Concept for Numerical Calculation of 3D MHD Equilibria with Flow and FLR Effects*)

Daniel RABURN and Atsushi FUKUYAMA

Kyoto University, Kyoto 606-8501, Japan

(Received 12 December 2011 / Accepted 8 May 2012)

Equilibrium flows and 3D effects can significantly impact plasma stability and energy confinement. Further, in equilibria with flow, FLR effects can play an important role. Presently, there exist a number of codes which can calculate MHD equilibria with a subset of the above effects, such as: the FLOW code [L. Guazzotto, R. Betti, J. Manickam and S. Kaye, Phys. Plasmas **11**(2), 604 (2004)], the PIES code [H.S. Greenside, A.H. Reiman and A. Salas, J. Comput. Phys. **81**(1), 102 (1989)], and the ItoGSEQ code [D. Raburn and A. Fukuyama, Phys. Plasmas **17**(12), 122504 (2010)]. Using insights gained from these codes, the concept for a new code for calculation of 3D MHD equilibria with flow and FLR effects has been developed; the code is called the Kyoto ITerative Equilibrium Solver (KITES).

Keywords: magnetohydrodynamics, equilibrium, computation, magnetic island, finite Larmor radius effect

DOI: 10.1585/pfr.7.2403108

1. Introduction

One of the simplest self-consistent models of plasma equilibrium is obtained using the single-fluid magnetohydrodynamic (MHD) model, assuming no flow, under axisymmetry. However, both nonsymmetric effects and flow can significantly alter equilibrium: equilibrium flows can produce transport barriers and profile pedestals [1, 2], and nonsymmetric effects can produce magnetic islands and stochastic regions. Further, in the presence of a profile pedestal, small scale-length effects – such as finite Larmor radius (FLR) effects – may be important. A review of MHD equilibrium is provided in Sec. 2.

There are a number of codes which are capable of calculating MHD equilibrium with a variety of the physical effects mentioned above. In Sec. 3, we provide a review of three selected codes: (1) the FLOW code, which can calculate axisymmetric single-fluid equilibria with flow; (2) the ItoGSEQ code, which can calculate axisymmetric reduced two-fluid equilibria with flow in an inverse-aspect-ratio expansion; and, (3) the PIES code, which can calculate nonsymmetric single-fluid equilibria without flow.

We have begun development on a code for calculating nonsymmetric reduced two-fluid equilibria with flow, known as the KITES (Kyoto ITerative Equilibrium Solver) code. The concept for KITES is presented in Sec. 4. A summary of the concept for KITES is presented in Sec. 5.

2. Review of MHD Equilibrium

Consider the following equations for the magnetohydrodynamic (MHD) model of a plasma:

$$\nabla \cdot \boldsymbol{B} = 0, \tag{1}$$

$$\nabla \times \boldsymbol{B} = \mu_0 \boldsymbol{j}, \tag{2}$$

$$\boldsymbol{j} \times \boldsymbol{B} = \nabla p + m_{\rm i} n \boldsymbol{u} \cdot \nabla \boldsymbol{u} + \alpha_{\rm i}^{\rm gv} \nabla \cdot \boldsymbol{\Pi}_{\rm i}^{\rm gv}, \tag{3}$$

$$\nabla \cdot (n\boldsymbol{u}) = 0, \tag{4}$$

$$\boldsymbol{u}_{\rm e} \times \boldsymbol{B} = \nabla \Phi - \alpha_{\rm Hall} \nabla p_{\rm e}/(en), \tag{5}$$

$$\boldsymbol{u}_{\rm e} \equiv \boldsymbol{u} - \alpha_{\rm Hall} \boldsymbol{j}/(en), \tag{6}$$

$$\boldsymbol{u} \cdot \nabla p_{\rm i} + \gamma p_{\rm i} \nabla \cdot \boldsymbol{u} = -\tfrac{2}{5} \alpha_{\rm i}^{\rm hf} \gamma \nabla \cdot \boldsymbol{q}_{\rm i}, \tag{7}$$

$$\boldsymbol{u}_{\rm e} \cdot \nabla p_{\rm e} + \gamma p_{\rm e} \nabla \cdot \boldsymbol{u}_{\rm e} = -\tfrac{2}{5} \alpha_{\rm e}^{\rm hf} \gamma \nabla \cdot \boldsymbol{q}_{\rm e}, \tag{8}$$

where $\boldsymbol{\Pi}_{\rm i}^{\rm gv}$ is the ion gyroviscous tensor, $\boldsymbol{q}_{\rm i}$ and $\boldsymbol{q}_{\rm e}$ are the ion and electron heat fluxes, and the terms $\alpha_{\rm Hall}$, $\alpha_{\rm i}^{\rm gv}$, $\alpha_{\rm i}^{\rm hf}$, and $\alpha_{\rm e}^{\rm hf}$ are artificial coefficients which have been introduced to provide control over various physical effects: in the most realistic model, all should be set to unity. The other symbols have their usual meaning [3]. The heat fluxes and gyroviscosity tensor can be calculated using the expressions given by Ramos [4].

Eqs. (7) and (8) are based on the standard MHD adiabatic model for closure, which is valid in the high collisionality limit. An alternative is Grad's guiding center particle (GCP) model, which instead uses a kinetic model for the dynamics of particles along the field lines [5, 6].

While we are interested in studying nonsymmetric two-fluid MHD equilibria with flow under both the adiabatic and GCP models, a short review of simpler cases is in order. One of the simplest self-consistent models for describing plasma equilibrium is the single-fluid MHD model ($\alpha_{...} = 0$) under axisymmetry ($\partial/\partial\phi = 0$) with no bulk fluid flow ($\boldsymbol{u} = 0$). In this case, equilibrium is described by the

author's e-mail: draburn@p-grp.nucleng.kyoto-u.ac.jp

*) This article is based on the presentation at the 21st International Toki Conference (ITC21).

well-known Grad-Shafranov equation [7]:

$$\nabla \cdot \left(R^{-2}\nabla\psi\right) = R^{-2}I\,(\mathrm{d}I/\mathrm{d}\psi) + \mu_0\,(\mathrm{d}p/\mathrm{d}\psi)\,, \tag{9}$$

where R is the major radius, ψ is the poloidal magnetic flux, and I is the poloidal current. Equilibrium depends on the boundary conditions for ψ, as well as the two free functions $I(\psi)$ and $p(\psi)$, which must be constant on each flux surface.

Axisymmetric single-fluid equilibria with flow under adiabatic closure ($\partial/\partial\phi = 0$, $\alpha_{...} = 0$, $\boldsymbol{u} \neq 0$) is governed by a *generalized* Grad-Shafranov equation and a Bernoulli equation [8]. In this case, equilibrium depends on the boundary conditions for ψ as well as *five* free functions; the five free functions are related to the following quantities: I, p, n, Φ, and Ψ, where Ψ is the stream function, which measures the poloidal fluid flow [9]. There are a number of important complications in this case. First, there may be a density discontinuity where the poloidal flow changes from sub- to super-poloidal sonic, known as the poloidal-sonic discontinuity [2]. Additionally, the differential equation governing equilibrium will typically become hyperbolic in some small region [6, 8, 10], making numerical calculation difficult. Two-fluid effects ($\alpha_{\mathrm{Hall}} = 1$, but other αs remaining zero) modify the criterion for hyperbolicity, but both elliptic and hyperbolic regions are still possible [11]. Under the GCP model, however, for a low-beta plasma, the system is always elliptic as long as the flow remains sub-Alfvénic [6].

Ito and Nakajima have developed a formulation for large aspect-ratio axisymmetric FLR reduced two-fluid MHD equilibria with flow under adiabatic closure ($\partial/\partial\phi = 0$, $\alpha_{...} \neq 0$, $\boldsymbol{u} \neq 0$) [3, 12]. The formulation makes use of an expansion in the inverse aspect-ratio ε with the ordering $\rho/a \sim \varepsilon$, where ρ is the ion Larmor radius and a is the plasma minor radius. In this formulation, through second order in ε, equilibrium depends on five free functions of the poloidal flux, which are related to the following quantities: I, p_{e}, p_{i}, n, and Φ. A sixth free function, related to Ψ, would be necessary through third order in ε. The system is always elliptic, but, the poloidal-sonic *discontinuity* instead becomes a *singularity*; the exact criteria for the singularity is modified by two-fluid and FLR effects.

Now, consider nonsymmetric single-fluid equilibria without flow ($\partial/\partial\phi \neq 0$, $\alpha_{...} = 0$, $\boldsymbol{u} = 0$). There is no guarantee of properly nested flux surfaces: magnetic islands and stochastic regions are possible. Regardless, just like the axisymmetric case, equilibrium is governed by boundary conditions and just *two* free functions, one related to the current and one related to the pressure. However, instead of these being defined in terms of the poloidal flux ψ, they are typically defined in terms of the full magnetic field $\boldsymbol{B}$, with each free function constant on each magnetic field line.

Finally, consider the case of nonsymmetric single-fluid MHD equilibria with flow ($\partial/\partial\phi \neq 0$, $\alpha_{...} = 0$, $\boldsymbol{u} \neq 0$). Bondeson and Iacono have shown that, under the adiabatic model, the system is always hyperbolic when there is flow across the field lines. However, under GCP, the system is always elliptic for a low-beta plasma as long as the flow remains sub-Alfvénic [6].

3. Review of Selected Codes

There exist many codes for calculating MHD equilibria in a variety of models. Here, we briefly review three codes: (1) the FLOW code; (2) the ItoGSEQ code; and, (3) the PIES code.

The FLOW code is capable of calculating axisymmetric single-fluid MHD equilibria with bulk fluid flow ($\partial/\partial\phi = 0$, $\alpha_{...} = 0$, $\boldsymbol{u} \neq 0$) [8]. FLOW handles the poloidal-sonic discontinuity by examining the Bernoulli equation explicitly and separating the computational domain in to sub- and super-poloidal sonic regions. The code encounters some difficulties when dealing with large poloidal flows in high-beta plasmas due to the presence of a hyperbolic region; however, the authors report that reliable results can still typically be achieved.

The ItoGSEQ code can calculate equilibria under the Ito-Nakajima formulation ($\partial/\partial\phi = 0$, $\alpha_{...} \neq 0$, $\boldsymbol{u} \neq 0$, $\varepsilon \ll 1$) [3, 12]. The solver has provided preliminary study of such equilibria, but, because of the linear expansion in ε, the poloidal-sonic *discontinuity* becomes a *singularity*, and the solver cannot be applied directly to trans-poloidal-sonic flows. However, the code has been used to verify the existence of equilibria with flow that do not have the discontinuity.

The PIES code is capable of calculating single-fluid MHD equilibria without flow in nonsymmetric systems ($\partial/\partial\phi \neq 0$, $\alpha_{...} = 0$, $\boldsymbol{u} = 0$) [13]. The basic idea of the algorithm will now be explained. Define λ:

$$\lambda \equiv j_{\|}/B, \tag{10}$$

where $j_{\|}$ is the component of $\boldsymbol{j}$ parallel to $\boldsymbol{B}$, and define λ_* to be the average of λ along a field line. Let ξ be a field line label: it must be constant on each field line and must be different between different field lines. PIES requires that the boundary conditions on $\boldsymbol{B}$ and the functions $\lambda_*(\xi)$ and $p(\xi)$ be specified. The PIES algorithm can be summarized as follows:

1. Start with some guess for $\boldsymbol{B}$ over the computational domain.
2. Calculate ξ over the computational domain by following field lines and assigning some distinct value of ξ to each line.
3. Calculate p over the computational domain using the specified $p(\xi)$.
4. Calculate the perpendicular current density $\boldsymbol{j}_\perp$ over the computational domain using $\boldsymbol{j}_\perp = \boldsymbol{B} \times \nabla p/B^2$.
5. Use the specified $\lambda_*(\xi)$ to determine λ over the computational domain such that: $\langle\lambda\rangle_\xi = \lambda_*$ and $\boldsymbol{B} \cdot \nabla\lambda = -\nabla \cdot \boldsymbol{j}_\perp$.

6. Calculate an updated guess for the magnetic field $\boldsymbol{B}^{(+)}$ such that: $\nabla \cdot \boldsymbol{B}^{(+)} = 0$, $\nabla \times \boldsymbol{B}^{(+)} = \lambda \boldsymbol{B} + \boldsymbol{j}_\perp$, and $\boldsymbol{B}^{(+)}$ satisfies the specified boundary conditions.
7. Using the updated guess for the magnetic field, repeat the above steps until the difference between successive guesses for the magnetic field is sufficiently small.

Steps 2 and 5 are very difficult, and PIES makes use of a system of magnetic coordinates in order to effectively treat magnetic islands and stochastic field lines. As a consequence of these complexities, PIES has been considered to be prohibitively slow for some applications.

Recently, an external Jacobian-free Newton-Krylov (JFNK) wrapper has been developed to speed-up PIES [14, 15]. Define the function $\underline{\boldsymbol{F}}(\underline{\boldsymbol{B}})$:

$$\underline{\boldsymbol{F}}(\underline{\boldsymbol{B}}) \equiv \underline{\boldsymbol{B}^{(+)}} - \underline{\boldsymbol{B}}, \tag{11}$$

where $\underline{\boldsymbol{B}}$ is some discretized representation of $\boldsymbol{B}$ and $\underline{\boldsymbol{B}^{(+)}}$ is the equivalent discretization of the $\boldsymbol{B}^{(+)}$ calculated by PIES for that $\boldsymbol{B}$. Observe that, if $\underline{\boldsymbol{F}} = \underline{0}$, then, the corresponding $\boldsymbol{B}$ must satisfy the equilibrium equations and specified functions to some discretization-dependent numerical precision. Thus, the calculation of equilibrium is reduced to finding a root of the function $\underline{\boldsymbol{F}}$. This is exactly what is accomplished by the external wrapper.

4. Concept for New Code

4.1 Overview

We have developed an algorithm for calculating nonsymmetric MHD equilibria with flow and two-fluid effects ($\partial/\partial\phi \neq 0$, $\alpha_{...} \neq 0$, $\boldsymbol{u} \neq 0$). We are writing a code to implement this algorithm; the code is called KITES for Kyoto ITerative Equilibrium Solver. The plan for KITES is to use an algorithm similar to PIES with the external JFNK wrapper. Thus, we only need to set up a function $\underline{\boldsymbol{F}}$ similar to that used in PIES. However, before getting in to the algorithm for calculating $\underline{\boldsymbol{F}}$, several comments are in order.

First, we must address the question of what parameters and free functions need to be specified in order for equilibrium to be properly constrained. Based on the simpler cases described in Sec. 2, we believe that it is appropriate to take as specified the field line averages of the following six quantities: λ, n, T_e, T_i, Φ, and ν, where λ is defined in Eq. (10) and ν is defined by:

$$\nu \equiv u_\parallel / B, \tag{12}$$

where $u_\parallel$ is the component of $\boldsymbol{u}$ parallel to $\boldsymbol{B}$.

Second, for the quantities λ, n, T_e, T_i, Φ, and ν, we intend to break each up in to a field line average and field line variation part. For example: $\lambda = \widetilde{\lambda} + \lambda_*$, with $\langle\widetilde{\lambda}\rangle_\xi = 0$ and $\langle\lambda_*\rangle_\xi = \lambda_*$. We take the functions λ_*, n_*, T_{e*}, T_{i*}, Φ_*, and ν_* to be given. We iteratively guess values for $\widetilde{n}$, $\widetilde{T}_e$, $\widetilde{T}_i$, $\widetilde{\Phi}$, and $\widetilde{\nu}$. Note that we do not need to guess $\widetilde{\lambda}$, because it can be easily calculated from the equilibrium equations.

Third, we take as unknown the plasma vector potential $\boldsymbol{A}_{\rm plas}$ under the Coulomb gauge ($\nabla\cdot\boldsymbol{A}_{\rm plas} = 0$), with the vacuum vector potential $\boldsymbol{A}_{\rm vac}$ as given. We use a free boundary for the plasma with a given vacuum vessel shape, and magnetic field lines intersecting the vacuum vessel wall will have n and j zero on that line. Note that taking $\boldsymbol{A}$ as unknown rather than $\boldsymbol{B}$ is is an important distinction from PIES because, when used with JFNK, even if the updated guess always has a divergence of zero, there is no guarantee that the input will have a divergence of zero – this is inconsequential for $\boldsymbol{A}$ but would cause difficulty with field line following for $\boldsymbol{B}$.

4.2 Major hurdles

There appear to be several major hurdles to the development of the code:

1. Calculating ξ and the topology of the magnetic field;
2. Applying the free functions and solving the magnetic differential equations;
3. Handling the poloidal-sonic discontinuity; and,
4. Validating the code in a case where a solution is known to exist.

The first two hurdles are shared with PIES and the third hurdle is shared with FLOW. Whereas PIES makes use of magnetic coordinates to help address the first two hurdles, we hope to use purely physical coordinates for KITES. This will presumably make KITES slower than PIES when handling good flux surfaces, but more robust overall.

The second hurdle involves solving problems of the following form: find λ such that $\boldsymbol{B}\cdot\nabla\lambda = -\nabla\cdot\boldsymbol{j}_\perp$ and $\langle\lambda\rangle_\xi = \lambda_*$. Rather than relying on the approach used in PIES, we have searched for a tractable approach which does not rely on magnetic coordinates. Because the operators $\boldsymbol{B}\cdot\nabla$ and $\langle...\rangle_\xi$ are both linear, given some discretization method, the problem can be cast in to the following form:

$$\underline{\underline{M}}\,\underline{\lambda} = \underline{b}, \tag{13}$$

for some matrix $\underline{\underline{M}}$ and vector $\underline{b}$, where $\underline{\lambda}$ indicates the discretized λ. $\underline{\underline{M}}$ and $\underline{b}$ are informally given by the following expressions:

$$\underline{\underline{M}} = \begin{pmatrix} \boldsymbol{B}\cdot\nabla \\ \langle...\rangle_\xi \end{pmatrix}, \quad \underline{b} = \begin{pmatrix} -\nabla\cdot\boldsymbol{j}_\perp \\ \lambda_* \end{pmatrix}, \tag{14}$$

Because the matrix $\underline{\underline{M}}$ is non-square, $\underline{\lambda}$ may be overdetermined. The least-squares solution to this problem is determined by:

$$\underline{\underline{M}}^{\rm T}\underline{\underline{M}}\,\underline{\lambda} = \underline{\underline{M}}^{\rm T}\underline{b}, \tag{15}$$

which can be solved for $\underline{\lambda}$ by a standard linear solver.

At present, it is unclear how the third hurdle will apply to KITES. In particular, it is unclear how the poloidal-sonic discontinuity translates to nonsymmetric equilibrium. With the inclusion of two-fluid and FLR effects, it is hoped that the discontinuity will be replaced by a gradient. Regardless, as demonstrated by the solver for the Ito-Nakajima formulation, we do know that equilibria with flow without the discontinuity are possible. We intend to use such equilibria as a starting point for addressing the discontinuity, if it remains.

Finally, consider the ellipticity of the system of equations. Although the inclusion of two-fluid effects may modify the ellipticity of the system, it is not clear if there will generally be solutions with flow across the field lines for the given boundary conditions. We plan to use the code to investigate the existence of solutions; however, before doing that, we need to be able to verify the code when including various physical effects. If necessary, we will do this using the GCP model, which is known to typically be elliptic without the two-fluid effects. If we are unable to verify the code under the adiabatic model, we hope to verify the code under GCP, then use the verified code to investigate the existence of equilibria under the adiabatic model.

4.3 Step-by-step algorithm

The algorithm for calculating $\underline{\boldsymbol{F}}$ under the adiabatic model is as follows:

1. Guess some $\boldsymbol{A}_{\text{plas}}$, $\widetilde{n}$, $\widetilde{T}_{\text{e}}$, $\widetilde{T}_{\text{i}}$, $\widetilde{\Phi}$, and $\widetilde{\nu}$.
2. Calculate $\boldsymbol{A}$ using: $\boldsymbol{A} = \boldsymbol{A}_{\text{vac}} + \boldsymbol{A}_{\text{plas}}$.
3. Calculate $\boldsymbol{B}$ using: $\boldsymbol{B} = \nabla \times \boldsymbol{A}$.
4. Calculate $\boldsymbol{j}$ using: $\boldsymbol{j} = \left[\nabla(\nabla\cdot\boldsymbol{A}) - \nabla^2\boldsymbol{A}\right]/\mu_0$.
5. Calculate ξ using field line following.
6. Calculate n, T_{e}, T_{i}, Φ, and ν using the guessed field line variation and the specified field line average.
7. Calculate $\boldsymbol{u}_\perp$ using $\boldsymbol{B}\times$ Ohm's law [Eq. (16)].
8. Calculate $\boldsymbol{u}$ and $\boldsymbol{u}_{\text{e}}$: $\boldsymbol{u} = \boldsymbol{u}_\perp + \nu\boldsymbol{B}$, $\boldsymbol{u}_{\text{e}} = \boldsymbol{u} - \alpha_{\text{Hall}}\boldsymbol{j}/(en)$.
9. Calculate an updated guess for $\widetilde{n}$ using conservation of mass $[\nabla\cdot\left(\widetilde{n}^{(+)}\boldsymbol{u}\right) = -\nabla\cdot(n_*\boldsymbol{u})]$ and $\langle\widetilde{n}^{(+)}\rangle_\xi = 0$.
10. Calculate an updated guess for $\widetilde{T}_{\text{e}}$ using the electron equation of state [Eq. (17)] and $\langle\widetilde{T}_{\text{e}}^{(+)}\rangle_\xi = 0$.
11. Calculate an updated guess for $\widetilde{T}_{\text{i}}$ using the ion equation of state [Eq. (18)] and $\langle\widetilde{T}_{\text{i}}^{(+)}\rangle_\xi = 0$.
12. Calculate an updated guess for $\widetilde{\Phi}$ using $\boldsymbol{B}\cdot$ Ohm's law $[\boldsymbol{B}\cdot\nabla\widetilde{\Phi}^{(+)} = -\alpha_{\text{Hall}}\boldsymbol{B}\cdot\nabla\widetilde{p}_{\text{e}}/(en)]$ and $\left\langle\widetilde{\Phi}^{(+)}\right\rangle_\xi = 0$.
13. Calculate a linearly updated guess for $\widetilde{\nu}$ using $\boldsymbol{B}\cdot$ the force balance equation [Eq. (19)] and $\left\langle\widetilde{\nu}^{(+)}\right\rangle_\xi = 0$.
14. Calculate an updated $\boldsymbol{j}_\perp$ using $\boldsymbol{B}\times$ the force balance equation [Eq. (20)].
15. Calculate an updated $\widetilde{\lambda}$ using conservation of charge $[\boldsymbol{B}\cdot\nabla\widetilde{\lambda}^{(+)} = -\nabla\cdot\boldsymbol{j}_\perp^{(+)}]$ and $\left\langle\widetilde{\lambda}^{(+)}\right\rangle_\xi = 0$.
16. Calculate an updated $\boldsymbol{j}$ using: $\boldsymbol{j}^{(+)} = \boldsymbol{j}_\perp^{(+)} + \left(\lambda_* + \widetilde{\lambda}^{(+)}\right)\boldsymbol{B}$.
17. Calculate an updated guess for $\boldsymbol{A}_{\text{plas}}$ using the Biot-Savart law on the computational boundary and a Poisson solver internally.

$$\boldsymbol{u}_\perp = \boldsymbol{B}\times\left[\nabla\Phi + \alpha_{\text{Hall}}\left(\boldsymbol{j}\times\boldsymbol{B} - \nabla p_{\text{e}}\right)/(en)\right]/B^2, \tag{16}$$

$$\begin{aligned}&\boldsymbol{u}_{\text{e}}\cdot\nabla\left(n\widetilde{T}_{\text{e}}^{(+)}\right) + \gamma n\widetilde{T}_{\text{e}}^{(+)}\nabla\cdot\boldsymbol{u}_{\text{e}} = -\tfrac{2}{5}\gamma\alpha_{\text{e}}^{\text{hf}}\nabla\cdot\boldsymbol{q}_{\text{e}}\\ &\quad -\boldsymbol{u}_{\text{e}}\cdot\nabla\left(nT_{\text{e}\,*}\right) - \gamma nT_{\text{e}\,*}\nabla\cdot\boldsymbol{u}_{\text{e}},\end{aligned} \tag{17}$$

$$\begin{aligned}&\boldsymbol{u}\cdot\nabla\left(n\widetilde{T}_{\text{i}}^{(+)}\right) + \gamma n\widetilde{T}_{\text{i}}^{(+)}\nabla\cdot\boldsymbol{u} = -\tfrac{2}{5}\gamma\alpha_{\text{i}}^{\text{hf}}\nabla\cdot\boldsymbol{q}_{\text{i}}\\ &\quad -\boldsymbol{u}\cdot\nabla\left(nT_{\text{i}\,*}\right) - \gamma nT_{\text{i}\,*}\nabla\cdot\boldsymbol{u},\end{aligned} \tag{18}$$

$$\begin{aligned}&m_{\text{i}}n\boldsymbol{B}\cdot\left\{\nabla\left[\left(\nu_* + \tfrac{1}{2}\widetilde{\nu}\right)B^2\widetilde{\nu}^{(+)}\right] - \boldsymbol{u}_\perp\times\left[\nabla\times\left(\widetilde{\nu}^{(+)}\boldsymbol{B}\right)\right]\right\}\\ &\quad = m_{\text{i}}n\boldsymbol{B}\cdot\left\{\boldsymbol{u}_\perp\times\left[\nabla\times\left(\boldsymbol{u}_\perp + \nu_*\boldsymbol{B}\right)\right] - \tfrac{1}{2}\nabla\left[u_\perp^2 + \nu_*^2B^2\right]\right\}\\ &\qquad -\boldsymbol{B}\cdot\nabla p - \alpha_{\text{i}}^{\text{gv}}\boldsymbol{\Pi}_{\text{i}}^{\text{gv}},\end{aligned} \tag{19}$$

$$\boldsymbol{j}_\perp^{(+)} = \boldsymbol{B}\times\left(m_i n\boldsymbol{u}\nabla\boldsymbol{u} + \nabla p + \alpha_{\text{i}}^{\text{gv}}\boldsymbol{\Pi}_{\text{i}}^{\text{gv}}\right)/B^2. \tag{20}$$

5. Summary

We have developed an algorithm for calculating non-symmetric reduced two-fluid MHD equilibria with flow. The algorithm is based on that used in the PIES code, but has been modified and extended to allow for the additional physics; in particular, the new algorithm uses a method for solving the magnetic differential equation which does not rely on magnetic coordinates. We expect that the code can be verified using Grad's guiding center particle (GCP) model for closure, and we plan to use the code to investigate equilibria under both the GCP and adiabatic models.

This work was supported by Grand-in-Aid for Scientific Research (S) (20226017) from JSPS, Japan.

[1] T.S. Taylor, H.ST. John, A.D. Turnbull *et al.*, Plasma Phys. Control. Fusion **36**, B229 (1994).
[2] R. Betti and J.P. Freidberg, Phys. Plasmas **7**(6), 2439 (2000).
[3] A. Ito and N. Nakajima, AIP Conference Proceedings **1069**, 121 (2008).
[4] J.J. Ramos, Phys. Plasmas **12**, 052102 (2005).
[5] H. Grad, in *Proceedings of the Symposium on Electromagnetic and Fluid Dynamics of Gaseous Plasma* (Polytech. Inst. of Brooklyn, New York, 1961) p.37.
[6] A. Bondeson and R. Iacono, Phys. Fluids B **1**, 1431 (1989).
[7] J. Wesson, *Tokamaks*, 3rd ed. (Oxford University Press, New York, NY, 2004) p.110.
[8] L. Guazzotto, R. Betti, J. Manickam and S. Kaye, Phys. Plasmas **11**(2), 604 (2004).
[9] D. Raburn and A. Fukuyama, Plasma Fusion Res. **6**, 2403044 (2011).
[10] H.P. Zehrfeldt and B.J. Greene, Nucl. Fusion **12**, 659 (1972).
[11] A. Ishida, C.O. Harahap, L.C. Steinhauer and Y.-K.M. Peng, Phys. Plasmas **11**, L81 (2004).
[12] D. Raburn and A. Fukuyama, Phys. Plasmas **17**(12), 122504 (2010).
[13] H.S. Greenside, A.H. Reiman and A. Salas, J. Comput. Phys. **81**(1), 102 (1989).
[14] D.A. Knoll and D.E. Keyes, SIAM J. Comput. Phys. **193**, 357 (2004).
[15] D. Raburn, PhD Thesis. Princeton University, Princeton, New Jersey, USA (2011).