Princeton Plasma Physics Laboratory

Steady-state benchmarks of DK4D: A time-dependent, axisymmetric drift-kinetic equation solvera)

B. C. Lyons, S. C. Jardin, and J. J. Ramos

July 2015

PRINCETON
PLASMA PHYSICS
LABORATORY

Princeton Plasma Physics Laboratory
Report Disclaimers

PPPL Report Availability

Princeton Plasma Physics Laboratory:

http://www.pppl.gov/techreports.cfm

Office of Scientific and Technical Information (OSTI):

http://www.osti.gov/scitech/

Related Links:

U.S. Department of Energy

U.S. Department of Energy Office of Science

U.S. Department of Energy Office of Fusion Energy Sciences

Steady-state benchmarks of DK4D: a time-dependent, axisymmetric drift-kinetic equation solver

B.C. Lyons,[1] S.C. Jardin,[2] and J.J. Ramos[3]

[1) Princeton University, Princeton, New Jersey, 08544, USA

[2) Princeton Plasma Physics Laboratory, Princeton, New Jersey 08543-0451, USA

[3) Plasma Science and Fusion Center, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139-4307, USA

The DK4D code has been written to solve a set of time-dependent, axisymmetric, finite-Larmor-radius drift-kinetic equations (DKEs) for the non-Maxwellian part of the electron and ion distribution functions using the full, linearized Fokker-Planck-Landau collision operator. The plasma is assumed to be in the low- to finite-collisionality regime, as is found in the cores of modern and future magnetic confinement fusion experiments. Each DKE is formulated such that the perturbed distribution function carries no net density, parallel momentum, or kinetic energy. Rather, these quantities are contained within the background Maxwellians and would be evolved by an appropriate set of extended magnetohydrodynamic (MHD) equations. This formulation allows for straight-forward coupling of DK4D to existing extended MHD time evolution codes. DK4D uses a mix of implicit and explicit temporal representations and finite element and spectral spatial representations. These, along with other computational methods used, are discussed extensively. Steady-state benchmarks are then presented comparing the results of DK4D to expected analytic results at low collisionality, qualitatively, and to the Sauter analytic fits for the neoclassical conductivity and bootstrap current, quantitatively. These benchmarks confirm that DK4D is capable of solving for the correct, gyroaveraged distribution function in stationary magnetic equilibria. Furthermore, the results presented demonstrate how the exact drift-kinetic solution varies with collisionality as a function of the magnetic moment and the poloidal angle.

I. INTRODUCTION

There are many circumstances in which understanding and modeling the dynamics of high-temperature plasmas requires a comprehensive model that incorporates both magnetohydrodynamic (MHD) and neoclassical physics. For example, neoclassical tearing modes (NTMs) can occur when the temperature and pressure profiles flatten across a seed magnetic island in the core of a tokamak plasma, causing a reduction in the neoclassical bootstrap current within the island. Since the bootstrap current outside of the island remains unaffected, the resulting hole in the bootstrap current profile can lead to further growth in the island size[1,2]. If the island grows large enough to alter the macroscopic magnetic equilibrium, the plasma can become MHD unstable and confinement is lost. A study on the Joint European Torus[3] (JET) found that these NTMs are the single most common root cause of disruptions[4]. Avoidance of these modes is expected to place a severe limit on plasma β for ITER[5,6]. Thus, a good understanding of, and predictive capability for, NTMs are crucial to the success of the ITER campaign.

Edge-localized modes (ELMs) provide another example of MHD phenomena that are strongly affected by neoclassical physics. The stability of certain types of ELMs is widely believed to be determined by the peeling-ballooning mode which is driven by both the pressure and current profiles in the edge[7–11]. As the pressure gradients can be quite strong in H-mode pedestals, the bootstrap current can make up a large fraction of the total current density in the edge. Thus, an accurate model for the bootstrap current is important for simulating peeling-ballooning, and thus ELM, stability.

A realistic and accurate numerical study of NTMs and ELMs, in addition to several other plasma instabilities, such as sawtooth modes[12], must accurately account for both the kinetic trapped particle dynamics that produce the bootstrap current and the magnetohydrodynamic evolution of the plasma equilibrium due to changes in the current profile. Furthermore, because many MHD phenomena are inherently three-dimensional, the code must work for nonaxisymmetric magnetic geometries. While sophisticated, three-dimensional MHD time-evolution codes exist (e.g., NIMROD[13] and M3D-C^1 [14]), it is more difficult to find kinetic neoclassical codes that can be coupled to these MHD codes to enable such a study. The XGC0[15] and DKES[16] codes do solve for neoclassical physics in 3D toroidal geometries.

XGC0, however, is a PIC code that requires enormous computing power and is thus not well-suited for efficient coupling with an MHD code. DKES uses a variational method to calculate neoclassical quantities but uses the Lorentz collision operator, which includes only pitch angle scattering. While this is the dominant process for electron-ion collisions in low-collisionality plasmas (such as those found in the cores of reactor-grade, toroidally-confined plasmas that we would like to study), energy scattering is just as important for like-particle collisions[17,18]. As Belli and Candy have shown[19], the use of model collision operators to study the bootstrap current, even ones substantially more sophisticated than the Lorentz operator, can lead to errors of 5-10% compared to the full Fokker-Planck-Landau collision operator. Several other neoclassical codes are widely used in the community, including NCLASS[20], CQLP/CQL3D[21,22], and NEO[19,23]. The NCLASS code uses a truncated form of the Hirshman-Sigmar[24] moment expansion of the collision operator and evaluates just the corresponding lower moments of the distribution functions. The CQLP/CQL3D codes include the complete, nonlinear Fokker-Planck-Landau collision operator, while the NEO code[19] uses the complete, linearized Fokker-Planck-Landau collision operator. Both then solve for the full distribution functions. All of these codes, however, assume a 2D axisymmetric toroidal geometry, limiting their use in studies of three-dimensional MHD instabilities. Furthermore, none of these codes was designed to couple to a three-dimensional, time-dependent MHD code and could require significant modifications to ensure self-consistency.

We, therefore, have begun development of a new code that is applicable to the cores of high-temperature fusion plasmas, has the potential to study neoclassical dynamics in three spatial dimensions, and can self-consistently and efficiently couple to a magnetohydrodynamics solver. An appropriate analytic model that could form the basis of such a code has been developed in Refs. 17 and 18. It features a Chapman-Enskog-like[25], coupled system of ion and electron fluid and drift-kinetic (DKE) equations in general 3D real space and 2D (gyroaveraged) velocity space with linearized Fokker-Planck-Landau collision operators. The equations follow an asymptotic expansion in the small parameter $\delta \sim \rho_i/L \ll 1$, the ratio of the ion Larmor radius to the macroscopic scale length. Additional orderings of small mass ratio between electrons and ions, and of low collisionality relevant to the high temperature plasmas of interest to fusion, are assumed. The expansions are carried out consistently to the frequency scale where collisions begin to influence the dynamics.

In Ref. 26, we presented the Neoclassical Ion-Electron Solver (NIES) which solved a reduced, low-collisionality version of this drift-kinetic formulation. Successful benchmarks of the NIES results to the Sauter equations[27] indicate that this formulation is both correct and computationally tractable. This has given us confidence to continue our pursuit of a numerical solution to the full formulation of Refs. 17 and 18. In this paper, we present the DK4D code which solves the time-dependent, finite collisionality ion and electron drift-kinetic equations in axisymmetric magnetic geometries to first order in each species Larmor radius parameter.

In Section II, we describe the DKEs solved by DK4D and explain all assumptions that we have made. Section III explains the computational methods used to solve the DKEs, which are largely based on the successful methods used by the NIES code. Finally, we present results from DK4D in Section V, particularly qualitative comparisons to the NIES formalism at low collisionality, the behavior of the exact drift-kinetic solution as the collisionality is varied, and benchmarks of the steady-state solutions to the Sauter equations. Note that convergence studies and additional details for DK4D can be found in the lead author's Ph.D. thesis[28].

II. ANALYTIC MODEL

We want to solve the time-dependent, finite-collisionality DKEs of Refs. 17 and 18, restricted to first-order in the Larmor radius parameter and to axisymmetric geometries. We work in a five-dimensional phase space. The three-dimensional configuration space, $\mathbf{x}$, is represented in flux coordinates defined by a normalized flux variable $\tilde{\Phi}$, a poloidal angle θ, and the toroidal angle ζ. In such coordinates, the axisymmetric magnetic field is defined as

$$\mathbf{B} = \nabla\psi \times \nabla\zeta + I\nabla\zeta = \psi'\nabla\tilde{\Phi} \times \nabla\zeta + I\nabla\zeta, \tag{1}$$

where ψ is the poloidal flux divided by 2π, $\psi' = d\psi/d\tilde{\Phi}$, and $I = RB_\zeta$ for major radius R and toroidal magnetic field strength B_ζ. The three-dimensional velocity space variable is $\mathbf{w} = \mathbf{v} - \mathbf{u}_s$, referenced to the frame of the relevant species s's macroscopic flow velocity, $\mathbf{u}_s$. In the following discussion, the species that is being considered, and thus the definition of $\mathbf{w}$, will be obvious. The gyro-averaging of the kinetic equations that results in our drift-kinetic equations eliminates the need to consider the gyroangle α, allowing us to consider

a two-dimensional velocity space consisting of $w = |\mathbf{w}|$ and $y = \cos\chi$, where χ is the pitch angle.

We will be solving the DKEs for $\bar{f}_{NMs}$, the non-Maxwellian part of the gyroaveraged distribution function for species s. This is the difference between the full, gyroaveraged distribution function, $\bar{f}_s$, and an evolving Maxwellian, f_{Ms}, such that

$$\bar{f}_{NMs} = \bar{f}_s - f_{Ms} = \bar{f}_s - \frac{n}{(2\pi)^{3/2} v_{ths}^3} \exp\left(-\frac{w^2}{2v_{ths}^2}\right), \tag{2}$$

where n is the density and v_{ths} is the thermal velocity for particles with mass m_s and temperature $T_s = m_s v_{ths}^2$. In this Chapman-Enskog-like expansion[25], the density, momentum, and temperature are entirely contained within the Maxwellian part of the distribution function and are evolved by an appropriate set of MHD equations. Note that in this formulation, the plasma is taken to be quasineutral with only electrons of charge $-e$ and a single ion species of charge $+e$. Thus, we consider a single density $n = n_e = n_i$.

We make two other assumptions about our plasma configuration. Firstly, we take the density and temperatures to be flux functions, allowing us to drop terms proportional to parallel gradients of those quantities in both DKEs. Secondly, we take $T_e = T_i$ since we do not have any external heat sources in this formulation to drive a temperature imbalance, allowing us to drop terms proportional to $(T_e - T_i)$ in the electron DKE. While these assumptions simplify the DKEs somewhat, they do not alter the structure of the equation in any significant way. Importantly, dropping these terms preserves the property of this formulation that $\bar{f}_{NMs}$ carries no net density, momentum, or temperature. This provides self-consistency with the MHD equations by ensuring that these low-order moments are only present in the Maxwellian part of the distribution function. Furthermore, these assumptions could be relaxed by simply restoring the dropped terms, without requiring a major reformulation of the code.

Given all this, the axisymmetric drift-kinetic equation for species s, reduced from the

more general DKEs derived in Refs. 17 and 18, can be written as

$$
\begin{aligned}
\frac{\partial \bar{f}_{NMs}}{\partial t} &+ wy\mathbf{b} \cdot \nabla \bar{f}_{NMs} - \frac{1}{2}w\left(1 - y^2\right)\mathbf{b} \cdot \nabla \ln B \frac{\partial \bar{f}_{NMs}}{\partial y} = \langle C_{ss} + C_{ss'} \rangle_\alpha \\
&+ \left\{ \frac{wy}{nT_s}\left[\frac{2}{3}\mathbf{b} \cdot \nabla\left(p_{s\parallel} - p_{s\perp}\right) - \left(p_{s\parallel} - p_{s\perp}\right)\mathbf{b} \cdot \nabla \ln B - F_{s\parallel}^{coll}\right] \right. \\
&+ P_2(y)\frac{w^2}{3v_{ths}^2}\left(\nabla \cdot \mathbf{u}_s - 3\mathbf{b} \cdot [\mathbf{b} \cdot \nabla\mathbf{u}_s]\right) + \frac{1}{3nT_s}\left(\frac{w^2}{v_{ths}^2} - 3\right)\nabla \cdot (q_{s\parallel}\mathbf{b}) \\
&- \left. \frac{\varsigma\left(e_s\right)I}{3m_s\Omega_s}\left[\frac{1}{2}P_2(y)\frac{w^2}{v_{ths}^2}\left(\frac{w^2}{v_{ths}^2} - 5\right) + \frac{w^4}{v_{ths}^4} - 10\frac{w^2}{v_{ths}^2} + 15\right]\mathbf{b} \cdot \nabla \ln B \frac{dT_s}{d\psi}\right\} f_{Ms}, \quad (3)
\end{aligned}
$$

where $B = |\mathbf{B}|$, $\mathbf{b}$ is the unit vector in the direction of $\mathbf{B}$, e_s is the species' charge, $\varsigma\left(e_s\right)$ is the sign of the charge, $\Omega_s = eB/m_s$ is the gyrofrequency, and P_l is the lth Legendre polynomial. The gyroaveraged collision operator is taken in the linearized Fokker-Planck-Landau form

$$
\begin{aligned}
\langle C_{ss} + C_{ss'} \rangle_\alpha =& \nu_{Ds}(w)\mathcal{L}[\bar{f}_{NMs}] - \nu_s f_{Ms}\frac{v_{ths}}{v_{ths'}^2}\frac{\mathbf{b} \cdot \mathbf{J}}{e_s n}\xi_{s'}y \\
&+ \frac{\nu_s v_{ths}^3}{w^2}\frac{\partial}{\partial w}\left\{\xi_s\left[w\frac{\partial \bar{f}_{NMs}}{\partial w} + \frac{w^2}{v_{ths}^2}\bar{f}_{NMs}\right] + \xi_{s'}\left[w\frac{\partial \bar{f}_{NMs}}{\partial w} + \frac{m_s w^2}{m_{s'}v_{ths'}^2}\bar{f}_{NMs}\right]\right\} \\
&+ \frac{\nu_s v_{ths}}{n}f_{Ms}\left(4\pi v_{ths}^2\bar{f}_{NMs} - \Phi_s[\bar{f}_{NMs}] + \frac{w^2}{v_{ths}^2}\frac{\partial^2\Psi_s[\bar{f}_{NMs}]}{\partial w^2}\right). \quad (4)
\end{aligned}
$$

where $\mathbf{J}$ is the current density and the collision frequency is

$$
\nu_s = \frac{e^4 n \ln\Lambda_s}{4\pi\epsilon_0^2 m_s^2 v_{ths}^3}, \quad (5)
$$

where $\ln\Lambda_s$ is the Coulomb logarithm and ϵ_0 is the permittivity of free space. We have defined

$$
\nu_{Ds}(w) = \frac{\nu_s v_{ths}^3}{w^3}\left[\varphi_s - \xi_s + \varphi_{s'} - \xi_{s'}\right], \quad (6)
$$

while the error function is

$$
\varphi_s = \varphi\left(x = \frac{w}{v_{ths}}\right) = \frac{2}{\sqrt{2\pi}}\int_0^x dt \, \exp\left(-t^2/2\right), \quad (7)
$$

the Chandrasekhar function is

$$
\xi_s = \xi\left(x = \frac{w}{v_{ths}}\right) = \frac{1}{x^2}\left[\varphi(x) - \frac{2x}{\sqrt{2\pi}}\exp\left(-x^2/2\right)\right], \quad (8)
$$

and the Lorentz operator is

$$
\mathcal{L}[f] = \frac{1}{2}\frac{\partial}{\partial y}\left[\left(1 - y^2\right)\frac{\partial f}{\partial y}\right]. \quad (9)
$$

6

The Rosenbluth potentials, Φ_s and Ψ_s, found in Eq. 15 are defined by a set of Poisson equations:

$$\frac{d}{dw}\left(w^2\frac{\partial\Phi_s}{\partial w}\right) + \frac{\partial}{\partial y}\left[(1-y^2)\frac{\partial\Phi_s}{\partial y}\right] = -4\pi w^2 \bar{f}_{NMs} \qquad (10)$$

and

$$\frac{d}{dw}\left(w^2\frac{\partial\Psi_s}{\partial w}\right) + \frac{\partial}{\partial y}\left[(1-y^2)\frac{\partial\Psi_s}{\partial y}\right] = w^2\Phi_s. \qquad (11)$$

Importantly, these coupled equations contain no $\tilde{\Phi}$ derivatives of $\bar{f}_{NMs}$, Φ_s, or Ψ_s. Thus, we can solve the DKEs independently on each flux surface.

In addition to the differential terms on the left-hand side of Eq. 3 and contained in the collision operator, this drift-kinetic equation is driven by moments of $\bar{f}_{NMs}$. In particular, we define three gyrotropic moments: the pressure anisotropy

$$(p_{s\parallel} - p_{s\perp}) = 2\pi m_s \int_0^\infty dw\, w^4 \int_{-1}^1 dy\, P_2(y)\, \bar{f}_{NMs}, \qquad (12)$$

the parallel heat flux

$$q_{s\parallel} = \pi m_s \int_0^\infty dw\, w^5 \int_{-1}^1 dy\, y\, \bar{f}_{NMs}, \qquad (13)$$

and the parallel component of the collisional friction force

$$F_{s\parallel}^{coll} = -\frac{2m_s\nu_s}{3\sqrt{2\pi}e_s}\mathbf{b}\cdot\mathbf{J} - 2\pi m_s\nu_s v_{ths}^3 \int_0^\infty dw \int_{-1}^1 dy\, y\, \bar{f}_{NMs}, \qquad (14)$$

where $s' = i$ if $s = e$ and vice versa.

Note that in the ion drift-kinetic equation, we drop all terms involving cross-species collisional interaction, namely $\mathbf{F}_i^{coll}$ in Eq. 3 and all terms containing $s' = e$ in Eq. 4.

III. COMPUTATIONAL METHODS

A. Time advance and discretization

In order to solve for $\bar{f}_{NMs}$, we advance the DKE in time, simultaneously with the Poisson equations for the Rosenbluth potentials. To do this properly, we must consider the various timescales that are found within these equations. In particular, the left-hand side of Eq. 3 is characterized by a transit timescale, $\tau_{ts} = qR/v_{ths}$ for safety factor q, while the collision operator is characterized by a collisional timescale, $\tau_{cs} = \nu_s^{-1}$. The cores of modern, high-temperature magnetic confinement fusion experiments operate in low-collisionality regimes,

7

that is

$$\hat{\nu} = \nu_* \epsilon^{3/2} = \frac{\tau_{ts}}{\tau_{cs}} \ll 1, \tag{15}$$

where ϵ is the inverse aspect ratio, ν_* is the more conventional collisionality parameter used in neoclassical theory, and we have again assumed that $T_e = T_i$. Thus, there exists a large separation of timescales within the DKEs. Furthermore, we must consider the timescales found in the MHD equations that get coupled to the DKEs. Ideal MHD activity is characterized by the Alfven time, $\tau_A = a\sqrt{\mu_0 m_i n}/B$, where a is the minor radius and μ_0 is the permeability of free space, while transport phenomena evolve on the resistive timescale, $\tau_r = \mu_0 \sigma a^2$, where σ is the conductivity. While the Alfven time is typically within an order of magnitude of the transit time for these tokamaks, it is typically much shorter than the collision time. Furthermore, the resistive time is much longer than any of the other timescales, typically exceeding the ion collision time by five to six orders of magnitude.

As we can expect the drift-kinetic equations to evolve to steady-states within several collision times, we can use two different ways of solving the time-dependent DKEs, depending on the application. If we are considering only stationary equilibria or magnetic configurations evolving on the resistive timescale, then the DKEs can just be solved to steady-state for a given configuration. Alternatively, if we are concerned with changes to the configuration occurring closer to the Alfven timescale, then the DKE could be evolved for the length of a single MHD time step when coupled with an MHD time-evolution code. In M3D-C^1, for example, it is typical to take time steps on the order of ten or a hundred Alfven times. The DKE solve would have to be done in iteration with M3D-C^1 without the assumption that the distribution functions would evolve to a steady-state as the magnetic configuration evolves.

Furthermore, given the wide range in the timescales of interest, it is appropriate to consider implicit methods that would allow us to avoid any numerical stability limits on the time steps that we could take. Of main concern are the differential terms in the DKEs. One would expect the hyperbolic transit terms on the left-hand side of Eq. 3 to create a stability condition that would restrict time steps to something on the order of the grid spacing over the thermal speed if treated explicitly. Likewise, the collision operator contains both hyperbolic and parabolic terms that could limit the time step to the order of the collision time. As we may wish to consider timescales significantly longer than transit or collision times, we treat all of these differential terms in the DKEs implicitly. The other homogeneous

terms in the DKEs are all moments of $\bar{f}_{NMs}$. We can expect the velocity space integrals in these terms to smooth out any oscillations that might grow up and thus keep these terms from driving numerical instabilities. Furthermore, these integrals would result in relatively denser blocks in the matrix that we would need to invert if the moment terms were treated implicitly, which could cause an unreasonable slowdown of our code. Thus, we will choose to treat these moment terms explicitly.

Applying this scheme to Eq. 3, and noting that $\mathbf{b} \cdot \nabla = -\frac{\psi'}{\mathcal{J}B}\frac{\partial}{\partial\theta}$ where the Jacobian is $\mathcal{J} = \left[\nabla\tilde{\Phi} \times \nabla\theta \cdot \nabla\zeta\right]^{-1}$, we can write the time-discretized DKE for a time step Δt as

$$\frac{\bar{f}_{NMs}^{n+1}}{\Delta t} + \mathcal{H}\left[\bar{f}_{NMs}^{n+1}\right] = \frac{\bar{f}_{NMs}^{n}}{\Delta t} + \mathcal{M}\left[\bar{f}_{NMs}^{n}\right] + \mathcal{I}, \tag{16}$$

where the homogeneous differential terms treated at the implicit time step $n+1$ are

$$\mathcal{H}\left[\bar{f}_{NMs}^{n+1}\right] = -\frac{wy\psi'}{\mathcal{J}B}\frac{\partial\bar{f}_{NMs}^{n+1}}{\partial\theta} + \frac{w\left(1-y^2\right)\psi'}{2\mathcal{J}B^2}\frac{\partial B}{\partial\theta}\frac{\partial\bar{f}_{NMs}^{n+1}}{\partial y} - \hat{\mathcal{C}}_s\left[\bar{f}_{NMs}^{n+1}\right], \tag{17}$$

the homogeneous moment terms treated at the explicit time step n are

$$\mathcal{M}\left[\bar{f}_{NMs}^{n}\right] = -\frac{1}{3nT_s}\left(\frac{w^2}{v_{ths}^2} - 3\right)f_{Ms}\frac{\psi'}{\mathcal{J}}\frac{\partial}{\partial\theta}\left(\frac{q_{s\parallel}^n}{B}\right)$$
$$-\frac{wy}{nT_s}f_{Ms}\left\{\frac{2}{3}\frac{\psi'}{\mathcal{J}B}\frac{\partial}{\partial\theta}\left(p_{s\parallel} - p_{s\perp}\right)^n - \frac{\psi'}{\mathcal{J}B^2}\frac{\partial B}{\partial\theta}\left(p_{s\parallel} - p_{s\perp}\right)^n + \hat{F}_s^n\right\}, \tag{18}$$

and the inhomogeneous source terms are

$$\mathcal{I} = \left\{P_2(y)\frac{w^2}{3v_{ths}^2}\left(\nabla \cdot \mathbf{u}_s - 3\mathbf{b} \cdot \left[\mathbf{b} \cdot \nabla\mathbf{u}_s\right]\right) - \nu_s\frac{v_{ths}}{v_{ths'}^2}\frac{J_\parallel}{e_s n}\xi_{s'}y + \frac{2}{3\sqrt{2\pi}}\nu_s\frac{w}{v_{ths}^2}\frac{J_\parallel}{e_s n}y\right\}f_{Ms}$$
$$+\frac{\varsigma(e_s)}{3m_s\Omega_s}f_{Ms}\left[\frac{1}{2}P_2(y)\frac{w^2}{v_{ths}^2}\left(\frac{w^2}{v_{ths}^2} - 5\right) + \frac{w^4}{v_{ths}^4} - 10\frac{w^2}{v_{ths}^2} + 15\right]\frac{I}{\mathcal{J}B^2}\frac{\partial B}{\partial\theta}\frac{dT_s}{d\tilde{\Phi}}. \tag{19}$$

Here, we've defined the homogeneous, gyroaveraged collision operator

$$\hat{\mathcal{C}}_s\left[\bar{f}_{NMs}\right] = \langle C_{ss} + C_{ss'}\rangle + \nu_s f_{Ms}\frac{v_{ths}}{v_{ths'}^2}\frac{J_\parallel}{e_s n}\xi_{s'}y, \tag{20}$$

with parallel current density $J_\parallel = \mathbf{b} \cdot \mathbf{J}$, and the homogeneous moment component of the parallel collisional friction force, $F_{s\parallel}^{coll} = \mathbf{b} \cdot \mathbf{F}_s^{coll}$,

$$\hat{F}_s = F_{s\parallel}^{coll} + \frac{2m_s\nu_s}{3\sqrt{2\pi}e_s}J_\parallel = -2\pi m_s\nu_s v_{ths}^3\int_0^\infty dw \int_{-1}^1 dy\, y\bar{f}_{NMs}. \tag{21}$$

Note that the inhomogeneous parts proportional to $J_\parallel$ that have been taken out of Eqs. 17 and 18 are put back into Eq. 19 without modification, such that this time-discretized DKE contains all the same terms as Eq. 3.

Finally, we note that the Poisson equations that define the Rosenbluth potentials, Eqs. 10 and 11, are time-independent and do not need to be time discretized. Rather, these elliptic equations are solved simultaneously with the DKE at each time step, giving self-consistent solutions for all times.

B. Spatial expansions

DK4D uses a finite element expansion in the speed variable, w, a Legendre polynomial expansion in the cosine of the pitch angle, $y = \cos\chi$, and a Fourier series in the poloidal spatial angle, θ. At time step n, we define

$$
\begin{pmatrix} \bar{f}^n_{NMs}\left(\tilde{\Phi}, \theta, w, y\right) \\ \Phi^n_s\left(\tilde{\Phi}, \theta, w, y\right) \\ \Psi^n_s\left(\tilde{\Phi}, \theta, w, y\right) \end{pmatrix} = \sum_{i=I_{min}}^{I_{max}} \sum_{l=0}^{L} \sum_{m=0}^{2M} \begin{pmatrix} \bar{f}^{i,l,m,n}_{NMs}\left(\tilde{\Phi}\right) \\ \Phi^{i,l,m,n}_s\left(\tilde{\Phi}\right) \\ \Psi^{i,l,m,n}_s\left(\tilde{\Phi}\right) \end{pmatrix} \phi_i(w) P_l(y) \mathcal{F}_m(\theta). \tag{22}
$$

There is no need for an expansion in $\tilde{\Phi}$ because the dynamical equations lack any derivatives of $\bar{f}_{NMs}$, Φ_s, and Ψ_s with respect to the flux coordinate and, thus, can be solved independently on each flux surface. For simplicity, we will drop the $\tilde{\Phi}$ dependence of the coefficients $\bar{f}^{i,l,m,n}_{NMs}$, $\Phi^{i,l,m,n}_s$, and $\Psi^{i,l,m,n}_s$ from this point forward.

In general, DK4D assumes an arbitrary finite element expansion, $\phi_i(w)$. The full range of w is divided into $N+1$ arbitrarily spaced grid points with $w_0 = 0$ and $w_N = w_{max}$. This maximum value of w is typically around $10 v_{ths}$ and can be varied to ensure convergence. The finite elements range from $i = I_{min} \leq 0$ to $I_{max} \geq N$, where the minimum and maximum value of i varies with the type of element considered. We currently have two elements implemented into DK4D, cubic B-splines and Hermite cubics, though more could easily be added if desired. A discussion of these elements can be found in Ref. 29. In particular, we note that the use of Hermite cubic finite elements has produced favorable results in other applications and appears to offer an efficient and general method for representing a wide class of functions[14,29–32].

We consider the full range of l in the Legendre polynomials and both sines and cosines in the Fourier series, where we've defined

$$
\mathcal{F}_m(\theta) = \begin{cases} \cos\left(\frac{m}{2}\theta\right) & : m \text{ even} \\ \sin\left(\frac{m+1}{2}\theta\right) & : m \text{ odd}. \end{cases} \tag{23}
$$

Both the Legendre and Fourier expansions are truncated to a maximum mode number (L and M respectively), which can be varied to ensure convergence of the solution.

C. Galerkin method

We use the Galerkin method in DK4D for both the finite element expansion in w and the spectral expansions in y and θ. To do this, we insert the expansion from Eq. 22 into the drift-kinetic equation (Eq. 16) and the Poisson equations for the Rosenbluth potentials (Eqs. 10 and 11). Then, we multiply these equations by $\phi_j(w)P_q(y)\mathcal{F}_r(\theta)$ for all possible combinations of $j = [I_{min}, I_{max}]$, $q = [0, L]$, and $r = [0, 2M]$. To avoid singularities in the collision operator at $w = 0$, we multiply the DKE by w^2 as well. We then integrate these equations over $w = [0, w_{max}]$, $y = [-1, 1]$, and $\theta = [0, 2\pi]$ to obtain the weak form of the equations.

Given this procedure, the homogeneous differential part of the DKE, Eq. 17, along with the $\bar{f}_{NMs}^{n=1}/\Delta t$ term in Eq. 16, and the two Poisson equations can be written as a matrix $\mathbf{A}$ acting on a vector $\mathbf{X}$ that contains all the $\bar{f}_{NMs}^{i,l,m,n+1}$, $\Phi_s^{i,l,m,n+1}$, and $\Psi_s^{i,l,m,n+1}$ coefficients. The chosen expansions result in an $\mathbf{A}$ that is tridiagonal in the l (Legendre) index and sparse banded in the i (w finite element) index. In particular, both the cubic B-spline and the Hermite cubic elements currently implemented in DK4D result in a block septadiagonal matrix in i. While, in general, the Fourier inner products create dense blocks in the m index, the matrix is still quite sparse due to the Legendre and finite element representations.

Applying the Galerkin method to the explicit moment terms and to the inhomogeneous source terms, on the other hand, results in a source vector $\mathbf{S}$. If treated implicitly, the moment terms would have added blocks to $\mathbf{A}$ that were dense in both m and i, since they would involve the exterior product of two separate integrals over θ and w. Thus, as noted previously, these terms would result in a denser matrix than the differential terms, which motivated our decision to treat these integral terms explicitly. Nevertheless, these terms would still be sparse in the Legendre expansion, leaving open the possibility that a fully implicit DKE solver could be tried in the future and remain computationally tractable. Furthermore, note that Eq. 19 is independent of time (for a fixed magnetic configuration) and would only need to be updated after a time-advancement of the MHD equations resulting in

11

a change to the magnetic configuration. Thus, this source can be calculated at the beginning of DK4D. The homogeneous source terms proportional to $\bar{f}^n_{NMs}$ can then be added to it each DK4D time step to form the complete source vector for a given time step. Ultimately, the Galerkin method transforms the coupled DKE and Poisson equation system into a linear matrix equation $\mathbf{AX} = \mathbf{S}$ to be solved numerically at each DKE time step.

D. Boundary conditions

We must impose appropriate boundary conditions on our DKE and Poisson equations for the Rosenbluth potentials given the spatial expansions we have chosen. The Fourier expansion already enforces the condition that the solutions be periodic in θ. In addition, the use of spectral elements in y means that we need no boundary condition in that dimension either. Thus, we only need boundary conditions for $w = 0$ and $w = w_{max}$.

As $\bar{f}_{NMs}$ is driven by terms proportional to a Maxwellian, we can expect the non-Maxwellian part of the distribution function to decay like a Maxwellian, or a polynomial in w times a Maxwellian, for $w > v_{ths}$. Given this, we impose a mixed boundary condition enforcing this decay at $w = w_{max}$:

$$\frac{\partial \bar{f}_{NMs}}{\partial w}(w = w_{max}) = -\frac{w}{v_{ths}^2} \bar{f}_{NMs}(w = w_{max}). \tag{24}$$

This is exact for a Maxwellian decay and the dominant decay for a polynomial times a Maxwellian. For the Rosenbluth potentials, we can assume that $\bar{f}_{NMs}(w = w_{max}) = 0$ for sufficiently high $w_{max} > v_{ths}$. Eq. 10 is then homogeneous near $w = w_{max}$ and has a regular solution that goes like w^{-l-1}. This then drives Eq. 11, which has a regular solution that goes like w^{-l+1}. Thus, the boundary conditions are

$$\frac{d\Phi_s^{l,m}}{dw}(w = w_{max}) = -\frac{l+1}{w_{max}} \Phi_s^{l,m}(w = w_{max}) \tag{25}$$

and

$$\frac{d\Psi_s^{l,m}}{dw}(w = w_{max}) = -\frac{l-1}{w_{max}} \Psi_s^{l,m}(w = w_{max}). \tag{26}$$

Including the non-zero, Maxwellian decay of $\bar{f}_{NMs}$ as a drive for Φ_s would only produce particular solutions for the Rosenbluth potentials that also decay like a Maxwellian and would thus be subdominant to these power law decays.

12

As $w = 0$ is not truly a boundary in phase space but just a degenerate point in our cylindrical velocity space, we only need to ensure regularity at that point. The simplest way to find this condition is to let $\bar{f}_{NMs} = w^\alpha P_l(y)\mathcal{F}_m(\theta)$, $\Phi_s = w^\beta P_l(y)\mathcal{F}_m(\theta)$, and $\Psi_s = w^\gamma P_l(y)\mathcal{F}_m(\theta)$. Plugging these into the DKE (Eq. 3) and the Poisson equations (Eqs. 10 and 11), one can find a consistent, dominant, regular solution at near $w = 0$ with $\alpha = \beta = \gamma = l$ for all values of l and m. Thus, the boundary condition at $w = 0$ is

$$\frac{d}{dw}\begin{pmatrix} \bar{f}_{NMs}^{l,m}(w = 0) \\ \Phi_s^{l,m}(w = 0) \\ \Psi_s^{l,m}(w = 0) \end{pmatrix} = 0 \quad \text{if} \quad l = 0 \tag{27}$$

$$\begin{pmatrix} \bar{f}_{NMs}^{l,m}(w = 0) \\ \Phi_s^{l,m}(w = 0) \\ \Psi_s^{l,m}(w = 0) \end{pmatrix} = 0 \quad \text{if} \quad l \neq 0. \tag{28}$$

E.   Solution method

The linear matrix equation $\mathbf{AX} = \mathbf{S}$ must be solved on each individual flux surface for each time step. As $\mathbf{X}$ is a vector containing the implicit expansion coefficients for all values of i, l, and m, it is of length

$$N_{tot} = 3N_{FE}(L + 1)(2M + 1), \tag{29}$$

where $N_{FE} = (I_{max} - I_{min} + 1)$ is the total number of finite elements. The vector $\mathbf{S}$ is, of course, also of length N_{tot}. $\mathbf{A}$ is a $N_{tot} \times N_{tot}$ matrix containing all the coupling between the coefficients at the implicit time step. Given the sparse structure of this matrix, it is reasonable to use a direct method to solve the $\mathbf{AX} = \mathbf{S}$ system with LU decomposition and back substitution.

As the DKEs are solved independently on each flux surface, we have parallelized DK4D perfectly over the number of flux surfaces that we wish to consider. In addition, DK4D uses the PETSc (Portable, Extensible Toolkit for Scientific Computation)[33–35] library in order to solve each flux surface's matrix equation. This flexible framework allows for the use of a variety of different parallel direct and iterative solvers. Typically, we run PETSc with MUMPS (MUltifrontal Massively Parallel Solver)[36,37] or SuperLU_DIST (Supernodal

13

LU Distributed)[38–40] which are parallel direct solvers for sparse systems. Both of these solvers are able to make full use of the sparsity of the matrix $\mathbf{A}$ to solve the linear system efficiently, both in terms of computational time and memory use. Furthermore, PETSc automatically takes care of the parallelization for a single matrix equation solve, both in terms of distributing the matrix and vectors across the processors within a communicator and the parallel LU decomposition and linear solve of the $\mathbf{AX} = \mathbf{S}$ system.

IV. PARALLEL HEAT FLUX MODIFICATION

Experience has shown us that one of the most difficult terms in Eq. 3 to deal with computationally is the term proportional to the divergence of the parallel heat flux. In the first-order formulation reduced from Refs. 17 and 18, the MHD temperature equation for species s can be written as

$$\frac{3n}{2}\frac{\partial T_s}{\partial t} + \frac{3n}{2}\mathbf{u}_s \cdot \nabla T_s + nT_s \nabla \cdot \mathbf{u}_s + \nabla \cdot \left(q_{s\parallel}\mathbf{b} + \frac{5nT_s\varsigma\,(e_s)}{2m_s\Omega_s}\mathbf{b} \times \nabla T_s \right) = 0, \qquad (30)$$

where, again, we've assumed that $T_e = T_i$. From this, we can see that if

$$\nabla \cdot \left(q_{s\parallel}\mathbf{b} \right) \neq -\nabla \cdot \left(\frac{5nT_s\varsigma\,(e_s)}{2m_s\Omega_s}\mathbf{b} \times \nabla T_s \right), \qquad (31)$$

that is, if the parallel heat flux does contain the appropriate Pfirsch-Schluter-like[41,42] return component necessary to cancel the divergence of the diamagnetic heat flux, then the background temperature will evolve in time. The fact that the temperature will develop a poloidal variation, if the inequality of Eq. 31 varies poloidally, is of particular difficulty, since this is something which we have excluded from the present model. In general, as Eq. 3 is advanced in time, there is no way to ensure that the divergence of the heat flux remains zero. Thus, a proper numerical treatment of the DKE would require iterating with an appropriate temperature equation, such as Eq. 30, in order to self-consistently establish the Pfirsh-Schluter-like return heat flux. Physical intuition tells us that any poloidal variation in T_s should drive an opposing parallel heat flux that would drive the temperature back to a flux surface quantity. Thus, while in steady-state we expect the temperature to be a flux function, it may require some poloidally-varying dynamics to arrive at that state. Such complex dynamics and such a tight coupling between the MHD temperature equations and

14

the drift-kinetic equations are beyond the scope of the present work, though they remain an area of future interest.

In order to avoid this complication, we can modify our DKEs to enforce that the divergence of the total heat flux remains zero by replacing the moment version of $q_{s\parallel}$, Eq. 13, with its expected Pfirsch-Schluter-like return value at all times. As we only need its divergence for the DKE, this amounts to using

$$\nabla \cdot \left(q_{s\parallel} \mathbf{b} \right) = -\nabla \cdot \left(\frac{5nT_s \varsigma\left(e_s \right)}{2m_s \Omega_s} \mathbf{b} \times \nabla T_s \right), \tag{32}$$

or

$$\frac{\psi'}{\mathcal{J}} \frac{\partial}{\partial \theta} \left(\frac{q_{s\parallel}^n}{B} \right) = -\frac{5nT_s \varsigma\left(e_s \right)}{m_s} \frac{I}{\mathcal{J}B^2} \frac{\partial B}{\partial \theta} \frac{dT_s}{d\tilde{\Phi}}. \tag{33}$$

Inserting Eq. 33 into the explicit moment term of Eq. 18, and transferring this new inhomogeneous term to Eq. 19, we find that the moment terms in the DKE are

$$\mathcal{M}\left[\bar{f}_{NMs}^n \right] = -\frac{wy}{nT_s} f_{Ms} \left\{ \frac{2}{3} \frac{\psi'}{\mathcal{J}B} \frac{\partial}{\partial \theta} \left(p_{s\parallel} - p_{s\perp} \right)^n - \frac{\psi'}{\mathcal{J}B^2} \frac{\partial B}{\partial \theta} \left(p_{s\parallel} - p_{s\perp} \right)^n + \hat{F}_s^n \right\}, \tag{34}$$

and the inhomogeneous drive for the DKE becomes

$$\mathcal{I} = \left\{ P_2(y) \frac{w^2}{3v_{ths}^2} \left(\nabla \cdot \mathbf{u}_s - 3\mathbf{b} \cdot \left[\mathbf{b} \cdot \nabla \mathbf{u}_s \right] \right) - \nu_s \frac{v_{ths}}{v_{ths'}^2} \frac{J_{\parallel}}{e_s n} \xi_{s'} y + \frac{2}{3\sqrt{2\pi}} \nu_s \frac{w}{v_{ths}^2} \frac{J_{\parallel}}{e_s n} y \right\} f_{Ms}$$
$$+ \frac{\varsigma(e_s)}{6m_s \Omega_s} f_{Ms} \left[2 + P_2(y) \right] \frac{w^2}{v_{ths}^2} \left(\frac{w^2}{v_{ths}^2} - 5 \right) \frac{I}{\mathcal{J}B^2} \frac{\partial B}{\partial \theta} \frac{dT_s}{d\tilde{\Phi}}. \tag{35}$$

In DK4D, we have implemented both Eqs. 18 & 19 and Eqs. 34 & 35 as the explicit, homogeneous moment and inhomogeneous drive terms. The former set will be used in future applications that are tightly coupled to the MHD temperature evolution equation. The latter set will be used in applications where the temperature profile remains stationary. This still allows us to calculate the distribution function in evolving magnetic configurations self-consistently with time-dependent electric fields, plasma currents, and density gradients. DK4D, using Eqs. 34 and 35, is also able to calculate the steady-state distribution function for magnetic equilibria, as will be considered in Section V.

V. RESULTS

While DK4D is designed to be coupled self-consistently with an MHD time-evolution code, it is prudent to first test it as a stand-alone application. Thus, as an initial test,
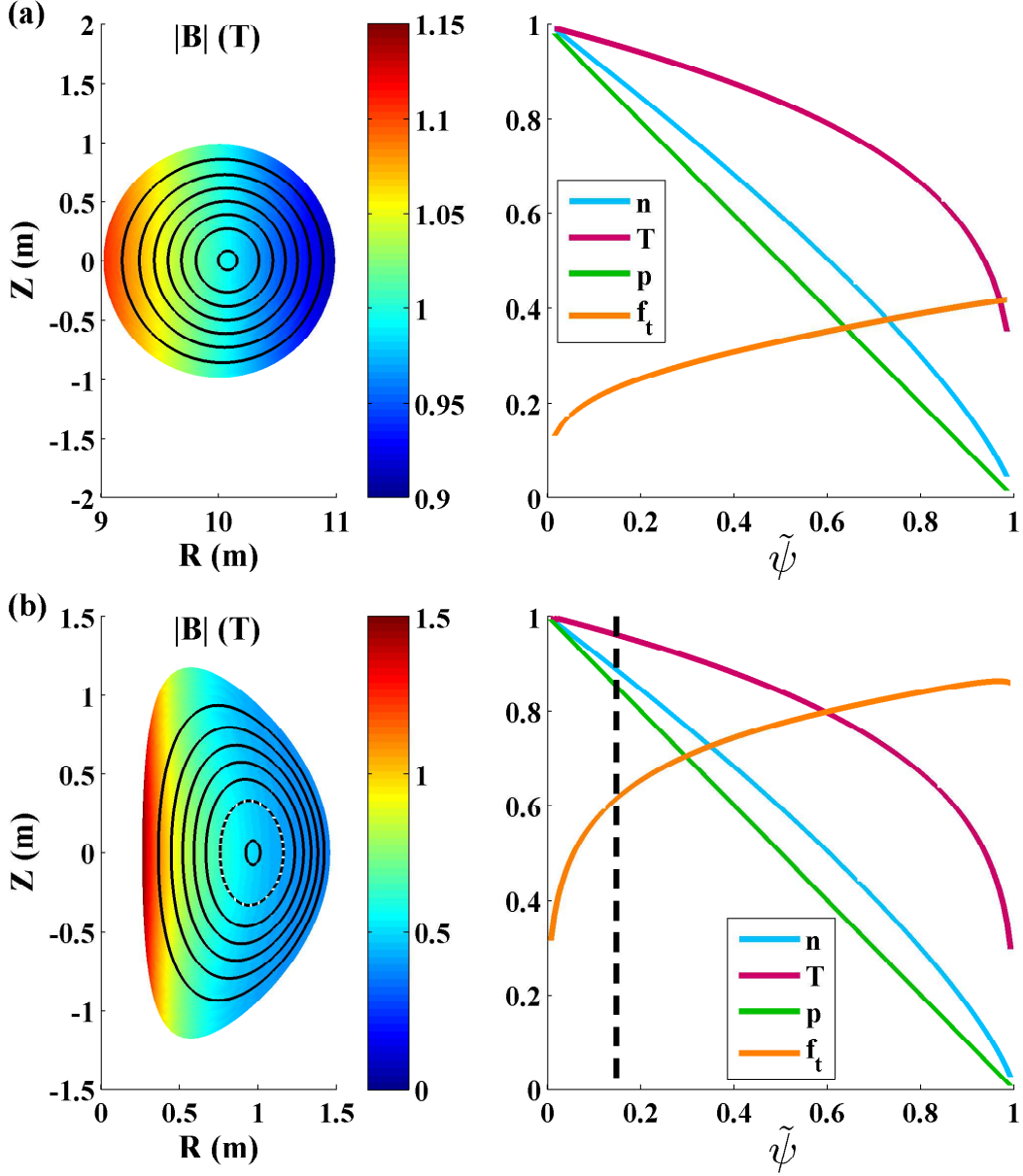
Figure 1. Left: The magnetic field strength in tesla in (a) a large aspect ratio JSOLVER equilibrium and (b) an NSTX JSOLVER equilibrium along with some corresponding flux surfaces. Right: The normalized density, temperature ($T = T_e = T_i$), and pressure, as well as the trapped particle fraction, f_t, for these equilibria. In (a), the pressure is normalized to $p_0 \sim 8.0 \times 10^2$ Pa, while in (b), it is normalized to $p_0 \sim 2.3 \times 10^4$ Pa. The normalizing density and temperature can be varied to explore various collisionality regimes. In (b), the dashed flux surface on the left and the dashed line on the right at $\tilde{\Phi} \approx 0.148$ show the location used for the study in Section V A.

we have used DK4D to solve for the steady-state distribution functions in two JSOLVER[43] Grad-Shafranov equilibria, namely a large aspect ratio equilibrium (Figure 1a) and a realistic NSTX equilibrium (Figure 1b). Note that these two equilibria were previously studied using DK4D's predecessor, NIES[26]. In order to explore various collisionalities, we have varied the normalizing values of the densities and temperatures, while keeping a fixed normalizing pressure and fixed normalized density and temperature profiles, such that the equilibrium itself remains unchanged.

When considering a stationary magnetic equilibrium, we can simplify the flow and current drive terms in Eq. 19. In particular, we can say that the mean flow for species s is

$$\mathbf{u}_s = U_s(\tilde{\Phi})\mathbf{B} + \frac{R^2}{\psi'}\left[\frac{dV}{d\tilde{\Phi}} + \frac{1}{e_s n}\frac{d(nT_s)}{d\tilde{\Phi}}\right]\nabla\zeta, \tag{36}$$

where U_s is the parallel flow stream function and V is the single-valued electric potential, such that

$$\frac{1}{\psi'}\frac{dI}{d\tilde{\Phi}} = \mu_0 en(U_i - U_e). \tag{37}$$

Then, the parallel current density is given by

$$J_\parallel = en(U_i - U_e)B + \frac{I}{\psi'B}\frac{dp}{d\tilde{\Phi}}, \tag{38}$$

while

$$\nabla \cdot \mathbf{u}_s = 0 \qquad \text{and} \qquad \mathbf{b} \cdot [\mathbf{b} \cdot \nabla \mathbf{u}_s] = U_s\mathbf{b} \cdot \nabla B. \tag{39}$$

Given these definitions, it is clear from Eq. 35 that $\bar{f}_{NMe}$ is driven by terms proportional to U_e, U_i, $dp/d\tilde{\Phi}$, and $dT_e/d\tilde{\Phi}$, while $\bar{f}_{NMi}$ is only driven by terms proportional to U_i and $dT_i/d\tilde{\Phi}$, as the $J_\parallel$ terms in Eq. 35 come from cross-species interaction terms that are dropped for the ions.

In Section V A, we qualitatively compare the distribution functions at low collisionality to the formalism of the NIES code and show, quantitatively, how the exact drift-kinetic solution behaves as we vary collisionality. In Section V B, we explain how Sauter-like ohmic and bootstrap current coefficients are computed in DK4D and benchmark these results to the Sauter formulae.

17

A. Qualitative behavior at low collisionality

As a first step in determining if DK4D is solving for the correct non-Maxwellian part of the electron and ion distribution functions, we can qualitatively compare our distribution functions at low collisionality to those found in NIES. This earlier code solved for the non-adiabatic part of the distribution functions, h_s as defined in Eqs. 6-9 of Ref. 26, in the $\delta_s \ll \hat{\nu} \ll 1$ limit. In doing so, it took advantage of certain convenient properties in this limit, namely that:

- h_s is independent of θ when written in terms of the normalized magnetic moment variable $\lambda = (1 - y^2) B_{max}/B$ instead of $y = \cos\chi$, where $B_{max}(\psi)$ is the maximum value of B on the $\tilde{\Phi}$ flux surface (assuming only one local maximum per surface),

- $h_s = 0$ in the $\lambda > 1$ trapped particle regime,

- and h_s is odd in y.

These properties were represented by writing

$$h_s = \varsigma(y) H(1 - \lambda) K_s(\tilde{\Phi}, w, \lambda), \tag{40}$$

where $\varsigma(y)$ is the sign of y and H is the Heaviside step function. NIES then solved for the quantity K_s. At finite collisionality, however, we expect that all three of these properties will be broken. The non-adiabatic parts of the distribution function should develop finite values in the trapped regime, some dependence on θ even when written in terms of λ, and some even structure in y.

In Figures 2, 3, and 4, we've plotted the non-adiabatic part of the electron distribution function at three collisionalities, $\hat{\nu} \sim 10^0$, $\hat{\nu} \sim 10^{-2}$, and $\hat{\nu} \sim 10^{-4}$, respectively. These were found using DK4D to calculate the converged solutions for f_{NMe} for the dashed flux surface of the NSTX equilibrium in Figure 1b and subtracting off the analytic, adiabatic $g_{e,1}$ defined in Ref. 17. As can clearly be seen in (y, θ) plots of these figures, denoted by (a), at higher collisionalities h_s does indeed have finite values within the trapped regime and an asymmetric structure in y, indicating contributions from both odd and even components. At the lower collisionalities, however, a steep boundary layer forms at the $\lambda = 1$ trapped-passing boundary, with h_s negligibly small in the trapped regime. Furthermore, the distribution
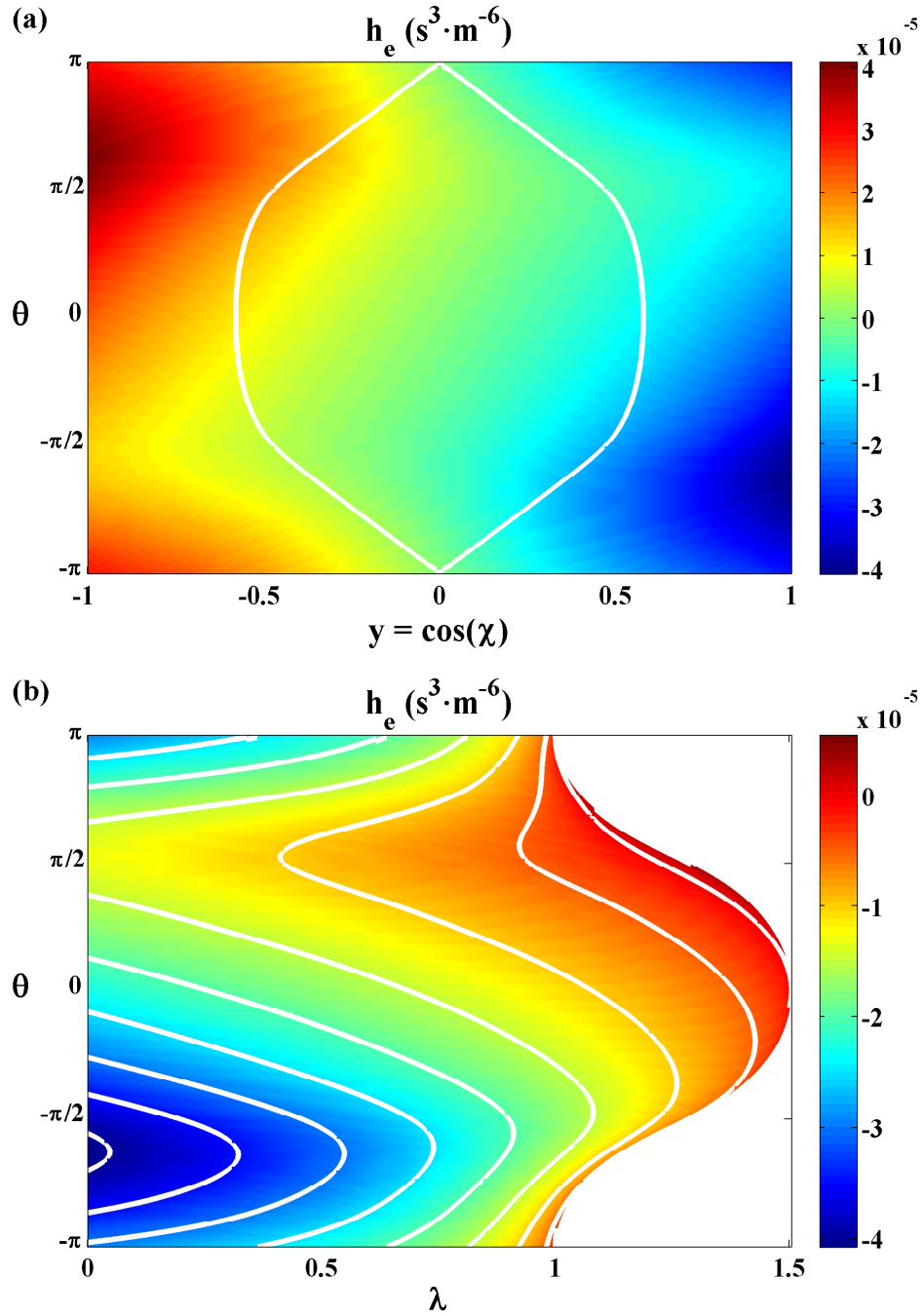
18

(a) h_e $(s^3 \cdot m^{-6})$

(b) h_e $(s^3 \cdot m^{-6})$
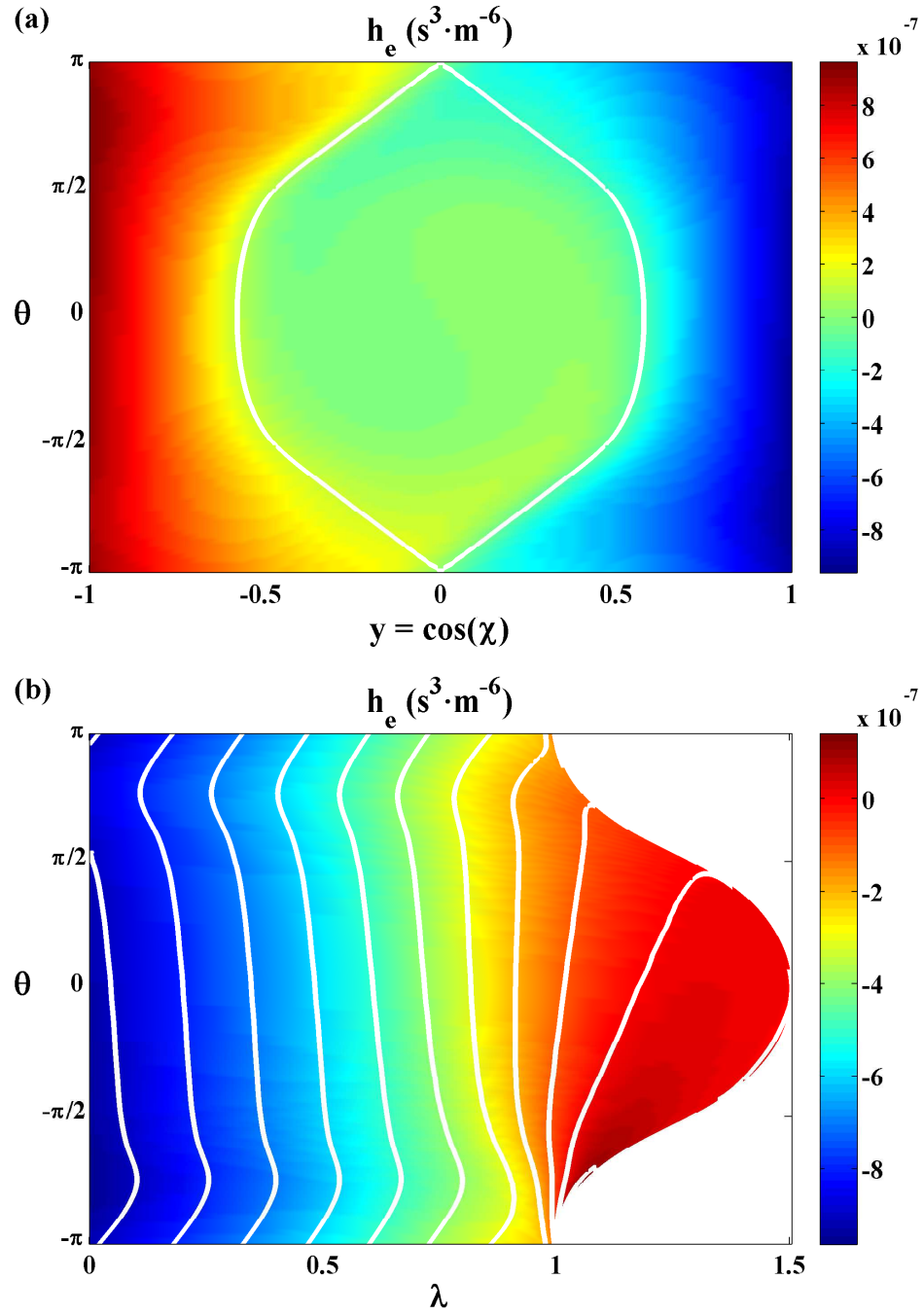
Figure 2. For $\hat{\nu} \sim 10^0$: the non-adiabatic electron distribution function, h_e, for the NSTX equilibrium of Figure 1 at $w \approx 1.2 v_{the0}$, where v_{the0} is the on-axis electron thermal velocity, and $\tilde{\Phi} \approx 0.148$ versus (a) y and θ, (b) λ and θ (for $y \geq 0$). In (a), the white line shows the $\lambda = 1$ trapped-passing boundary. In (b), the white lines show contours of constant h_e.

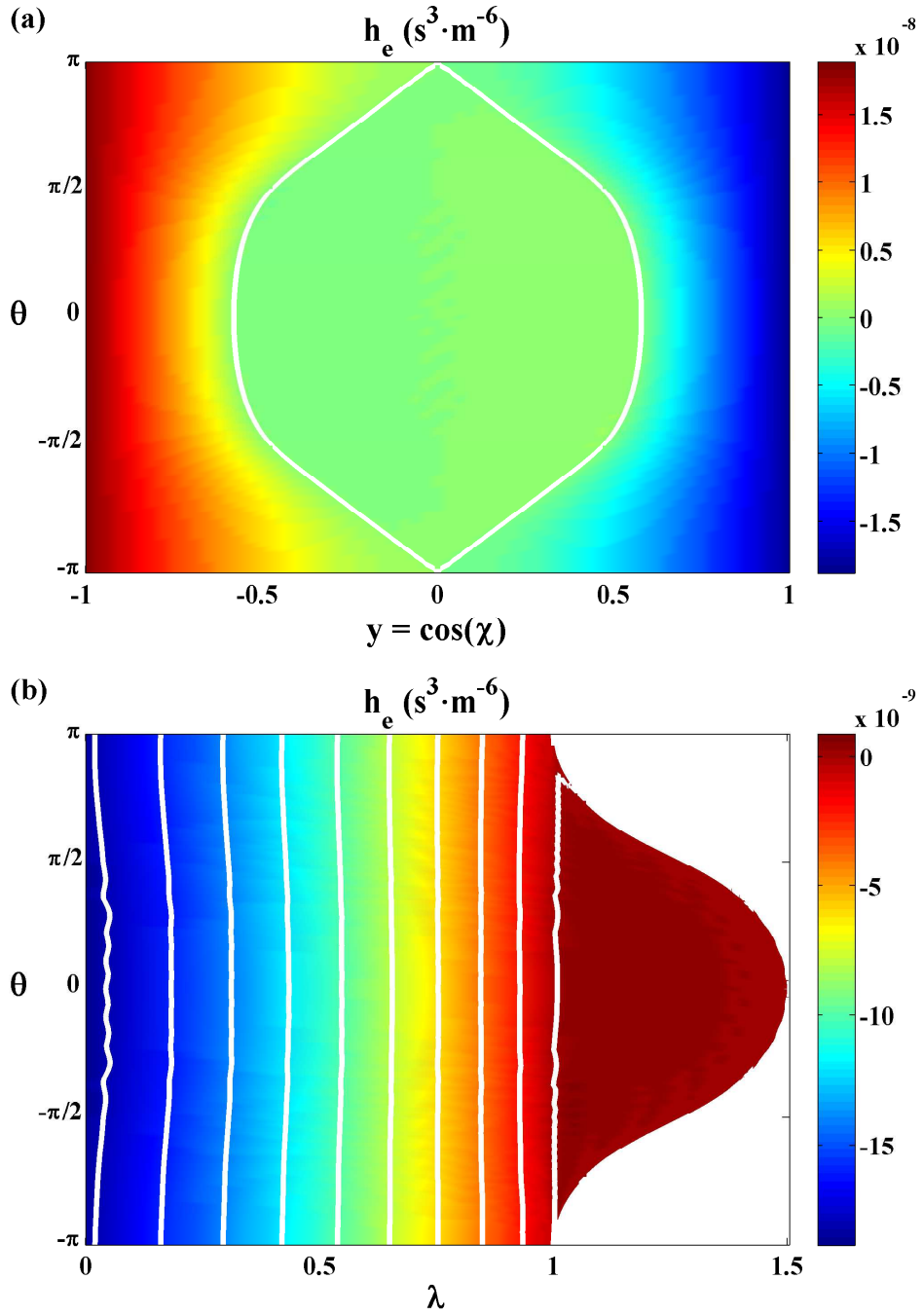Figure 3. For $\hat{\nu} \sim 10^{-2}$: the non-adiabatic electron distribution function, h_e, for the NSTX equilibrium of Figure 1 at $w \approx 1.2 v_{the0}$ and $\tilde{\Phi} \approx 0.148$ versus (a) y and θ, (b) λ and θ (for $y \geq 0$). In (a), the white line shows the $\lambda = 1$ trapped-passing boundary. In (b), the white lines show contours of constant h_e.

(a)

$h_e \ (\mathrm{s^3 \cdot m^{-6}})$

$\times 10^{-8}$

$y = \cos(\chi)$

(b)

$h_e \ (\mathrm{s^3 \cdot m^{-6}})$

$\times 10^{-9}$

λ

Figure 4. For $\hat{\nu} \sim 10^{-4}$: the non-adiabatic electron distribution function, h_e, for the NSTX equilibrium of Figure 1 at $w \approx 1.2 v_{the0}$ and $\tilde{\Phi} \approx 0.148$ versus (a) y and θ, (b) λ and θ (for $y \geq 0$). In (a), the white line shows the $\lambda = 1$ trapped-passing boundary. In (b), the white lines show contours of constant h_e.

function becomes increasingly odd in y at lower collisionalities. In the (λ, θ) plots of these figures, denoted by (b), one can also see that there is substantial variation with θ at high collisionality, while h_s is essentially independent of θ for $\hat{\nu} \sim 10^{-4}$.

Based on these observations, we can see that the DK4D solutions are behaving, qualitatively, just as we should expect based on the analytic analysis used to formulate the NIES code. Furthermore, these results show, quantitatively, the behavior of the exact drift-kinetic solution, as a function of both λ and θ, when the collisionality is varied. It bears repeating here that these simulations were done using the full, linearized Fokker-Planck-Landau collision operator and for a numerically-computed, finite aspect ratio Grad-Shafranov equilibrium.

B. Finite collisionality benchmark

Now that we've demonstrated that the DK4D distribution functions have the properties we would expect at low collisionality, we should benchmark the neoclassical conductivity and bootstrap current found in steady-state for finite collisionality. Note that a successful, cross-code, bootstrap current benchmark was performed for the NSTX equilbrium of Figure 1b using DK4D, NCLASS[20], NEO[19,23], and another new DKE solver written within the NIMROD MHD code framework[44], the results of which can be found in Ref. 44. Furthermore, we can compare our results to the well-known Sauter formulae, which defines the flux surface average of $\mathbf{J} \cdot \mathbf{B}$ as

$$\left\langle J_{\parallel} B \right\rangle = \sigma_{neo} \left\langle E_{\parallel} B \right\rangle + \frac{I}{\psi'} \left[\mathcal{L}_{31} \frac{dp}{d\tilde{\Phi}} + \mathcal{L}_{32} n_e \frac{dT_e}{d\tilde{\Phi}} + \alpha \mathcal{L}_{34} n_i \frac{dT_i}{d\tilde{\Phi}} \right]. \tag{41}$$

Here $E_{\parallel}$ is the parallel electric field, while the coefficients σ_{neo}, $\mathcal{L}_{31}$, $\mathcal{L}_{32}$, and $\mathcal{L}_{34}$ are functions of the trapped particle fraction,

$$f_t = 1 - \frac{3}{4} \left\langle \frac{B^2}{B_{max}^2} \right\rangle \int_0^1 \frac{\lambda d\lambda}{\langle 1 - \lambda B/B_{max} \rangle}, \tag{42}$$

and the electron effective collisionality, ν_{e*}. The ion flow coefficient, α, comes from

$$U_i = \alpha \frac{I}{e\psi' \langle B^2 \rangle} \frac{dT_i}{d\tilde{\Phi}}, \tag{43}$$

and is a function of f_t and the ion effective collisionality, ν_{i*}. All of these coefficients and the effective collisionalities are defined in Ref. 27.

In order to compare the DK4D results to the Sauter coefficients, we need to be able to write our results in a form like Eq. 41. We know from Eq. 38 that

$$\left\langle J_\parallel B \right\rangle = en \left(U_i - U_e \right) \left\langle B^2 \right\rangle + \frac{I}{\psi'} \frac{dp}{d\tilde{\Phi}}. \tag{44}$$

Furthermore, in our formulation, the flux surface average of the parallel Ohm's law, i.e., the stationary, parallel part of the electron momentum equation, can be written as

$$\left\langle E_\parallel B \right\rangle = en\eta_c \left\langle J_\parallel B \right\rangle + \left\langle \hat{F}_e B \right\rangle + \left\langle \left(p_{e\parallel} - p_{e\perp} \right) \mathbf{b} \cdot \nabla B \right\rangle, \tag{45}$$

where the classical resistivity is

$$\eta_c = \frac{2m_e \nu_e}{3\sqrt{2\pi} e^2 n}. \tag{46}$$

Now, since $\bar{f}_{NMe}$ is driven by U_e, U_i, $dp/d\tilde{\Phi}$, and $dT_e/d\tilde{\Phi}$ in equilibrium, we can expand the two moments in Eq. 45 as

$$\hat{F}_e = \Gamma_{U_e} U_e + \Gamma_{U_i} U_i + \frac{\Gamma_p}{\psi'} \frac{dp}{d\tilde{\Phi}} + \frac{\Gamma_{T_e}}{\psi'} \frac{dT_e}{d\tilde{\Phi}} \tag{47}$$

and

$$\left(p_{e\parallel} - p_{e\perp} \right) = \Delta_{U_e} U_e + \Delta_{U_i} U_i + \frac{\Delta_p}{\psi'} \frac{dp}{d\tilde{\Phi}} + \frac{\Delta_{T_e}}{\psi'} \frac{dT_e}{d\tilde{\Phi}}. \tag{48}$$

Thus, the parallel Ohm's law gives

$$en\eta_c \left\langle J_\parallel B \right\rangle = en \left\langle E_\parallel B \right\rangle - \left\langle \Gamma_{U_e} B + \Delta_{U_e} \mathbf{b} \cdot \nabla B \right\rangle U_e - \left\langle \Gamma_{U_i} B + \Delta_{U_i} \mathbf{b} \cdot \nabla B \right\rangle U_i$$
$$- \left\langle \Gamma_p B + \Delta_p \mathbf{b} \cdot \nabla B \right\rangle \frac{1}{\psi'} \frac{dp}{d\tilde{\Phi}} - \left\langle \Gamma_{T_e} B + \Delta_{T_e} \mathbf{b} \cdot \nabla B \right\rangle \frac{1}{\psi'} \frac{dT_e}{d\tilde{\Phi}}. \tag{49}$$

Using Eqs. 44 and 43 to eliminate U_e and U_i, we can rewrite Eq. 49 as

$$\left[e^2 n^2 \left\langle B^2 \right\rangle \eta_c - \left\langle \Gamma_{U_e} B + \Delta_{U_e} \mathbf{b} \cdot \nabla B \right\rangle \right] \left\langle J_\parallel B \right\rangle = e^2 n^2 \left\langle B^2 \right\rangle \left\langle E_\parallel B \right\rangle$$
$$- \frac{I}{\psi'} \left\{ \left[\frac{en \left\langle B^2 \right\rangle}{I} \left\langle \Gamma_p B + \Delta_p \mathbf{b} \cdot \nabla B \right\rangle + \left\langle \Gamma_{U_e} B + \Delta_{U_e} \mathbf{b} \cdot \nabla B \right\rangle \right] \frac{dp}{d\tilde{\Phi}} \right.$$
$$+ \frac{en \left\langle B^2 \right\rangle}{I} \left\langle \Gamma_{T_e} B + \Delta_{T_e} \mathbf{b} \cdot \nabla B \right\rangle \frac{dT_e}{d\tilde{\Phi}}$$
$$\left. + \alpha n \left[\left\langle \Gamma_{U_e} B + \Delta_{U_e} \mathbf{b} \cdot \nabla B \right\rangle + \left\langle \Gamma_{U_i} B + \Delta_{U_i} \mathbf{b} \cdot \nabla B \right\rangle \right] \frac{dT_i}{d\tilde{\Phi}} \right\}. \tag{50}$$

Thus, comparing to Eq. 41, we find that

$$\sigma_{neo} = \frac{e^2 n^2 \left\langle B^2 \right\rangle}{e^2 n^2 \left\langle B^2 \right\rangle \eta_c - \left\langle \Gamma_{U_e} B + \Delta_{U_e} \mathbf{b} \cdot \nabla B \right\rangle}, \tag{51}$$

23

$$\mathcal{L}_{31} = -\frac{en \langle B^2 \rangle \langle \Gamma_p B + \Delta_p \mathbf{b} \cdot \nabla B \rangle + I \langle \Gamma_{U_e} B + \Delta_{U_e} \mathbf{b} \cdot \nabla B \rangle}{I \left[e^2 n^2 \langle B^2 \rangle \eta_c - \langle \Gamma_{U_e} B + \Delta_{U_e} \mathbf{b} \cdot \nabla B \rangle \right]}, \tag{52}$$

$$\mathcal{L}_{32} = -\frac{en \langle B^2 \rangle \langle \Gamma_{T_e} B + \Delta_{T_e} \mathbf{b} \cdot \nabla B \rangle}{I \left[e^2 n^2 \langle B^2 \rangle \eta_c - \langle \Gamma_{U_e} B + \Delta_{U_e} \mathbf{b} \cdot \nabla B \rangle \right]}, \tag{53}$$

and

$$\mathcal{L}_{34} = -\frac{\langle \Gamma_{U_e} B + \Delta_{U_e} \mathbf{b} \cdot \nabla B \rangle + \langle \Gamma_{U_i} B + \Delta_{U_i} \mathbf{b} \cdot \nabla B \rangle}{e^2 n^2 \langle B^2 \rangle \eta_c - \langle \Gamma_{U_e} B + \Delta_{U_e} \mathbf{b} \cdot \nabla B \rangle}. \tag{54}$$

Now we need to find the ion flow coefficient α from DK4D. In our formulation, the flux surface average of the parallel component of the stationary, first-order ion momentum equation reduces to

$$\left\langle \left(p_{i\parallel} - p_{i\perp} \right) \mathbf{b} \cdot \nabla B \right\rangle = 0. \tag{55}$$

Recall that $\bar{f}_{NMi}$ is driven by terms proportional to U_i and $dT_i/d\tilde{\Phi}$. Thus, we can say that

$$\left(p_{i\parallel} - p_{i\perp} \right) = \Theta_{U_i} U_i + \frac{\Theta_{T_i}}{\psi'} \frac{dT_i}{d\tilde{\Phi}}, \tag{56}$$

and, by Eq. 55,

$$\langle \Theta_{U_i} \mathbf{b} \cdot \nabla B \rangle U_i + \langle \Theta_{T_i} \mathbf{b} \cdot \nabla B \rangle \frac{dT_i}{d\tilde{\Phi}} = 0. \tag{57}$$

Finally, comparing to Eq. 43, we find that

$$\alpha = -\frac{e \langle B^2 \rangle \langle \Theta_{T_i} \mathbf{b} \cdot \nabla B \rangle}{I \langle \Theta_{U_i} \mathbf{b} \cdot \nabla B \rangle}. \tag{58}$$

In Figures 5 and 6, we have plotted the conductivity and bootstrap current drive coefficient profiles for both the Sauter formulae and DK4D for the large aspect ratio equilibrium in Figure 1a as a function of trapped particle fraction for various collisionalities. Figures 7 and 8 provide the same comparison for the NSTX equilibrium of Figure 1b. As can be seen, there is generally very good agreement between the code and the analytic fits, particularly at low collisionality, across a wide range of trapped particle fractions (from $f_t \approx 0.1$ to $f_t \approx 0.9$) and collisionalities (from $\hat{\nu} \sim 10^{-4}$ to $\hat{\nu} \sim 1$). The largest deviations appear at the higher collisionalities, particularly for $\mathcal{L}_{32}$. This coefficient, however, involves a lot of cancellation between contributions from the electron-electron and electron-ion collision operators, which could explain the increased fractional difference observed between Sauter and DK4D. There is also some disagreement at low collisionality, though this appears mostly due to the difficulty of resolving the steep trapped-passing boundary layer with the Legendre polynomial and Fourier mode expansions in y and θ, respectively. This is a common problem in drift-kinetic

(a) Neoclassical Conductivity

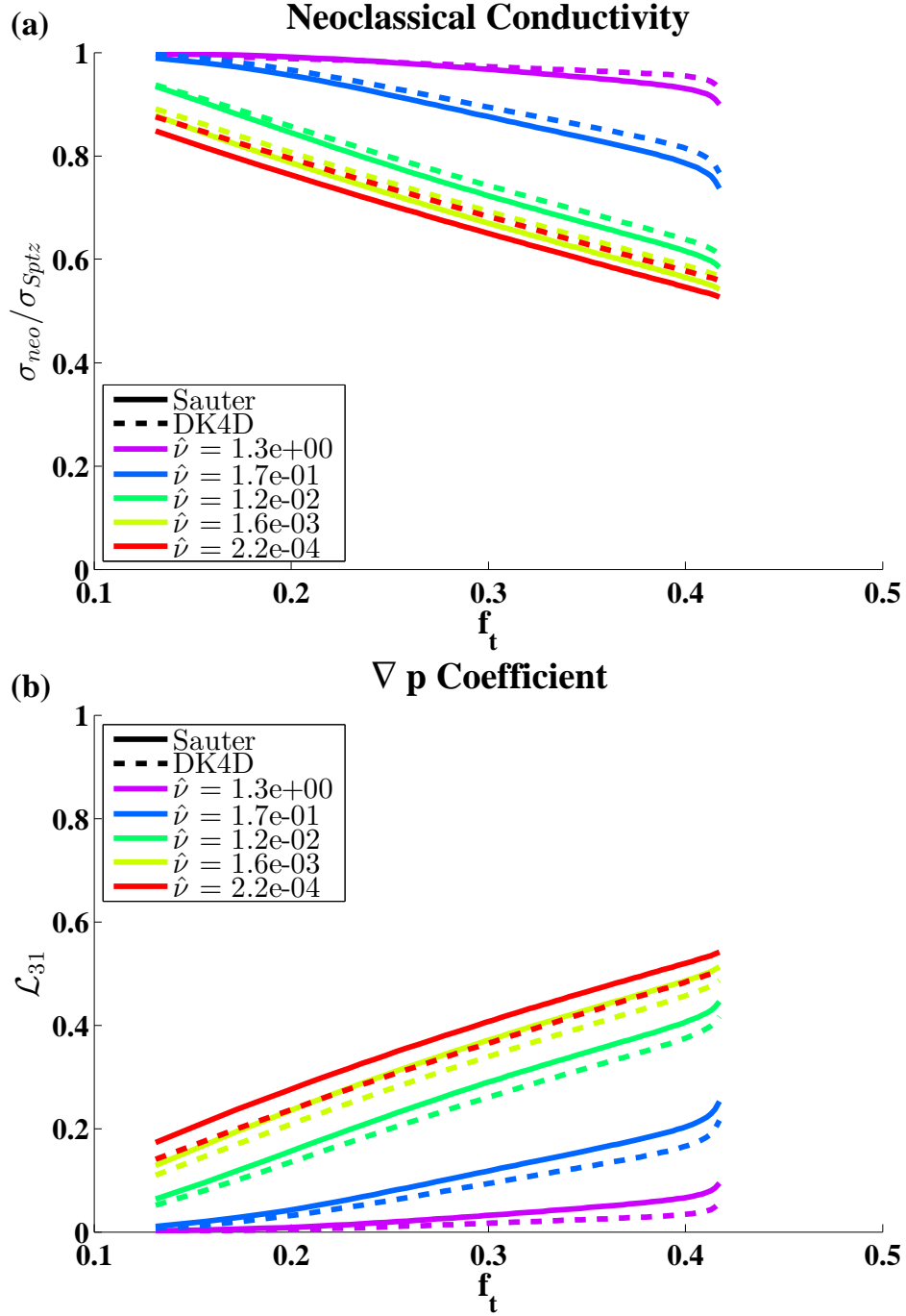**(b)** ∇ **p Coefficient**

Figure 5. A comparison between the neoclassical conductivity ratio (a) and pressure gradient bootstrap drive coefficient (b) calculated by the DK4D code (dashed lines) with the Sauter analytic fits (solid lines) as a function of trapped particle fraction for several collisionalities in the large aspect ratio equilibrium in Figure 1. Collisionalities in the legend are for the innermost flux surface.
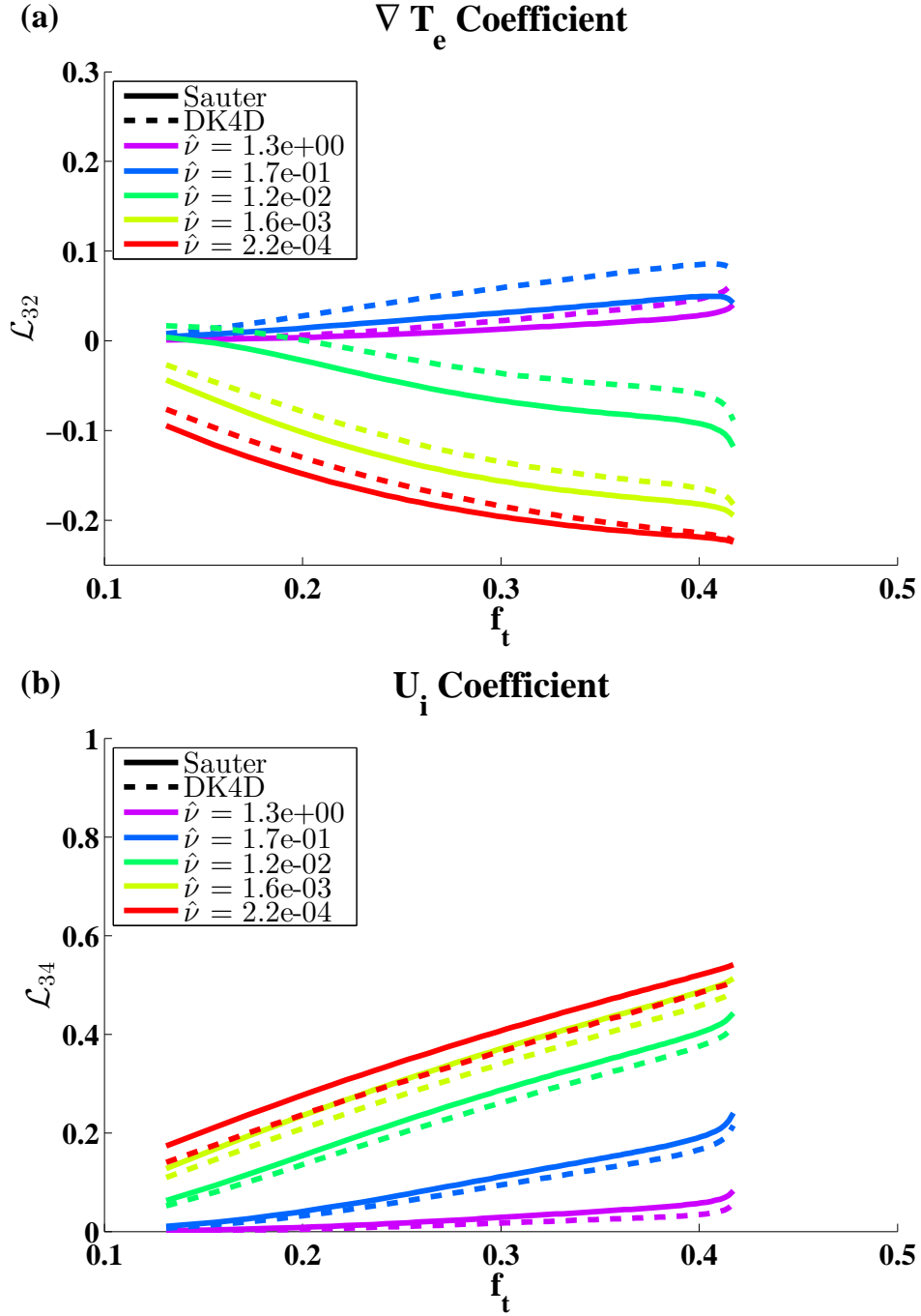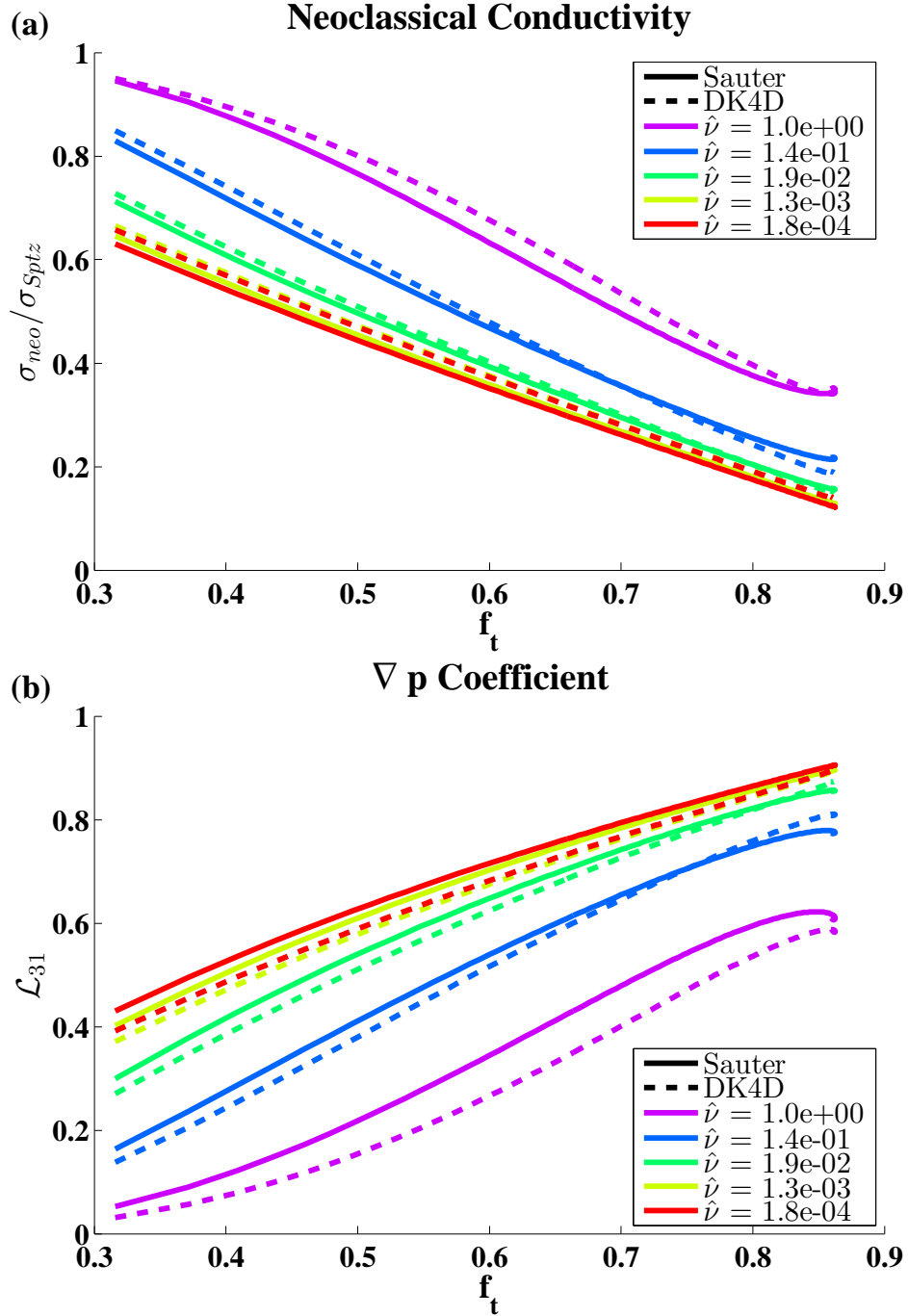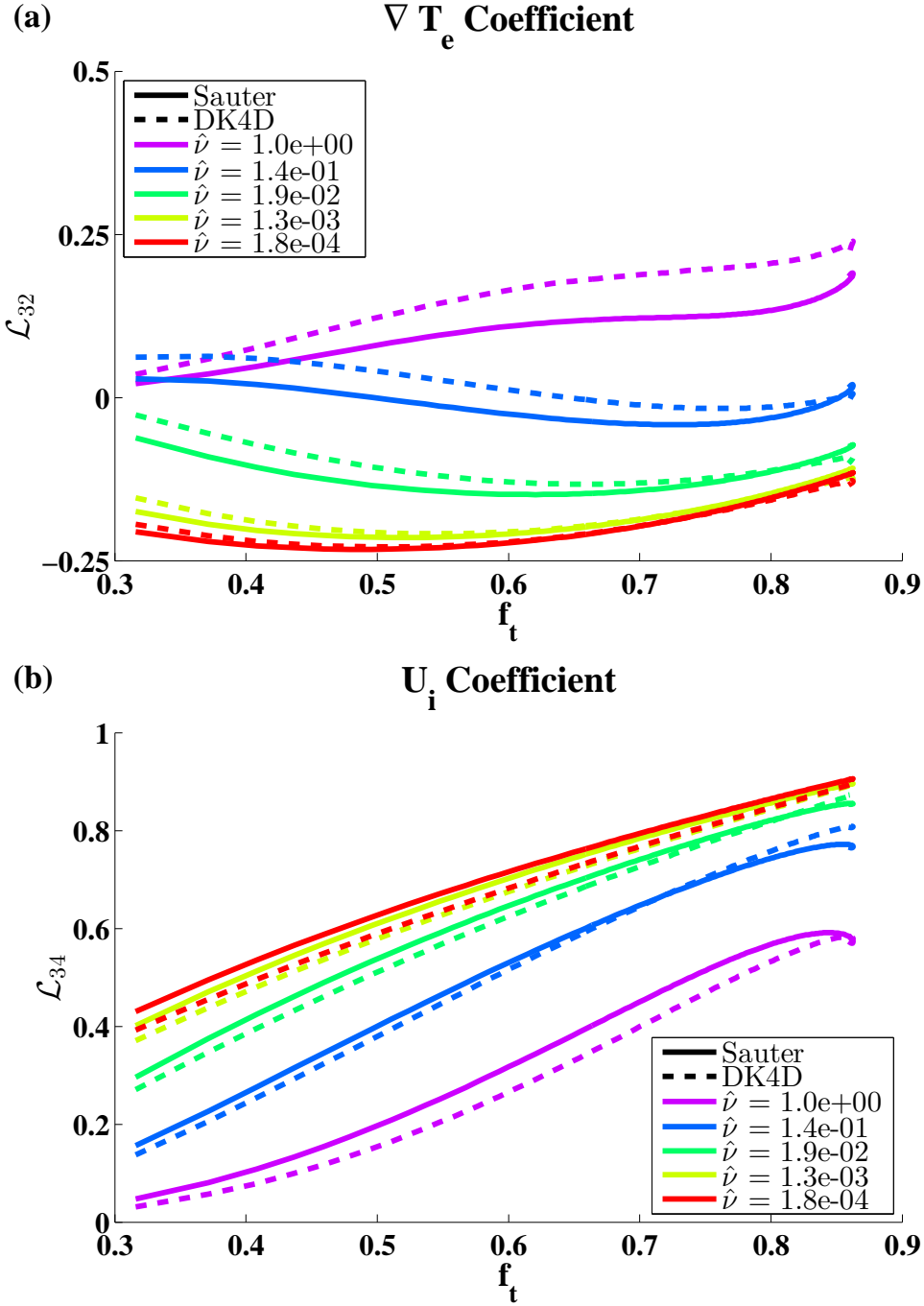
Figure 6. A comparison between the electron temperature gradient bootstrap drive coefficient (a) and ion flow bootstrap drive coefficient (b) calculated by the DK4D code (dashed lines) with the Sauter analytic fits (solid lines) as a function of trapped particle fraction for several collisionalities in the large aspect ratio equilibrium in Figure 1. Collisionalities in the legend are for the innermost flux surface.

(a) Neoclassical Conductivity

(b) ∇ p Coefficient

Figure 7. A comparison between the neoclassical conductivity ratio (a) and pressure gradient bootstrap drive coefficient (b) calculated by the DK4D code (dashed lines) with the Sauter analytic fits (solid lines) as a function of trapped particle fraction for several collisionalities in the NSTX equilibrium in Figure 1. Collisionalities in the legend are for the innermost flux surface.
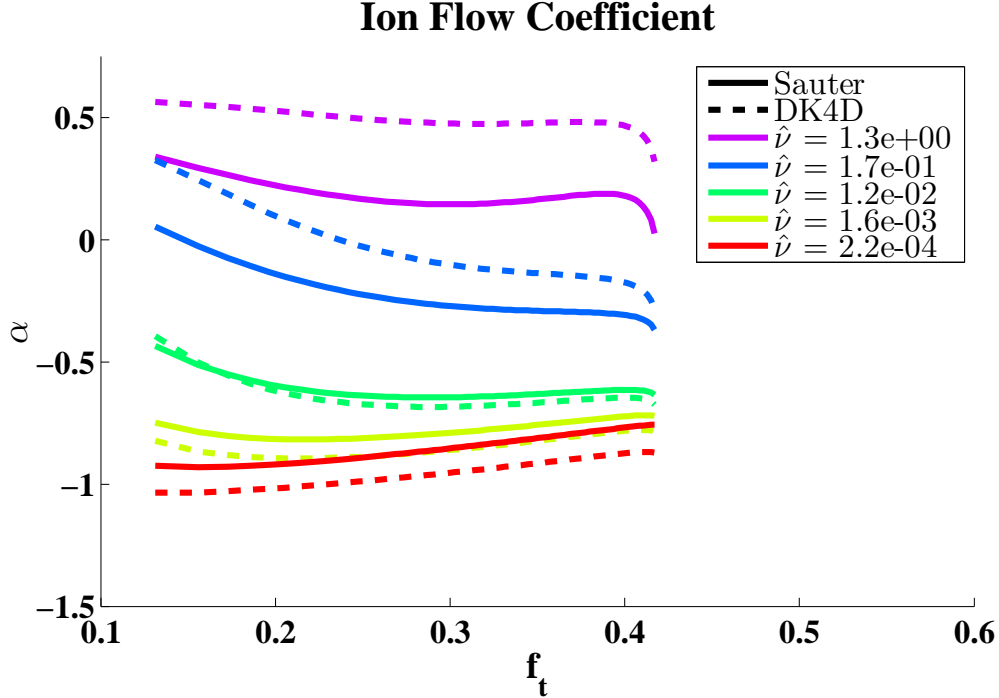
(a) ∇ T$_e$ **Coefficient**

(b) U$_i$ **Coefficient**

Figure 8. A comparison between the electron temperature gradient bootstrap drive coefficient (a) and ion flow bootstrap drive coefficient (b) calculated by the DK4D code (dashed lines) with the Sauter analytic fits (solid lines) as a function of trapped particle fraction for several collisionalities in the NSTX equilibrium in Figure 1. Collisionalities in the legend are for the innermost flux surface.

Ion Flow Coefficient

Figure 9. A comparison between the ion flow coefficient calculated by the DK4D code (dashed lines) with the Sauter analytic fits (solid lines) as a function of trapped particle fraction for several collisionalities in the large aspect ratio equilibrium in Figure 1. Collisionalities in the legend are for the innermost flux surface.

solvers (e.g., DKES[16]) that use these expansions. Overall, however, these results give us confidence that the electron solver within DK4D is working well.

Furthermore, the ion flow coefficients for the large aspect ratio and NSTX equilibria are plotted versus trapped particle fraction for various collisionalities in Figures 9 and 10, respectively. While the agreement between DK4D and Sauter is not quite as good for this coefficient as it was for the conductivity and bootstrap coefficients, we do still observe good agreement overall. Thus, this benchmark also gives us confidence that our ion solver is working well in DK4D.

VI. CONCLUSION

The DK4D code was developed to solve the time-dependent, axisymmetric form of the drift-kinetic equations derived in Refs. 17 and 18. We treated all the differential terms
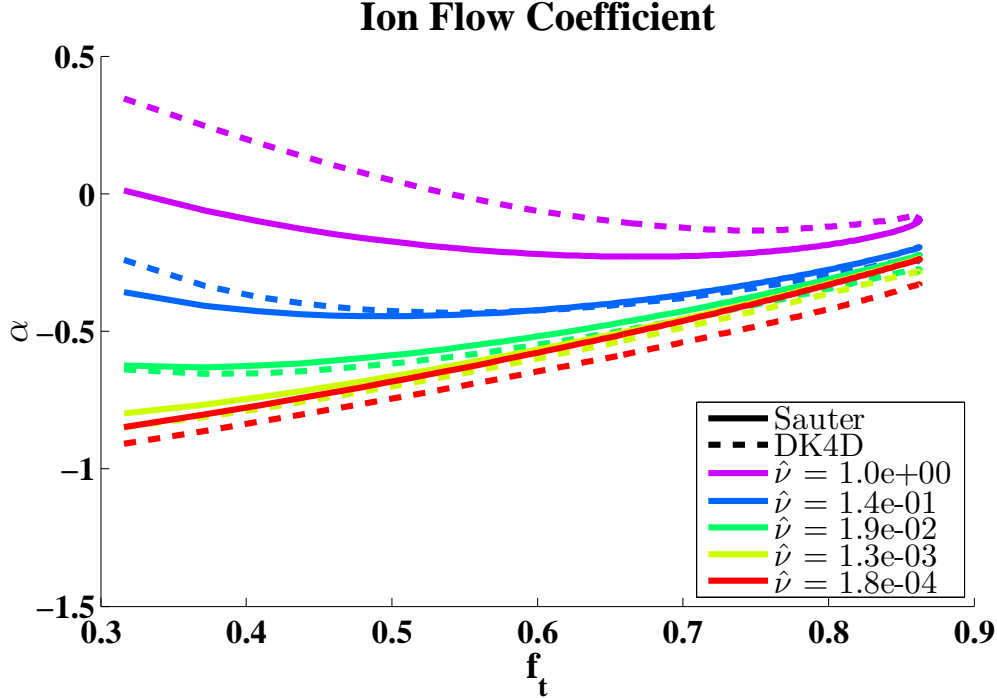
Ion Flow Coefficient

Figure 10. A comparison between the ion flow coefficient calculated by the DK4D code (dashed lines) with the Sauter analytic fits (solid lines) as a function of trapped particle fraction for several collisionalities in the NSTX in Figure 1. Collisionalities in the legend are for the innermost flux surface.

of the DKEs implicitly, while all the moment terms were treated explicitly. We expanded the non-Maxwellian part of the distribution function and the Rosenbluth potentials in finite elements in w, Legendre polynomials in y, and Fourier modes in θ, and then applied the Galerkin method to transform the DKE into a matrix equation. This was then implemented using the PETSc framework, allowing us access to a wide variety of direct and iterative solvers to advance the DKE in time. We then demonstrated that, when run to steady-state on a numerical Grad-Shafranov equilibrium, DK4D reproduced qualitatively some of the features of the distribution functions at low collisionality that were observed in NIES[26]. This study also demonstrated how the exact drift-kinetic solution behaves, as a function of λ and θ, as the collisionality is varied. Finally, we benchmarked these steady-state results to the Sauter analytic fits. Together with the cross-code benchmark performed between NEO, NCLASS, and a new solver written in the NIMROD MHD code framework that has

30

been published in Ref. 44, these Sauter benchmarks have given us confidence that DK4D is capable of calculating the correct steady-state distribution function at finite collisionality for an axisymmetric equilibrium. We should note that DK4D is a more comprehensive code than NIES, which was only meant to serve as a proof-of-principle. Still, DK4D can have trouble converging at very low collisionalities, a property common to all DKE solvers that use Legendre expansions in the pitch angle variable. Therefore, there may be applications for which NIES would be better suited than DK4D.

It is important to note where this work stands in relation to the present state of the field. When considering all the properties of DK4D, we can say with confidence that, in one way or another, it has some capabilities beyond those of any other continuum DKE solver in the literature, excepting, perhaps, CQLP and the new solver written in NIMROD. DK4D solves for the full distribution function using the linearized Fokker-Planck-Landau collision operator. Neither DKES nor NCLASS is capable of this. Furthermore, while NEO is a more comprehensive drift-kinetic equation solver in many ways, it solves only the steady-state DKE, while DK4D solves for the full time-dependence. In any case, our approach has the unique feature of solving for the distribution functions directly in the moving reference frame of each species' macroscopic flow, which simplifies the task of evaluating accurately the higher gyrotropic moments needed for the fluid closure, namely the pressure anisotropy, the parallel heat flux, and the parallel collisional friction force. Furthermore, the formulation considered ensures that the net density, momentum, and temperature of our computed non-Maxwellian parts of the distribution function are all zero (excepting some small numerical error). Thus, DK4D is ideally suited for self-consistent coupling to MHD time-evolution codes.

The ultimate goal of this ongoing work is to create a drift-kinetic solver capable of coupling to a nonaxisymmetric MHD time-evolution code to perform realistic hybrid simulations of macroscopic plasma instabilities, like sawtooth oscillations, NTMs, and ELMs. We believe that DK4D forms the basis for such a solver, though significant work remains to reach this goal. Initial simulations that couple DK4D to a reduced, axisymmetric, transport-timescale MHD solver have already been presented in the lead author's dissertation[28]. These results, in addition to some new ones involving an ELM-like pressure collapse, are expected to be the subject of a forthcoming, peer-reviewed publication.

ACKNOWLEDGMENTS

REFERENCES

[1] R. Carrera, R. D. Hazeltine, and M. Kotschenreuther, Physics of Fluids **29**, 899 (1986).

[2] Z. Chang, J. D. Callen, E. D. Fredrickson, R. V. Budny, C. C. Hegna, K. M. McGuire, M. C. Zarnstorff, and T. group, Phys. Rev. Lett. **74**, 4663 (1995).

[3] P. H. Rebut, R. J. Bickerton, and B. E. Keen, Nuclear Fusion **25**, 1011 (1985).

[4] P. C. de Vries, M. F. Johnson, B. Alper, P. Buratti, T. C. Hender, H. R. Koslowski, V. Riccardo, and JET-EFDA Contributors, Nuclear Fusion **51**, 053018 (2011).

[5] O. Sauter, R. J. La Haye, Z. Chang, D. A. Gates, Y. Kamada, H. Zohm, A. Bondeson, D. Boucher, J. D. Callen, M. S. Chu, T. A. Gianakon, O. Gruber, R. W. Harvey, C. C. Hegna, L. L. Lao, D. A. Monticello, F. Perkins, A. Pletzer, A. H. Reiman, M. Rosenbluth, E. J. Strait, T. S. Taylor, A. D. Turnbull, F. Waelbroeck, J. C. Wesley, H. R. Wilson, and R. Yoshino, Physics of Plasmas **4**, 1654 (1997).

[6] R. J. La Haye, Physics of Plasmas **13**, 055501 (2006).

[7] P. B. Snyder, H. R. Wilson, J. R. Ferron, L. L. Lao, A. W. Leonard, T. H. Osborne, A. D. Turnbull, D. Mossessian, M. Murakami, and X. Q. Xu, Physics of Plasmas **9**, 2037 (2002).

[8] P. Snyder, H. Wilson, J. Ferron, L. Lao, A. Leonard, D. Mossessian, M. Murakami, T. Osborne, A. Turnbull, and X. Xu, Nuclear Fusion **44**, 320 (2004).

[9] G. T. A. Huysmans, Plasma Physics and Controlled Fusion **47**, B165 (2005).

[10] H. R. Wilson, S. C. Cowley, A. Kirk, and P. B. Snyder, Plasma Physics and Controlled Fusion **48**, A71 (2006).

[11] S. Saarelma, A. Alfier, M. N. A. Beurskens, R. Coelho, H. R. Koslowski, Y. Liang, I. Nunes, and JET-EFDA Contributors, Plasma Physics and Controlled Fusion **51**, 035001 (2009).

[12] S. von Goeler, W. Stodiek, and N. Sauthoff, Phys. Rev. Lett. **33**, 1201 (1974).

[13] C. R. Sovinec, A. H. Glasser, T. A. Gianakon, D. C. Barnes, R. A. Nebel, S. E. Kruger, D. D. Schnack, S. J. Plimpton, A. Tarditi, and M. S. Chu, Journal of Computational Physics **195**, 355 (2004).

[14] S. C. Jardin, N. Ferraro, J. Breslau, and J. Chen, Computational Science and Discovery **5**, 014002 (2012).

[15] C. S. Chang, S. Ku, and H. Weitzner, Physics of Plasmas **11**, 2649 (2004).

[16] W. I. van Rij and S. P. Hirshman, Physics of Fluids B: Plasma Physics **1**, 563 (1989).

[17] J. J. Ramos, Physics of Plasmas **17**, 082502 (2010).

[18] J. J. Ramos, Physics of Plasmas **18**, 102506 (2011).

[19] E. A. Belli and J. Candy, Plasma Physics and Controlled Fusion **54**, 015015 (2012).

[20] W. A. Houlberg, K. C. Shaing, S. P. Hirshman, and M. C. Zarnstorff, Physics of Plasmas **4**, 3230 (1997).

[21] O. Sauter, R. W. Harvey, and F. L. Hinton, Contributions to Plasma Physics **34**, 169 (1994).

[22] R. W. Harvey and M. G. McCoy, in *Proceedings of the IAEA Technical Committee Meeting on Advances in Simulation and Modeling of Thermonuclear Plasmas* (1992) pp. 489–526, available through USDOC/NTIS No. DE93002962; see also, `http://www.compxco.com/cql3d.html` CQL3D Manual.

[23] E. A. Belli and J. Candy, Plasma Physics and Controlled Fusion **50**, 095010 (2008).

[24] S. P. Hirshman and D. J. Sigmar, Nuclear Fusion **21**, 1079 (1981).

[25] S. Chapman and T. G. Cowling, *The Mathematical Theory of Non-uniform Gases* (Cambridge University Press, 1939).

[26] B. C. Lyons, S. C. Jardin, and J. J. Ramos, Physics of Plasmas **19**, 082515 (2012).

[27] O. Sauter, C. Angioni, and Y. R. Lin-Liu, Physics of Plasmas **6**, 2834 (1999), erratum: Ibid. **9**, 5140 (2002).

[28] B. C. Lyons, *Self-consistent hybrid neoclassical-magnetohydrodynamic simulations of ax-*

isymmetric plasmas, Ph.D. thesis, Princeton University (2014).

[29] S. Jardin, *Computational Methods in Plasma Physics* (CRC Press, 2010).

[30] C. Strang and G. Fix, *An Analysis of the Finite Element Method* (Prentice Hall, Englewood Cliffs, NJ, 1973).

[31] F. Hariri and M. Ottaviani, Computer Physics Communications **184**, 2419 (2013).

[32] T. Nakamura and T. Yabe, Computer Physics Communications **120**, 122 (1999).

[33] S. Balay, M. F. Adams, J. Brown, P. Brune, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, and H. Zhang, "PETSc Web page," `http://www.mcs.anl.gov/petsc` (2014).

[34] S. Balay, M. F. Adams, J. Brown, P. Brune, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, and H. Zhang, "PETSc users manual," Tech. Rep. ANL-95/11 - Revision 3.4 (Argonne National Laboratory, 2013).

[35] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith, in *Modern Software Tools in Scientific Computing*, edited by E. Arge, A. M. Bruaset, and H. P. Langtangen (Birkhäuser Press, 1997) pp. 163–202.

[36] P. R. Amestoy, I. S. Duff, J. Koster, and L'Excellent, J.-Y., SIAM Journal of Matrix Analysis and Applications **23**, 15 (2001).

[37] P. R. Amestoy, A. Guermouche, L'Excellent, J.-Y., and S. Pralet, Parallel Computing **32**, 136 (2006).

[38] X. S. Li, ACM Trans. Math. Software **31**, 302 (2005).

[39] X. Li, J. Demmel, J. Gilbert, L. Grigori, M. Shao, and I. Yamazaki, "SuperLU Users' Guide," Tech. Rep. LBNL-44289 (Lawrence Berkeley National Laboratory, 1999) `http://crd.lbl.gov/~xiaoye/SuperLU/`. Last update: August 2011.

[40] X. S. Li and J. W. Demmel, ACM Trans. Mathematical Software **29**, 110 (2003).

[41] D. Pfirsch and A. Schluter, Tech. Rep. MPI/PA/7/62 (Max-Planck-Institut, 1962).

[42] F. L. Hinton and R. D. Hazeltine, Rev. Mod. Phys. **48**, 239 (1976).

[43] J. DeLucia, S. C. Jardin, and A. M. M. Todd, Journal of Computational Physics **37**, 183 (1980).

[44] E. D. Held, S. E. Kruger, J.-J. Ji, E. A. Belli, and B. C. Lyons, Physics of Plasmas (2014), (forthcoming).

Princeton Plasma Physics Laboratory
Office of Reports and Publications

Managed by
Princeton University

under contract with the
U.S. Department of Energy
(DE-AC02-09CH11466)