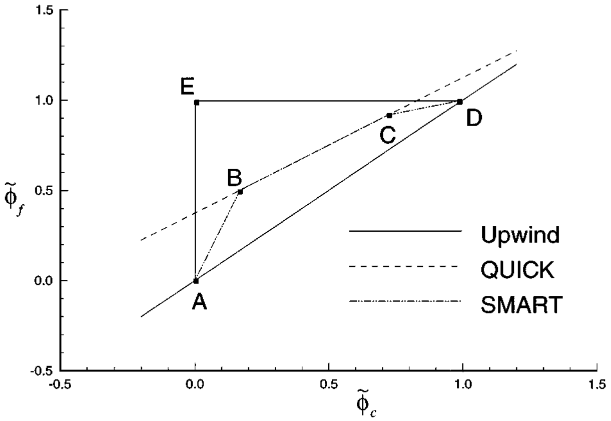

FIG. 2. Normalized variable diagram.

Next some insight is given as to the structure of the limiter. In choosing a face value for a convected variable, one possible choice is to use QUICK inside the triangle (ΔAED), and first-order upwind (the solid line) outside the triangle. However, this may cause problems around $\tilde{\phi}_c = 0$ and for $\frac{5}{6} < \tilde{\phi}_c < 1$. An indeterminacy results around $\tilde{\phi}_c = 0$ since one can get two distinct values as $\tilde{\phi}_c \rightarrow 0^+$ and $\tilde{\phi}_c \rightarrow 0^-$. Following the prescription of Gaskell and Lau [10] the line segment $\overline{AB}$ is followed, where $B = (\frac{1}{6}, \frac{1}{2})$. In the region $\frac{5}{6} < \tilde{\phi}_c < 1$, choosing $\tilde{\phi}_f = 1$ is equivalent to downwinding and can result in a loss of coupling of ϕ_C (the cell center value) to the volume convective flux balance (i.e., nearly zero diagonal for convective dominated flow). To avoid this problem the line segment $\overline{CD}$ is followed where the coordinates of C are chosen so as to define the slope of $\overline{CD}$ equal to 0.3. For more details see Gaskell and Lau [10]. This prescription defines a unique path through the NVD and, thus, a unique value of ϕ_e^* given $\tilde{\phi}_c$. Other choices are possible for the points B and C, but we do not address that issue here.

Given this detail, we now demonstrate how these ideas are implemented algorithmically. Again, this is not new material [13] but it is included for self-consistency. The following function is defined

$$fmed(\phi_1, \phi_2, \phi_3) = min[max(\phi_1, \phi_2), max(\phi_3, min(\phi_1, \phi_2))]. \tag{8}$$

Depending on the exact locations of points B and C, line segment $\overline{AB}$ has a slope we will call $slp1$ and line segment $\overline{CD}$ has a slope we will call $slp2$. First, three temporary values of ϕ_e^*, based on the QUICK line, line segment $\overline{AB}$, and line segment $\overline{CD}$, respectively are defined

$$\phi_{tmp1} = (\phi_C + \phi_E)/2 - (\phi_E - 2\phi_C + \phi_W)/8, \tag{9}$$

$$\phi_{tmp2} = slp1 * \phi_C + (1 - slp1) * \phi_W, \tag{10}$$

$$\phi_{tmp3} = slp2 * \phi_E + (1 - slp2) * \phi_C. \tag{11}$$

Next, a temporary value is chosen based on *fmed* operating on ϕ_{tmp2}, ϕ_{tmp3}, and the first order upwind value, ϕ_C,

$$\phi_{tmp4} = fmed(\phi_C, \phi_{tmp2}, \phi_{tmp3}). \tag{12}$$

Then a final determination is made between the QUICK line, first-order upwinding, and ϕ_{tmp4},

$$\phi_e^* = fmed(\phi_C, \phi_{tmp4}, \phi_{tmp1}). \tag{13}$$

At this point a unique face value, ϕ_e^*, has been chosen.

2.2. *Solution Algorithm*

We briefly outline the solution algorithm with more details given in Refs. [6, 7]. Application of Newton's method requires the solution of the linear system, $\mathbf{J^k}\delta\mathbf{x^k} = -\mathbf{F}(\mathbf{x^k})$, where $\mathbf{J}$ is the Jacobian matrix, $\mathbf{F}(\mathbf{x})$ is the nonlinear system of equations, and $\mathbf{x}$ is the state vector. The new solution approximation at iteration $k+1$ is obtained from, $\mathbf{x}^{k+1} = \mathbf{x}^k + s\delta\mathbf{x^k}$, where s is a damping scalar, which is adaptively chosen to be less than or equal to one. A pseudo-transient relaxation technique is used to increase the radius of convergence, and the time step is adaptively varied based upon the current level of nonlinear convergence.

We use the restarted generalized minimal residual (GMRES) algorithm [14] to solve the linear problem on each Newton step. The dimension of the Krylov subspace was chosen to be 40 for the calculations presented here (i.e., GMRES (40)), and the linear system is preconditioned with an ILU-based preconditioner. An "inexact" Newton's method linear convergence criteria is used. Specifically, the GMRES iteration is assumed converged when

$$\frac{\|\mathbf{J^k}\delta\mathbf{x^k} + \mathbf{F}(\mathbf{x^k})\|_2}{\|\mathbf{F}(\mathbf{x^k})\|_2} < \gamma_k. \tag{14}$$

The preconditioned GMRES algorithm requires the action of the Jacobian only in the form of matrix–vector products, which may be approximated by [15],

$$\mathbf{Jv} \approx \frac{\mathbf{F}(\mathbf{x} + \epsilon\mathbf{v}) - \mathbf{F}(\mathbf{x})}{\epsilon}, \tag{15}$$

where $\mathbf{v}$ is a Krylov vector (in GMRES), and ϵ is a small perturbation given by

$$\epsilon = \frac{1}{N\|\mathbf{v}\|_2}\sum_{m=1}^{N}(a|x_m| + a), \tag{16}$$

where N is the system dimension and a is a constant whose magnitude is on the order of the square root of machine roundoff.

Equation (15) enables the action of the Jacobian without explicitly forming or storing the matrix. This property can be extremely advantageous in problems where forming the Jacobian represents a significant fraction of the total CPU time and/or storing the Jacobian matrix is prohibitive. In many instances, however, the Jacobian or parts thereof are still needed to generate an effective preconditioning matrix, $\mathbf{P}$. In this situation, one of the primary advantages of this matrix-free Newton–Krylov implementation may lie in reducing the total number of required discrete function evaluations by amortizing the cost of forming the preconditioner over several Newton steps [6, 7].

When considering the use of high-order, monotone, convection schemes with a fully implicit Newton's method, the following concerns arise:

1. The numerical stencil is larger, thus requiring more storage for both **J** and **P**.
2. The matrix **J** is potentially less diagonally dominant, resulting in a less stable inversion , $\mathbf{P}^{-1}$.
3. The flux-limiter may be nondifferentiable, analytically.

Within the matrix-free algorithm storage is only required for **P**, not **J**, one is required to invert **P**, not **J**, and one only needs to evaluate some approximation to **J** to define **P**. Given this information, consider the option of forming the preconditioner using a first-order upwind convection scheme,

$$\mathbf{J}\mathbf{P}^{-1}\mathbf{v} \approx \frac{\mathbf{F}_{high}\left(\mathbf{x} + \epsilon \mathbf{P}_{low}^{-1}\mathbf{v}\right) - \mathbf{F}_{high}(\mathbf{x})}{\epsilon}. \tag{17}$$

Equation (17) is a right preconditioned version of Eq. (15). $\mathbf{F}_{high}(\mathbf{x})$ denotes the nonlinear function evaluated with a high-order, monotone, discretization, and $\mathbf{P}_{low}^{-1}$ denotes the inversion of a preconditioner formed with first-order upwinding. This option will result in less storage and a potentially more stable inversion, in contrast to

$$\mathbf{J}\mathbf{P}^{-1}\mathbf{v} \approx \frac{\mathbf{F}_{high}\left(\mathbf{x} + \epsilon \mathbf{P}_{high}^{-1}\mathbf{v}\right) - \mathbf{F}_{high}(\mathbf{x})}{\epsilon}, \tag{18}$$

where the preconditioner (**P**) is evaluated using the same discretization as the nonlinear residual ($\mathbf{F}(\mathbf{x})$). The question of interest is "does the mismatch between the Jacobian and preconditioner cause a significant increase in the number of Krylov iteration per Newton iteration?" Note that this approximation does not compromise the nonlinear convergence characteristics, although it may produce a deterioration in the linear convergence rate. This technique has proven successful for the compressible Euler equations [16], the incompressible Navier–Stokes equation [17], and combustion problems [18]. Both matrix-free implementations above, Eq. (17) and Eq. (18), involve a fully implicit treatment of the high-order differencing. An additional option is to consider a standard Newton–Krylov method applied in a defect-correction mode [19],

$$\mathbf{J}_{low}\delta\mathbf{x} = -\mathbf{F}_{high}(\mathbf{x}). \tag{19}$$

This mismatch between the Jacobian and the residual will result in degradation of the nonlinear convergence rate and possibly prevent nonlinear convergence. We will compare the performance of all of these options. The method of Eq. (17) will be refered to as $J_H P_L$, the method of Eq. (18) will be referred to as $J_H P_H$, and the method of Eq. (19) will be referred to as $J_L P_L$.

3. PHYSICS MODEL AND EQUATION SYSTEM

The tokamak edge plasma is made up of hydrogen ions, electrons, impurity ions from the vessel structure, such as carbon, as well as hydrogen and impurity neutral atoms and molecules. The system of equations solved in this paper only models hydrogen ions, atoms, and electrons. For more details regarding the equation system and boundary conditions (see [8]). The following equations are solved on a two-dimensional (x, y) Cartesian grid after they are first nondimensionalized to improve scaling:

Plasma continuity,

$$\frac{\partial n_i}{\partial t} + \nabla \cdot (n_i \mathbf{U}_i) = \nu_{ion} n_o - \nu_{rec} n_i; \tag{20}$$

Neutral continuity,

$$\frac{\partial n_o}{\partial t} + \nabla \cdot (n_o \mathbf{U}_o) = -\nu_{ion} n_o + \nu_{rec} n_i; \tag{21}$$

Ion parallel momentum,

$$\begin{aligned}\frac{\partial m n_i u_\parallel}{\partial t} + \nabla \cdot (m n_i \mathbf{U}_i u_\parallel - \tilde{\eta}_i \cdot \nabla u_\parallel) &= -\frac{B_x}{B}\left(\frac{\partial P_i}{\partial x} + \frac{\partial P_e}{\partial x}\right) \\ &+ \nu_{ion} m n_o u_{\parallel o} - \nu_{rec} m n_i u_\parallel + \nu_{i-n} m n_i (u_{\parallel o} - u_\parallel);\end{aligned} \tag{22}$$

Neutral vector momentum (three components),

$$\frac{\partial m n_i \mathbf{U}_o}{\partial t} + \nabla \cdot (m n_i \mathbf{U}_o \mathbf{U}_o) = \nabla \cdot \tau_{i,j} - \nu_{ion} m n_o \mathbf{U}_o + \nu_{rec} m n_i \mathbf{U}_i - \nu_{i-n} m n_i (\mathbf{U}_o - \mathbf{U}_i); \tag{23}$$

Ion + neutral internal energy,

$$\begin{aligned}\frac{\partial}{\partial t}\left(\frac{3}{2}(n_i + n_o) T_H\right) + \nabla \cdot \left(\frac{3}{2} T_H (n_i \mathbf{U}_i + n_o \mathbf{U}_o) - (\tilde{\kappa}_i + \tilde{I}\kappa_o) \cdot \nabla T_H\right) \\ = -(P_i \nabla \cdot \mathbf{U}_i + P_o \nabla \cdot \mathbf{U}_o) + \kappa_{eq}(T_e - T_H);\end{aligned} \tag{24}$$

Electron internal energy,

$$\begin{aligned}\frac{\partial}{\partial t}\left(\frac{3}{2} n_e T_e\right) + \nabla \cdot \left(\frac{3}{2} n_e \mathbf{U}_i T_e - \tilde{\kappa}_e \cdot \nabla T_e\right) \\ = -P_e \nabla \cdot \mathbf{U}_i - \kappa_{eq}(T_e - T_H) - I_p (n_o \nu_{ion} - n_i \nu_{rec}) - L_{rad}.\end{aligned} \tag{25}$$

Here, n_i is the ion number density, n_e is the electron number density (equal to n_i assuming quasi-neutrality), n_o is the neutral atom number density. $u_\parallel$ is the parallel ion velocity along the total magnetic field (B), with the parallel direction being a combination of the x and z directions, i.e., $\mathbf{B} = B_x \hat{x} + B_z \hat{z}$. u is the x-direction ion velocity which is equal to $(B_x/B) u_\parallel$, where B_x/B is the magnetic field pitch. v is the y-direction ion velocity which is obtained from a diffusive approximation, $nv = -D_\perp (\partial n/\partial y)$ [8]. w is the z-direction ion velocity which is equal to $(B_z/B) u_\parallel$. Thus, the Cartesian ion velocity is given by $\mathbf{U}_i = u\hat{x} + v\hat{y} + w\hat{z}$. Ambipolar flow is assumed so that the electron velocity is equal to the ion velocity. $\mathbf{U}_o$ is the neutral atom velocity, with $\mathbf{U}_o = u_o \hat{x} + v_o \hat{y} + w_o \hat{z}$. m is the ion/atom mass; T_H is the ion/atom temperature; T_e is the electron temperature. The equations of state give the following relations for pressure: $P_e = n_e T_e$; $P_i = n_i T_H$; and $P_o = n_o T_H$. ν_{ion} and ν_{rec} are the ionization and recombination frequencies, and ν_{i-n} is the ion–atom elastic collision frequency. η is the viscosity, κ is the thermal conductivity, $\tau_{i,j}$ is the neutral (atom) fluid stress tensor, and κ_{eq} is a thermal energy transfer coefficient. The viscosities and thermal conductivities for the plasma are modeled as diagonal tensors, i.e., $\kappa_x \neq \kappa_y$, due to the strong magnetic field. The parallel transport coefficients are classical Braginskii [20], and the radial transport coefficients are given by $D_\perp =$ input constant, $\kappa_{y,i} = n_i \chi_i$, $\kappa_{y,e} = n_e \chi_e$, with

χ_i, $\chi_e =$ input constants, $\eta_y = n_i m \eta_\perp$, $\eta_\perp =$ input constant. I_p is the ionization potential energy of a hydrogen atom (13.6 eV), and L_{rad} represents energy loss due to atomic line radiation. Because we solve for three components of neutral momentum there is a total of eight equations with the dependant variables being n_i, n_o, $u_\parallel$, u_o, v_o, w_o, T_H and T_e. These equations are strongly coupled through nonlinear source/sink terms which represent the effects of ionization, recombination, ion-neutral elastic collisions, and ion–electron coulomb collisions. The ionization and recombination rates are two-dimensional table lookup functions of plasma density and electron temperature with collisional-radiative corrections. The problem exhibits large spreads in time scales and space scales. For instance, the ratio between the electron thermal conduction time scale along x in the "hot" upstream region (fast time time scale) and plasma diffusion in the y-direction (slow time scale) can be as large as 1×10^6. The ratio between the poloidal length of the simulation region and the width of an ionization front [8] can be as large as 1×10^4.

4. COMPUTATIONAL RESULTS

Figure 3 presents the geometry for model problem 1. This is a simplified, Cartesian representation of the right half-plane of the poloidal cross section of a single null divertor. This simplified representation invokes many assumptions, including both toroidal and poloidal symmetry. As such, this geometry is an idealized representation of the transport of particles and energy from the core region to the divertor plate. The plasma density is fixed at the "core" boundary along with ion and electron energy flows. No mass or energy flow is allowed to exit the symmetry plane (left boundary) or the wall (top) boundary. Plasma energy can exit the problem at the divertor plate according to prescribed sheath boundary conditions [8], or through a volumetric loss term that represents atomic line radiation. All plasma that flows into the divertor plate is assumed to recombine and is redirected back into the problem as a hydrogen atom flux. The total poloidal (x) length is 1 m, 25 cm from the null point to the strike point (the intersection of the separatrix and the divertor plate). The overall radial (y) width is 5 cm, 4 cm from the separatrix to the outer wall. Figure 4 is the geometry for model problem 2. The same type of boundary conditions are used here. This geometry is larger with an expanded slot region and baffles in the slot to contain the neutral

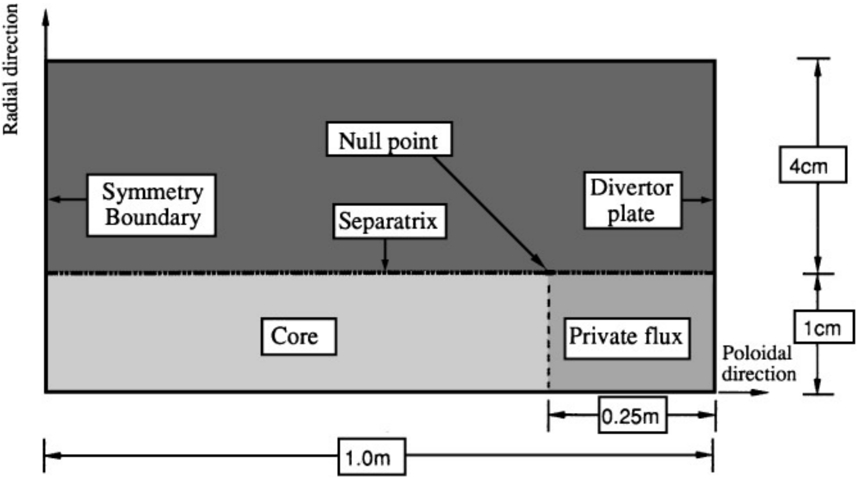

FIG. 3. Schematic representation of model problem 1 geometry.

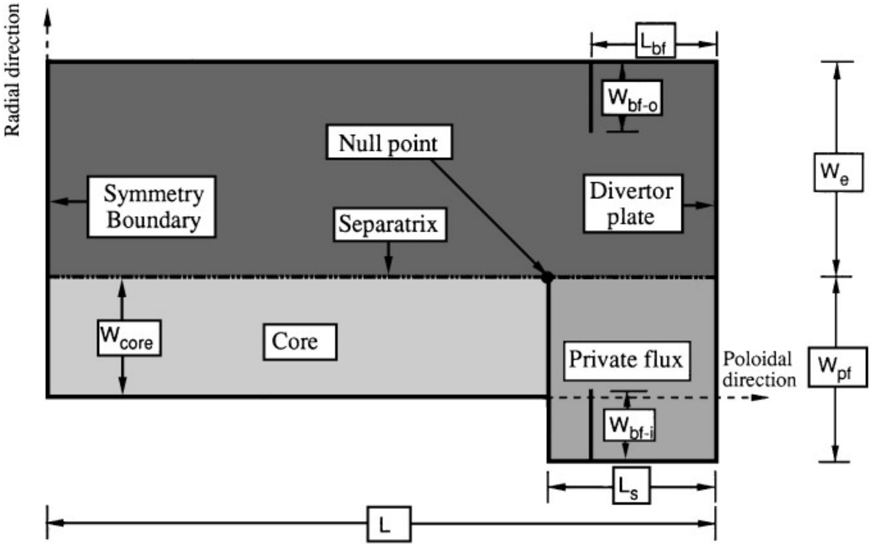

FIG. 4. Schematic representation of model problem 2 geometry.

flow. This is the type of geometry which has been used in Ref. [21]. The total poloidal length (L) is 2 m, 50 cm from the null point to the strike point (L_s). The overall radial width is 11 cm to the left of the null point $(w_{\text{core}} + w_e)$ and 20 cm to the right of the null point $(w_{pf} + w_e)$. Twenty-five centimeters separate the baffle location and the divertor plate in the poloidal direction (L_{bf}).

4.1. *Problem* 1

This problem has been studied extensively in Ref. [8]. Relatively speaking, it is a high upstream (core) density, low power problem on a small geometry. In this problem the core density and temperatures were fixed at $n = 1.5 \times 10^{20}$ m^3, $T_e = T_H = 40$ eV, and the radial transport coefficients were $D_\perp = 0.5$, $\chi_i = \chi_e = 0.5$, $\eta_\perp = 0.2$. We first consider the algorithmic performance on this problem, and then we examine the physical solution structure and the effect of improved differencing.

Table 1 presents memory requirements, iteration requirements, and CPU time for problem 1 on a 64×32 grid starting with a poor initial guess. The grid is uniform in the y direction and nonuniform in the x direction with 44 grid cells in the last 25 cm. The initial guess, while not a consistent solution to the equations, has properly directed gradients. In this problem, unless otherwise stated, a new Jacobian is formed only every 10 Newton iteration to evaluate a new

TABLE 1
Algorithm Performance Data for Problem 1, Starting from a Poor Initial Guess on a 64 × 32 Grid (ILU (1) Preconditioning, $\gamma_k = 5 \times 10^{-2}$, $\Delta t^0 = 1 \times 10^{-8}$)

Solution method	Jacobian+precond. memory (MWORDS)	Newton iterations	GMRES iterations	HP 735 CPU hours
$J_H P_L$	2.5	341	3432	5.9
$J_H P_H$	4.2	354	4479	10.2
$J_L P_L$ (DC, 10)	4.0	DIV.	DIV.	DIV.
$J_L P_L$ (DC, 2)	4.0	435	2579	22.2

TABLE 2

Algorithm Performance Data for Problem 1, Starting from a Good Initial Guess on a 64 × 32 Grid (ILU (1) Preconditioning, $\gamma_k = 5 \times 10^{-2}$, $\Delta t^0 = 1 \times 10^{-5}$)

Solution method	Jacobian+precond. memory (MWORDS)	Newton iterations	GMRES iterations	HP 735 CPU hours
$J_H P_L$	2.5	24	328	0.6
$J_H P_H$	4.2	39	708	1.7
$J_L P_L$ (DC, 10)	4.0	104	435	1.3

preconditioner [6, 7]. As stated, $J_H P_L$ and $J_H P_H$ do not require one to store J. The ILU (1) preconditioner (by itself) requires approximately 60% more memory than the Jacobian, J. While only $\|\mathbf{F}(\mathbf{x})\|_\infty$ will be plotted, both $\|\delta\mathbf{x}/\mathbf{x}\|_2$ and $\|\mathbf{F}(\mathbf{x})\|_\infty$ are used to declare solution convergence on a given grid. On this problem we required $\|\mathbf{F}(\mathbf{x})\|_\infty < 5.0 \times 10^{-7}$ and $\|\delta\mathbf{x}/\mathbf{x}\|_2 < 5.0 \times 10^{-4}$. We can see when comparing $J_H P_L$ and $J_H P_H$, the number of nonlinear iterations is almost identical, the number of Krylov iterations for $J_H P_H$ is 30% greater, and the CPU time for $J_H P_H$ is 73% greater.The growth in the number of Krylov iterations for $J_H P_H$ is somewhat surprising, since in this option the preconditioner is derived from the high-order discretization; i.e., it is more consistent with the Jacobian. Our interpretation of this is that P_{low} (first-order upwind) provides a more stable inversion and thus provides an improved preconditioner, even though it has a mismatch in convective discretization with J_{high}. This behavior is consistent with that observed on a combustion problem [18]. The use of defect correction ($J_L P_L$), with a new Jacobian formed every 10 nonlinear iterations (DC, 10), did not converge, and a new Jacobian formation every two nonlinear iterations was required to enable convergence. Thus, while only 34 Jacobian evaluation were required for $J_H P_H$, 218 Jacobian evaluations were required for $J_L P_L$ (DC, 2). This resulted in an increased CPU time of 275%.

Table 2 presents the same performance indicators as for Table 1 but with a much improved initial guess. The initial guess was an interpolation of a converged 32 × 16 grid solution. Due to this much improved initial guess, the initial pseudo-time step was three orders of magnitude larger. Again, in terms of both memory and CPU time, $J_H P_L$ is the most efficient. This time, with the much improved initial guess, $J_L P_L$ (DC, 10) was able to converge, although with four times as many Newton iterations and Jacobian evaluations. The maximum steady-state residual is plotted as a function of Newton iteration and CPU time in Fig. 5.

Tables 3 and 4 use $J_H P_L$ from Table 2 to compare the sensitivities of the inexact Newton parameter γ_k in Eq. (14), and the constant, a in Eq. (16), is used to evaluate the matrix-free

TABLE 3

Effect of γ on Algorithm Performance Data for Problem 1, Starting from a Good Initial Guess on a 64 × 32 Grid (ILU (1) Preconditioning, $J_H P_L$, $\gamma_k = 5 \times 10^{-2}$, $\Delta t^0 = 1 \times 10^{-5}$)

Solution method	Jacobian+precond. memory (MWORDS)	Newton iterations	GMRES iterations	HP 735 CPU hours
$\gamma_k = 2 \times 10^{-1}$	2.5	35	354	0.77
$\gamma_k = 5 \times 10^{-2}$	2.5	24	328	0.6
$\gamma_k = 1 \times 10^{-2}$	2.5	23	412	0.67

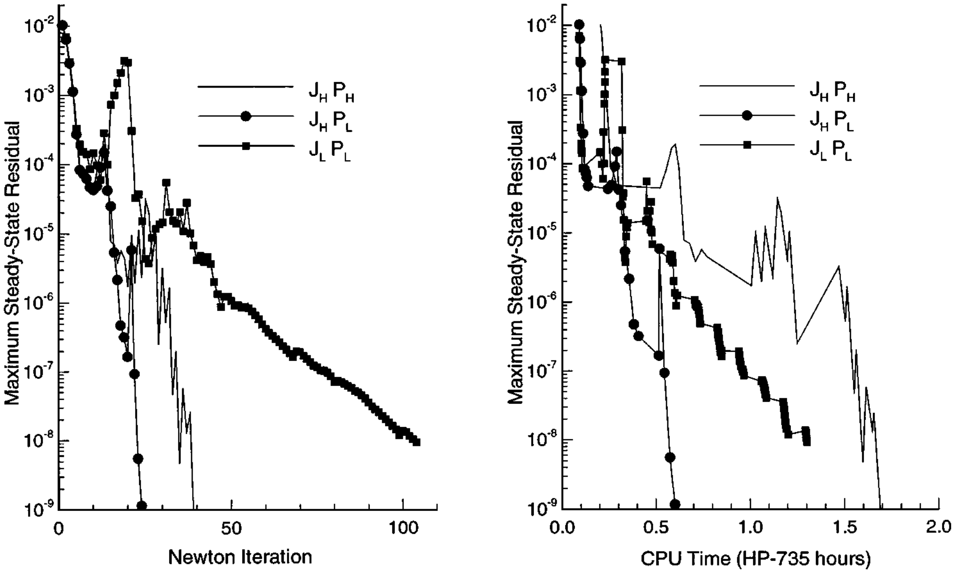

FIG. 5. Convergence history for problem 1 on a 64 × 32 grid. Initial guess was an interpolation of a 32 × 16 converged solution.

perturbation, ϵ. We can see that while there are effects changing these parameters, overall the effects do not appear to be large.

Figures 6 and 7 compare the difference between the first-order upwind solution and that obtained with the QUICK differencing and the SMART flux-limiter, both on a 128 × 64 gird. Figure 6 compares contour plots of $log(n_o)$ for the last 25 cm poloidally. Figure 7 compares electron temperature, T_e, and plasma density, n_i, along the separatrix for the last 25 cm poloidally. As seen in the contour plot, this solution is fairly one-dimensional with a nearly planar ionization front. The neutral density gradient does appear somewhat sharper with the improved convective differencing. In Fig. 7 the expected behavior of improved convective differencing is also observed. Both profiles of T_e and n_i appear sharper. The structure of a recombining divertor plasma is evident in both Figs. 6 and 7. This includes the low plasma temperature $\approx 1\,eV$, plasma density decreasing towards the plate due to recombination, and the neutral density being larger than the plasma density at the plate.

4.2. *Problem* 2

In this problem the scale lengths are somewhat larger. The divertor slot is wider and baffling has been added in order to help contain the neutral density in the divertor region

TABLE 4
Effect of ϵ on Algorithm Performance Data for Problem 1, Starting from a Good Initial Guess on a 64 × 32 Grid (ILU (1) Preconditioning, $J_H P_L$, $\Delta t^0 = 1 \times 10^{-5}$)

Solution method	Jacobian+precond. memory (MWORDS)	Newton iterations	GMRES iterations	HP 735 CPU hours
$a = 1 \times 10^{-4}$	2.5	27	378	0.65
$a = 1 \times 10^{-6}$	2.5	24	328	0.6
$a = 1 \times 10^{-8}$	2.5	28	367	0.65

TABLE 5

Algorithm Performance Data for Problem 2, Starting from a Converged Upwind Initial Guess on a 128 × 64 Grid (ILU (1) Preconditioning, $\gamma_k = 5 \times 10^{-2}$, $\Delta t^0 = 5 \times 10^{-6}$

Solution method	Jacobian+precond. memory (MWORDS)	Newton iterations	GMRES iterations	HP 735 CPU hours
$J_H P_L$	10.0	82	869	7.4
$J_H P_H$	16.8	78	1218	16.8
$J_L P_L$ (DC, 5)	16.0	118	396	10.9

[21]. Additionally, there is a lower upstream density and a higher upstream temperature (and pressure), as compared to the first problem. In this problem the core density and temperatures were fixed at $n = 5.0 \times 10^{19}\,\text{m}^3$, $T_e = T_H = 200\,eV$, and the radial transport coefficients were $D_\perp = 0.5$, $\chi_i = \chi_e = 0.5$, $\eta_\perp = 0.2$.

Table 5 presents convergence performance for this problem on a 128 × 64 grid, starting with a converged first-order upwind solution (on the same grid) as an initial guess. The grid is uniform in y. It has 88 uniform poloidal, (x), grid cells from the null point to the divertor plate, and 40 nonuniform poloidal grid cells from the null point to the symmetry plane. Convergence behavior, in terms of CPU time and number of iterations, is nearly identical when starting from an interpolation of a converged 64 × 32 grid. As with problem 1, we see that $J_H P_L$ outperforms $J_H P_H$ and $J_L P_L$ in both CPU time and memory requirements. Compared to Table 2, this problem was more difficult to converge in terms of Newton iterations. This is believed to be mostly a result of the increased structure in this problem as compared to problem 1. For this problem (DC, 10) did not converge, but (DC, 5) did. Figure 8 is a plot of maximum steady-state residual as a function of Newton iteration and CPU time. As compared to Fig. 5, we see a significant (factor of 3) increase in the number of Newton iterations. Part of this is due to the increased size of the grid (128 × 64 vs. 64 × 32). However, again a significant part of the increase is believed to be a result of the more complex solution structure of this problem. Large oscillations are observed in

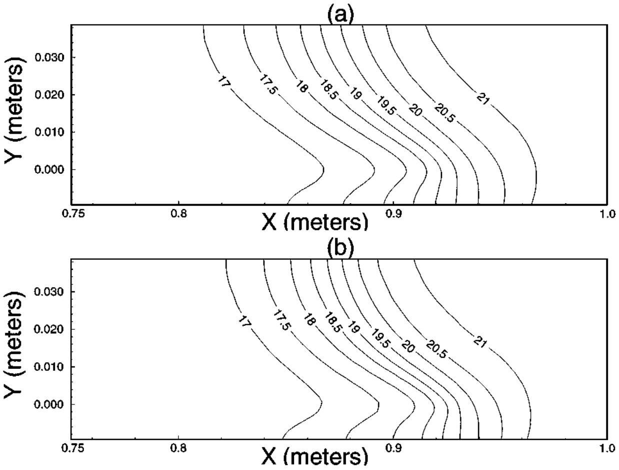

FIG. 6. Contour plots of log of neutral density $(1/m^3)$ contours for (a) first-order upwind solution and (b) higher-order solution.

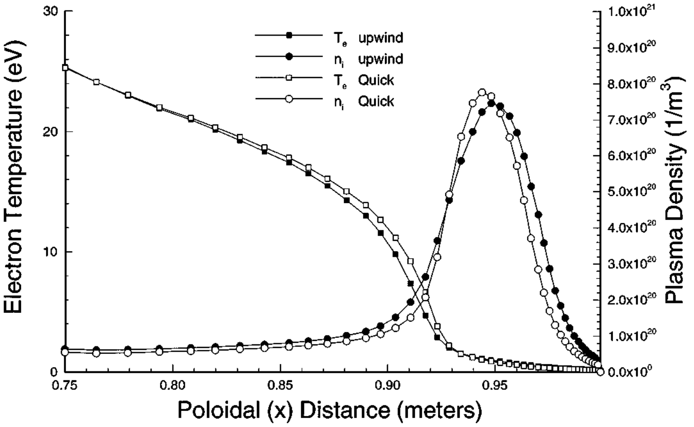

FIG. 7. Electron temperature and density variations along separatrix for problem 1.

Fig. 8 which do not appear in Fig. 5. These can be seen to occur with the frequency of re-evaluating J and more importantly Δt, the pseudo-time step. Here the change in Δt per Jacobian re-evaluation has been limited to a factor of 10. Again, while only $\|\mathbf{F}(\mathbf{x})\|_\infty$ is being plotted, both $\|\delta\mathbf{x}/\mathbf{x}\|_2$ and $\|\mathbf{F}(\mathbf{x})\|_\infty$ are used to declare solution convergence on a given grid. On this problem we required $\|\mathbf{F}(\mathbf{x})\|_\infty < 5.0 \times 10^{-7}$ and $\|\delta\mathbf{x}/\mathbf{x}\|_2 < 5.0 \times 10^{-4}$. The large oscillations in Fig. 8 can be traced back to the sharp ionization front [8] "settling in" on the grid.

The effect of the improved convective differencing on this problem is more pronounced. Figure 9 is a contour plot of neutral flux vectors overlayed on electron temperature contours (1–10 eV) and Fig. 10 is a contour plot of $log(n_o)$. Figure 11 is a line plot along the separatrix of electron temperature and plasma density, for the last 50 cm poloidally. These

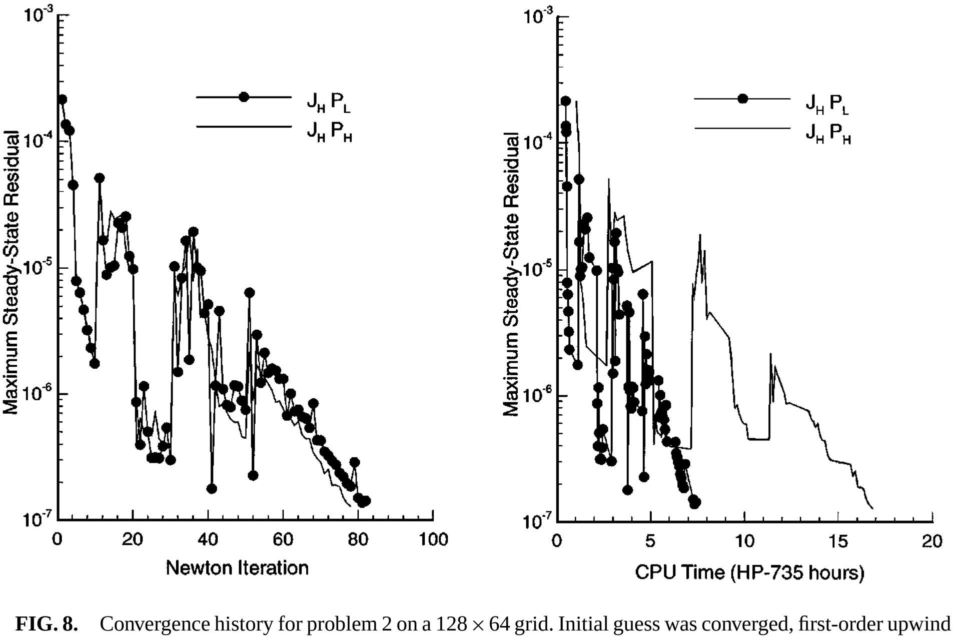

FIG. 8. Convergence history for problem 2 on a 128 × 64 grid. Initial guess was converged, first-order upwind solution on the same grid.

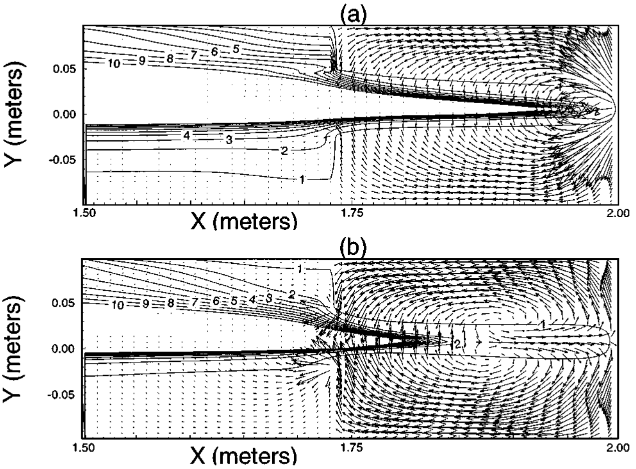

FIG. 9. Problem 2 contour plots of electron temperature (eV) contours superimposed upon neutral flux $(1/m^2s)$ vectors for (a) first-order upwind and (b) higher-order convective differencing.

results indicate that the improved accuracy in the convective differencing has enhanced the plasma recombination. The neutral density contour of 1.0×10^{20} m^{-3} has moved a significant distance, poloidally, as has the $2\,eV$ contour for T_e. Figure 11 displays a distinct difference in the gradients of plasma temperature and density along the separatrix. Defining the ionization front to be the $2\,eV$ contour of T_e, we can see this front has moved approximately 10 to 13 cm poloidally.

A common measure for tokamak divertor performance, both experimentally and theoretically, is to look at the peak heat flux and plasma particle flux on the divertor surface. The difference in the peak heat flux and plasma particle flux on the divertor surface between the two solutions (different convective discretization) is about a factor of 3. However, the more significant impact may be with self-consistent impurity transport, such as carbon [6], which is not considered in this study. The parallel (along B) transport of

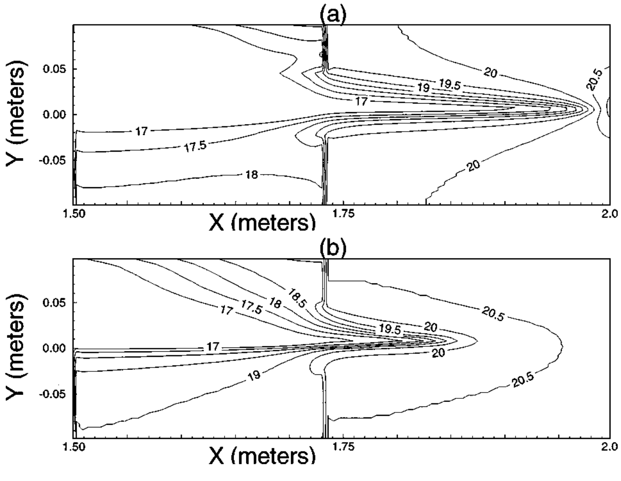

FIG. 10. Contour plots of log of neutral density $(1/m^3)$ for (a) first-order upwind and (b) higher-order convective differencing.

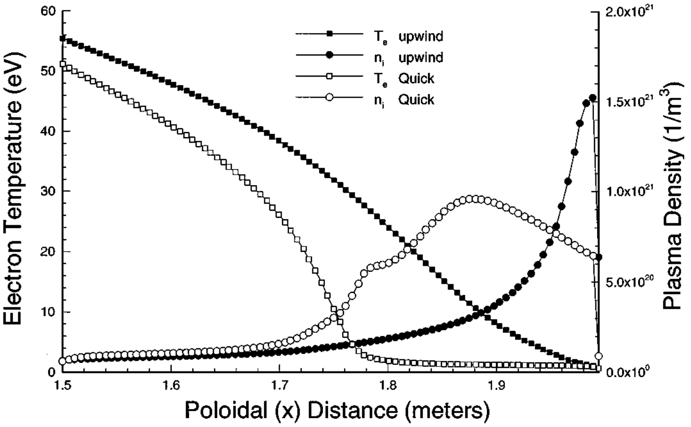

FIG. 11. Electron temperature and plasma density variations along separatrix for problem 2.

plasma impurity ions is a strong function of parallel temperature and density gradients, and these quantities are seen to change drastically as seen in Fig. 11. It has been noted previously that achieving high accuracy in the background (hydrogen) plasma solution may be paramount if one is interested in performing self-consistent impurity ion transport simulations [6, 22].

5. CONCLUSIONS

A high-order convection scheme, QUICK (quadratic upstream interpolation for convective kinematics) [9], along with a monotonicity preserving flux-limiter, SMART (sharp and monotonic algorithm for realistic transport) [10], has been applied to a combined system edge plasma/Navier–Stokes neutral transport equations [8] for modeling a recombining divertor plasma. It has been shown that the existing matrix-free Newton–Krylov algorithm provides a natural frame work for the fully implicit implementation of such discretization schemes. It has been demonstrated that using a preconditioner derived from a first-order upwind discretization of convection can reduce both required storage and CPU time.

A solution of current interest has been shown to be sensitive to the accuracy of the convective differencing, while another solution has been shown to be relatively insensitive. The two major differences between the two model problems are upstream plasma pressure and geometry. It was demonstrated in Ref. [21] that increasing the upstream plasma pressure leads to an increase in the neutral flow speeds in the divertor region, upon detachment. An increased importance of neutral convection in problem 2 (higher upstream pressure) could well explain the increased sensitivity to the improved convective differencing. Also, adding baffling to the geometry has been shown to increase the radial neutral density gradient in the divertor region. This sharp gradient will be more sensitive to the accuracy of the convective discretization employed. A more comprehensive study will be required to better characterize parameter regimes of sensitivity to this improved convective discretization.

ACKNOWLEDGMENTS

The author gratefully acknowledges the help of Dr. P. R. McHugh in the development of this computational capability. This work was performed under U.S. Department of Energy Contract W-7450-ENG-36 at Los Alamos National Laboratory.

REFERENCES

1. J. Wesson, *Tokamaks* (Clarendon Press, Oxford, 1987).
2. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow* (Hemisphere, New York, 1980).
3. B. J. Braams, *Computational Studies in Tokamak Equilibrium and Transport*, Ph.D. thesis, University of Utrecht, 1986.
4. T. D. Rognlien, J. L. Milovich, M. E. Rensink, and G. D. Porter, A fully implicit, time-dependent 2-d fluid code for modeling tokamak edge plasmas, *J. Nucl. Mat.* **196**–**198**, 347 (1992).
5. D. A. Knoll and P. R. McHugh, An inexact Newton algorithm for solving the tokamak edge plasma fluid equations on a multiply connected domain, *J. Comput. Phys.* **116**, 281 (1995).
6. D. A. Knoll and P. R. McHugh, Newton–Krylov methods applied to a system of convection-diffusion-reaction equations, *Comput. Phys. Commun.* **88**, 141 (1995).
7. D. A. Knoll and P. R. McHugh, Enhanced nonlinear iterative techniques applied to a nonequilibrium plasma flow, *SIAM J. Sci. Comput.* **19**, 291 (1998).
8. D. A. Knoll *et al.*, Simulation of dense recombining divertor plasmas with a Navier–Stokes neutral transport model, *Physics of Plasmas* **3**, 293 (1996).
9. B. P. Lenard, A stable and accurate convective modelling procedure based on quadratic upstream interpolation, *Comput. Meth. Appl. Mech. and Eng.* **19**, 59 (1979).
10. P. H. Gaskell and A. K. C. Lau, Curvature-compensated convective transport: SMART, a new boundedness-preserving transport algorithm, *Int. J. Num. Meth. Fluids* **8**, 617 (1988).
11. V. Venkatakrishnan, Convergence to steady state solutions of the Euler equations on unstructured grids with limiters, *J. Comput. Phys.* **118**, 120 (1995).
12. T. Hayase, J. A. C. Humphery, and R. Grief, A consistently formulated QUICK scheme for fast and stable convergence using finite-volume iterative calculation procedures, *J. Comput. Phys.* **98**, 108 (1992).
13. B. P. Lenard and J. E. Drummond, Why you should not use “hybrid,” “power-law,” or related exponential schemes for convective modelling—there are much better alternatives, *Int. J. Numer. Methods Fluids* **20**, 421 (1995).
14. Y. Saad and M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving non-symetric linear systems, *SIAM J. Sci. Stat. Comput.* **7**, 856 (1986).
15. P. N. Brown and Y. Saad, Hybrid Krylov methods for nonlinear systems of equations, *SIAM J. Sci. Stat. Comput.* **11**, 450 (1990).
16. D. E. Keyes, Aerodynamic applications of Newton–Krylov–Schwarz solvers, in *Proceedings of the 14th Conf. on Num. Meth. in Fluid Dynamics* (Springer-Verlag, Berlin, 1995), p. 1.
17. R. W. Johnson, P. R. McHugh, and D. A. Knoll, High-order scheme implimentation using Newton–Krylov solution methods, *Numerical Heat Transfer, Part B* **31**, 295 (1997).
18. P. R. McHugh, D. A. Knoll, and D. E. Keyes, Application of a Newton–Krylov–Schwarz algorithm to low Mach number compressible combustion, *AIAA J.* **36**, 290 (1998).
19. R. W. Johnson, P. R. McHugh, and D. A. Knoll, Defect correction with a fully coupled inexact Newton method, *Numerical Heat Transfer, Part B* **26**, 173 (1994).
20. S. I. Braginskii, Transport processes in a plasma, in *Reviews of Plasma Physics* (Consultants Bureau, New York, 1965).
21. D. A. Knoll, Computational study of ITER-like dissipative divertor plasmas in the collisional limit, *Nucl. Fusion* **38**(1), 133 (1998).
22. D. A. Knoll *et al.*, Fluid simulation of beryllium transport in the ITER gaseous divertor, *Contrib. Plasma Phys.* **34**, 386 (1994).