Physics of Plasmas

ATEQ: Adaptive toroidal equilibrium code

Cite as: Phys. Plasmas **29**, 072503 (2022); https://doi.org/10.1063/5.0091015
Submitted: 10 March 2022 • Accepted: 09 June 2022 • Published Online: 06 July 2022

Linjin Zheng, M. T. Kotschenreuther, F. L. Waelbroeck, et al.

Physics of Plasmas ARTICLE scitation.org/journal/php

ATEQ: Adaptive toroidal equilibrium code

Cite as: Phys. Plasmas **29**, 072503 (2022); doi: 10.1063/5.0091015
Submitted: 10 March 2022 · Accepted: 9 June 2022 ·
Published Online: 6 July 2022

Linjin Zheng,[1,a)] M. T. Kotschenreuther,[1] F. L. Waelbroeck,[1] and Y. Todo[2]

AFFILIATIONS

[1]Institute for Fusion Studies, University of Texas at Austin, Austin, Texas 78712, USA
[2]National Institute for Fusion Science, National Institutes of Natural Sciences, Toki, Gifu 509-5292, Japan

[a)]Author to whom correspondence should be addressed: lzheng@austin.utexas.edu

ABSTRACT

A radially adaptive numerical scheme is developed to solve the Grad–Shafranov equation for axisymmetric magnetohydrodynamic equilibrium. A decomposition with independent solutions is employed in the radial direction, and Fourier decomposition is used in the poloidal direction. The independent solutions are then obtained using an adaptive shooting scheme together with the multi-region matching technique in the radial direction. Accordingly, the adaptive toroidal equilibrium (ATEQ) code is constructed for axisymmetric equilibrium studies. The adaptive numerical scheme in the radial direction improves considerably the accuracy of the equilibrium solution. The decomposition with independent solutions effectively reduces the matrix size in solving the magnetohydrodynamic equilibrium problem. The reduction of the matrix size is about an order of magnitude as compared with the conventional radially grid-based numerical schemes. Also, in this ATEQ numerical scheme, no matter how accuracy in the radial direction is imposed, the size of matrices basically does not change. The small matrix size scheme gives ATEQ more flexibility to address the requirement of the number of Fourier components in the poloidal direction in tough equilibrium problems. These two unique features, the adaptive shooting and small matrix size, make ATEQ useful to improve tokamak equilibrium solutions.

I. INTRODUCTION

Solving the magnetohydrodynamic (MHD) equilibrium problem is fundamental in plasma physics and plays an essential role, in particular, in the magnetic confinement approach to fusion. In axisymmetric geometries, the equilibrium problem is reduced to solving the Grad–Shafranov equation.[1,2] Since this equation is nonlinear, a numerical solution is necessary in general. In advanced tokamaks,[3] two circumstances conspire to make the solution of the Grad–Shafranov equation particularly challenging. First, advanced tokamaks rely on broad current distributions to increase β, the ratio of kinetic to magnetic pressure. This leads to a current profile peak near the edge and to the sensitivity of the stability limit to details in the geometry of the plasma edge. Second, they rely on the H-mode for confinement. The pressure gradients in H-mode drive localized, peaked bootstrap currents near the edge that add to the difficulty in two ways, first by increasing the stiffness of the Grad–Shafranov problem and second by increasing the accuracy needed to calculate the stability of edge localized modes (ELM)[4] as well as resistive wall modes (RWM).[5]

Great efforts have been made previously to develop numerical solvers for the Grad–Shafranov equation. The applications for these numerical solvers are diverse, ranging from the interpretation of experimental observations[6,7] to the design of operation scenarios,[8] real-time control of experiments,[7,9,10] the analysis of the stability and transport properties of various configurations,[11] and the optimization of machine designs.[3,12] The diversity of the applications leads to different requirements regarding properties of the algorithm such as speed, accuracy, stability, and flexibility. These different requirements are partly responsible for the multiplicity of solution strategies. The 1991 review article by Takeda and Tokuda[13] describes early codes including J-Solver,[14] VMEC,[15] TOQ,[16] and others.[17–19] Subsequent efforts led to the development of the codes CHEASE,[20] CORSICA,[21] and EFIT.[6] References 22–33 describe further works. As reviewed in Ref. 13, the methods for solving the Grad–Shafranov equation are categorized into two types: the Eulerian or "direct" and the Lagrangian or "inverse" numerical schemes. The finite difference, finite element, and Fourier decomposition methods are employed to discretize the equation. In all cases, iteration is used to handle the nonlinearity.

Despite the great successes achieved with the existing codes in various scenarios, challenges remain for solving the equilibrium problem, especially for the cases with high beta, strong shaping, and diverter geometries that give rise to separatrices. The need for adaptive solvers was realized a long time ago. It has, for example, led to the development of the VMEC code for 3D equilibria.[15] Later, the edge

equilibrium code (EEC) was developed in order to address the numerical challenges pertaining to the tokamak edge equilibrium problem.[34]

In this work, we introduce a new adaptive numerical scheme to solve the Grad–Shafranov equation and describe its implementation in the ATEQ (adaptive toroidal equilibrium) code for tokamaks. The code uses a decomposition with independent solutions in the radial direction and Fourier decomposition in the poloidal direction. It then obtains the independent solutions with adaptive shooting together with the multi-region matching technique in the radial direction. The adaptive numerical scheme in the radial direction improves considerably the accuracy of the equilibrium solution. The decomposition with independent solutions effectively reduces the matrix size in solving the magnetohydrodynamic equilibrium problem. The adaptive numerical scheme has been successfully used in the linear MHD and kinetic stability codes, AEGIS[35] and AEGIS-K.[36]

In addition to its adaptive nature, the reduction of the matrix size by ATEQ is about an order of magnitude, as compared to the conventional radially grid-based numerical schemes. Also, in this ATEQ numerical scheme, no matter how accuracy in the radial direction is imposed, the size of the matrices basically does not change. Note that all numerical schemes for solving the Grad–Shafranov equation ultimately reduce to solving matrix equations. The size of matrices then matters. To achieve high accuracy, especially for tough problems related to the axis, X-point, or pedestal, etc., one has to increase the grid density in the radial and poloidal directions in the grid-based codes, or the radial grid density and the number of poloidal Fourier components in the Fourier-decomposition based codes. The dramatic reduction of matrix size by ATEQ is important for this research. The Fourier-decomposition based codes remain to be important tools in this field, for example, CORSICA is used for ITER and VMEC is still popular. The small matrix size scheme gives ATEQ more flexibility to address the requirement of the number of Fourier components in the poloidal direction for tough equilibrium problems.

The remainder of this paper is organized as follows: Sec. II introduces the MHD equilibrium equations. Section III describes the formulation of numerical equations. Section IV gives the numerical procedure and results. Section V presents the benchmark studies and comparison with the existing equilibrium codes. Finally, Sec. VI presents the conclusions and discussion.

II. MHD EQUILIBRIUM EQUATIONS

In this section, we describe the MHD equilibrium equations and the goal of this work. Force balance, Ampére's law, and the absence of magnetic charge form the following basic set of equations describing the MHD equilibrium for a static plasma ($\mathbf{V} = 0$):[38]

$$\mathbf{J} \times \mathbf{B} = \nabla p, \tag{1}$$

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{J}, \tag{2}$$

$$\nabla \cdot \mathbf{B} = 0, \tag{3}$$

where $\mathbf{B}$ is the magnetic field, $\mathbf{J}$ represents the current density, p denotes the pressure, μ_0 is the magnetic constant, and boldface denotes the vectors.

The paper addresses axisymmetric toroidal equilibria. For such equilibria, it is convenient to use a cylindrical coordinate system (X, Z, ϕ), where ϕ is the toroidal angle, Z denotes vertical coordinate, and X is radial coordinate from the toroidal axisymmetric axis on the $\phi = 0$ plane. In this coordinate system, the magnetic field in the axisymmetric case can be represented as[38]

$$\mathbf{B} = \nabla\phi \times \nabla\chi + g\nabla\phi, \tag{4}$$

where χ is the poloidal magnetic flux. Both pressure $p(\chi)$ and $g(\chi)$ are flux functions.

Using the representation in Eq. (4) and equilibrium equations (1)–(3), one can derive the following Grad–Shafranov equation:[1,2]

$$X^2 \nabla \cdot \frac{\nabla\chi}{X^2} = -\mu_0 X^2 p' - gg', \tag{5}$$

where prime denotes the derivative with respect to the poloidal flux χ. The MHD equilibrium is fully determined by χ.

The two free functions $p(\chi)$ and $g(\chi)$ need to be specified to determine χ from Eq. (5). In practice, one usually specify p and g as the functions of normalized flux $\hat{\chi} = \chi/\chi_a$, where χ_a is the poloidal flux at the edge or on the last closed flux surface and the poloidal flux is assumed to be zero at the magnetic axis.

The goal of the present paper is to lay out a new numerical scheme to solve Eq. (5) and describe the ATEQ code that implements this scheme. The paper restricts attention to the fixed boundary problem, i.e., solving Eq. (5) with the plasma boundary specified. We defer consideration of the free boundary problem to future work.

III. FORMULATION OF NUMERICAL EQUATIONS

In this section, we describe the numerical scheme to solve the Grad–Shafranov equation (5) with the fixed boundary condition. We begin by describing the decomposition of the Grad–Shafranov equation before giving the computation of the metric parameters. We then describe the iteration scheme and boundary conditions. We conclude this section with the description of the numerical scheme to solve the equilibrium equations with the independent solution decomposition in the radial direction and the Fourier decomposition in the poloidal direction.

A. Decomposition of the Grad–Shafranov equation

In this subsection, we introduce the radial, poloidal, and toroidal coordinates and project the Grad–Shafranov equation onto this coordinate system. We then use Fourier decomposition to decompose the equations.

To solve the Grad–Shafranov equation, we introduce the coordinate system (ψ, θ, ϕ), with ψ labeling the radial grids and θ being the poloidal angle. The coordinates ψ and θ are general, only requiring that the Jacobian

$$\mathcal{J} = \frac{1}{\nabla\psi \times \nabla\theta \cdot \nabla\phi}$$

remains finite. In this coordinate system, one can obtain

$$\begin{aligned}\frac{1}{X^2}\nabla\chi &= \frac{1}{X^2}\frac{\partial\chi}{\partial\psi}\nabla\psi + \frac{1}{X^2}\frac{\partial\chi}{\partial\theta}\nabla\theta \\ &= A_1 \nabla\theta \times \nabla\phi + A_2 \nabla\phi \times \nabla\psi,\end{aligned}$$

where

$$A_1 = \frac{1}{X^2}\frac{\partial \chi}{\partial \psi}\mathcal{J}|\nabla\psi|^2 + \frac{1}{X^2}\frac{\partial \chi}{\partial \theta}\mathcal{J}\nabla\psi\cdot\nabla\theta,$$
$$A_2 = \frac{1}{X^2}\frac{\partial \chi}{\partial \psi}\mathcal{J}\nabla\psi\cdot\nabla\theta + \frac{1}{X^2}\frac{\partial \chi}{\partial \theta}\mathcal{J}|\nabla\theta|^2. \quad (6)$$

Therefore, one has

$$\mathcal{J}\nabla\cdot\frac{\nabla\chi}{X^2} = \frac{\partial A_1}{\partial\psi} + \frac{\partial A_2}{\partial\theta}$$
$$= \frac{\partial A_1}{\partial\psi} + i\mathcal{M}\left(\frac{1}{X^2}\mathcal{J}\nabla\psi\cdot\nabla\theta\right)\frac{\partial\chi}{\partial\psi}$$
$$- \mathcal{M}\left\{\frac{1}{X^2}\mathcal{J}|\nabla\theta|^2\right\}\mathcal{M}\chi.$$

Here, we have denoted $\frac{\partial}{\partial\theta} = i\mathcal{M}$ with $\mathcal{M}$ being the matrix specifying the poloidal Fourier numbers, since the Fourier decomposition with θ will be introduced later on. Using this decomposition, the Grad–Schafranov equation (5) can be reduced to the following set of first order differential equations:

$$\frac{\partial\chi}{\partial\psi} = F_{11}\chi + F_{12}A_1, \quad (7)$$
$$\frac{\partial A_1}{\partial\psi} = F_{21}\chi + F_{22}A_1 + S, \quad (8)$$

where

$$F_{11}(\psi,\theta) = -i\frac{\nabla\psi\cdot\nabla\theta}{|\nabla\psi|^2}\mathcal{M},$$
$$F_{12}(\psi,\theta) = \frac{X^2}{\mathcal{J}|\nabla\psi|^2},$$
$$F_{21}(\psi,\theta) = \mathcal{M}\frac{1}{X^2}\mathcal{J}|\nabla\theta|^2\mathcal{M} - \mathcal{M}\frac{1}{X^2}\mathcal{J}\nabla\psi\cdot\nabla\theta\frac{\nabla\psi\cdot\nabla\theta}{|\nabla\psi|^2}\mathcal{M},$$
$$F_{22}(\psi,\theta) = -i\mathcal{M}\frac{1}{X^2}\mathcal{J}\nabla\psi\cdot\nabla\theta\frac{X^2}{\mathcal{J}|\nabla\psi|^2},$$
$$S(\psi,\theta) = \frac{1}{\chi_a}\left(-\mathcal{J}p'_{\hat{\chi}} - \frac{\mathcal{J}}{X^2}gg'_{\hat{\chi}}\right).$$

To solve the set of equilibrium equations, Eqs. (7) and (8), the following Fourier decompositions are introduced:

$$\begin{pmatrix}\chi\\ A_1\\ S\end{pmatrix} = \frac{1}{\sqrt{2\pi}}\sum_{m=-M_{\max}}^{M_{\max}}\begin{pmatrix}\chi_m\\ A_{1m}\\ S_m\end{pmatrix}\exp\{im\theta\}. \quad (9)$$

Here, M_{max} represents the maximum Fourier component to be used. Introducing the Fourier decomposition in Eq. (9), the set of Eqs. (7) and (8) becomes the set of matrix equations with the coefficients becoming the matrices as defined as follows:

$$\mathcal{F}_{ij,mm'} = \frac{1}{2\pi}\int_{-\pi}^{\pi}d\theta F_{ij}(\psi,\theta)e^{i(m'-m)\theta}.$$

Note that for the non-up-down symmetric system, the Fourier components are complex. The set of matrix equations in complex can be written as

$$\frac{\partial}{\partial\psi}\begin{pmatrix}\chi\\ A_1\end{pmatrix} - \begin{pmatrix}\mathcal{F}_{11} & \mathcal{F}_{12}\\ \mathcal{F}_{21} & \mathcal{F}_{22}\end{pmatrix}\begin{pmatrix}\chi\\ A_1\end{pmatrix} = \begin{pmatrix}0\\ S\end{pmatrix}. \quad (10)$$

Here, χ and A_1 are the vectors in the Fourier space with the total components $M = M_{\max} + 1$ and $\mathcal{F}_{ij}$ are the matrices with dimension $M\times M$. Therefore, Eq. (10) represents a set of $2M$ differential equations. The matrix equation, Eq. (10), can be rewritten concisely as follows:

$$\frac{\partial\mathbf{u}}{\partial\psi} - \mathcal{F}\mathbf{u} = \mathbf{s}(\mathbf{u}), \quad (11)$$

where the source term $\mathbf{s}$ is usually a nonlinear function of $\mathbf{u}$.

B. Computation of the metric parameters

In this subsection, we describe how the matrix $\mathcal{F}$ is computed in the ATEQ code. This is related to the determination of the metric parameters, such as $|\nabla\psi|^2$, $\nabla\psi\cdot\nabla\theta$, etc.

As in the PEST code,[38] we introduce the polar coordinates to compute the following metric parameters:

$$S_r = x^2 + z^2,$$
$$\Theta = \arctan(z/x),$$

where $x = X - X_0$ and $z = Z$ with X_0 being the major radius at the magnetic axis locating at $Z = 0$. Since $X(\psi,\theta)$ and $Z(\psi,\theta)$ are given when introducing the (ψ,θ) grids, one can also determine $S_r(\psi,\theta)$ and $\Theta(\psi,\theta)$. Consequently, one can derive both

$$\frac{\partial(S_r,\Theta)}{\partial(X,Z)} \equiv \begin{pmatrix}\frac{\partial S_r}{\partial X} & \frac{\partial S_r}{\partial Z}\\ \frac{\partial\Theta}{\partial X} & \frac{\partial\Theta}{\partial Z}\end{pmatrix} \quad \text{and} \quad \frac{\partial(S_r,\Theta)}{\partial(\psi,\theta)} \equiv \begin{pmatrix}\frac{\partial S_r}{\partial\psi} & \frac{\partial S_r}{\partial\theta}\\ \frac{\partial\Theta}{\partial\psi} & \frac{\partial\Theta}{\partial\theta}\end{pmatrix}.$$

Using these results, one can compute the metric parameters in the (S_r,Θ,ϕ) coordinate system.

We first work on the Jacobian $\mathcal{J}$. Note that

$$\mathcal{J} = \frac{1}{\nabla\psi\times\nabla\theta\cdot\nabla\phi}$$
$$= \frac{X}{\left|\frac{\partial(\psi,\theta)}{\partial(X,Z)}\right|} = X\left|\frac{\partial(X,Z)}{\partial(\psi,\theta)}\right|.$$

Note further that

$$\frac{\partial(S_r,\Theta)}{\partial(\psi,\theta)} = \frac{\partial(S_r,\Theta)}{\partial(X,Z)}\frac{\partial(X,Z)}{\partial(\psi,\theta)} \quad \text{and} \quad \left|\frac{\partial(S_r,\Theta)}{\partial(X,Z)}\right| = 2.$$

One obtains the Jacobian expression in the polar coordinates as follows:

$$\mathcal{J} = X\left|\frac{\partial(X,Z)}{\partial(\psi,\theta)}\right| = \frac{X}{2}\left|\frac{\partial(S_r,\Theta)}{\partial(\psi,\theta)}\right|.$$

Next, we work on other metric parameters. By straightforward reduction, one can obtain

$$\frac{\partial(\psi,\theta)}{\partial(X,Z)} = \frac{\partial(\psi,\theta)}{\partial(S,\Theta)}\frac{\partial(S,\Theta)}{\partial(X,Z)} = \left(\frac{\partial(S,\Theta)}{\partial(\psi,\theta)}\right)^{-1}\frac{\partial(S,\Theta)}{\partial(X,Z)} = \frac{X}{\mathcal{J}}\begin{pmatrix} z\frac{\partial S_r}{\partial\theta}\Big/2S_r + x\frac{\partial\Theta}{\partial\theta} & -x\frac{\partial S_r}{\partial\theta}\Big/2S_r + z\frac{\partial\Theta}{\partial\theta} \\ -z\frac{\partial S_r}{\partial\psi}\Big/2S_r - x\frac{\partial\Theta}{\partial\psi} & x\frac{\partial S_r}{\partial\psi}\Big/2S_r - z\frac{\partial\Theta}{\partial\psi} \end{pmatrix},$$

where it has been noted that

$$\begin{pmatrix} \frac{\partial S_r}{\partial\psi} & \frac{\partial S_r}{\partial\theta} \\ \frac{\partial\Theta}{\partial\psi} & \frac{\partial\Theta}{\partial\theta} \end{pmatrix}^{-1} = \frac{X}{2\mathcal{J}}\begin{pmatrix} \frac{\partial\Theta}{\partial\theta} & -\frac{\partial S_r}{\partial\theta} \\ -\frac{\partial\Theta}{\partial\psi} & \frac{\partial S_r}{\partial\psi} \end{pmatrix}.$$

Noting further that

$$|\nabla\psi|^2 = \frac{\partial\psi}{\partial X}\frac{\partial\psi}{\partial X} + \frac{\partial\psi}{\partial Z}\frac{\partial\psi}{\partial Z},$$
$$|\nabla\theta|^2 = \frac{\partial\theta}{\partial X}\frac{\partial\theta}{\partial X} + \frac{\partial\theta}{\partial Z}\frac{\partial\theta}{\partial Z},$$
$$\nabla\psi\cdot\nabla\theta = \frac{\partial\psi}{\partial X}\frac{\partial\theta}{\partial X} + \frac{\partial\psi}{\partial Z}\frac{\partial\theta}{\partial Z},$$

one obtains

$$|\nabla\psi|^2 = \left(\frac{X}{\mathcal{J}}\right)^2\left[\frac{1}{4S_r}\left(\frac{\partial S_r}{\partial\theta}\right)^2 + S_r\left(\frac{\partial\Theta}{\partial\theta}\right)^2\right],$$
$$|\nabla\theta|^2 = \left(\frac{X}{\mathcal{J}}\right)^2\left[\frac{1}{4S_r}\left(\frac{\partial S_r}{\partial\psi}\right)^2 + S_r\left(\frac{\partial\Theta}{\partial\psi}\right)^2\right],$$
$$\nabla\psi\cdot\nabla\theta = \left(\frac{X}{\mathcal{J}}\right)^2\left[-\frac{1}{4S_r}\frac{\partial S_r}{\partial\psi}\frac{\partial S_r}{\partial\theta} - S_r\frac{\partial\Theta}{\partial\psi}\frac{\partial\Theta}{\partial\theta}\right].$$

Using the toroidal symmetry property, we can also find that

$$|\nabla\phi|^2 = \frac{1}{X^2}, \quad \nabla\psi\cdot\nabla\phi = 0, \quad \text{and} \quad \nabla\theta\cdot\nabla\phi = 0.$$

The expressions of metric parameters given above can be used to compute the matrix $\mathcal{F}$ and the vector $\mathbf{s}$ in Eq. (11).

C. Iteration scheme and boundary conditions

With the computation of metric parameters given in Sec. III B, we describe the iteration scheme to solve the Grad–Shafranov equation with proper boundary conditions.

Since Eq. (11) is nonlinear, an iteration process is necessary. One can follow the usual iteration scheme to get the converged solution

$$\frac{\partial\mathbf{u}^{(n+1)}}{\partial\psi} - \mathcal{F}\mathbf{u}^{(n+1)} = \mathbf{s}(\mathbf{u}^{(n)}). \tag{12}$$

Here, n denotes the iteration step.

Equation (12) is a set of inhomogeneous differential equations of first order. Its general solutions at step $n+1$ can be expressed as

$$\mathbf{u} = \sum_{k=1}^{2M} c_k\mathbf{u}^k + \mathbf{u}^s, \tag{13}$$

where c^k are the complex constants to be determined by the boundary conditions, $\mathbf{u}^k$ are the independent solutions to the homogeneous equations, and $\mathbf{u}^s$ is the specific solution to take into account the source term $\mathbf{s}$ on the right hand side of Eq. (12). For brevity, the step index n has been dropped.

Since the number of equations is $2M$, the solutions are completely determined by the M boundary conditions in complex at the magnetic axis and M boundary conditions in complex at plasma edge χ_a. The boundary conditions at plasma edge χ_a are specified by the given shape of the last closed flux surface in the fixed boundary value problem. The boundary conditions at the magnetic axis are just the requirement that the independent solutions are "small" in terms of the terminology of differential equation theory. The "large" solution causes the system energy to diverge, while the "small" solution is square-integrable with respect to the energy integral. Near the magnetic axis, the homogeneous part of the Grad–Shafranov equation can be approximated by the cylinder model. In this limit, the solutions are given as follows:[37]

$$\chi_m = a_m r^{|m|} + b_m r^{-|m|}, \quad \text{for } m \neq 0,$$
$$\chi_0 = a_0 + b_0\ln r, \quad \text{for } m = 0,$$

where r is the minor radius and a_m and b_m are constants. Therefore, the boundary conditions for small solutions are simply $b_m = 0$. This yields

$$\frac{d\chi_m/dr}{\chi_m} = |m|r^{|m|-1}, \quad \text{for } m \neq 0, \tag{14}$$
$$\frac{d\chi_0/dr}{\chi_0} = 0, \quad \text{for } m = 0. \tag{15}$$

The boundary conditions for A_{1m} can be obtained using the definition of A_1 in Eq. (6).

Note that the general solution to the set of differential equations is the summation of homogeneous solutions and specific solution and the boundary conditions are satisfied by the constants c_k tied to the homogeneous solutions. Therefore, the boundary conditions for specific solution are arbitrary.

D. Solution of equilibrium equations

The principle to solve Eq. (12) is laid out in Sec. III C. The actual implementation is more complicated. One needs to divide multiple regions in the radial direction and then match the solutions in the individual regions to get the global solution. In this subsection, we will outline the actual numerical process in the ATEQ code to solve the Grad–Shafranov equation.

The M boundary conditions at the magnetic axis in Eqs. (14) and (15) can be used to eliminate M independent solutions by shooting outwardly with the boundary conditions at the axis as the initial conditions. There are only M independent solutions $\mathbf{u}^k$ left as a result. In principle, the remaining M constants c_k can be determined by the other M boundary conditions at the plasma edge, while the specific solution $\mathbf{u}^s$ can be also determined by the numerical shooting with the boundary condition at the magnetic axis $\mathbf{u}^s(0) = 0$. The procedure

looks straightforward. However, this straightforward procedure to shoot all the way from the axis to the edge usually does not work due to the numerical pollution of large solutions. One has to divide the whole region into multiple regions and then match the solutions in the individual regions to get the global solutions. A similar numerical scheme has been successfully used in the MHD stability code: AEGIS.[35]

Suppose there are L regions with their boundaries labeled, respectively, as ψ_l $(l = 0, 1, 2, \ldots, L)$, where ψ_0 and ψ_L represent, respectively, the magnetic axis and the last closed flux surface. For the first region, one can shoot from ψ_0 with the boundary conditions at the magnetic axis to get M independent solutions. The M independent solutions at the other end ψ_1 are used the construct the independent solution matrix: ${}^1\mathcal{U}^{2M\times M} = (\mathbf{u}^1, \ldots, \mathbf{u}^M)_{\psi=\psi_1}$. Here, the left superscript indicates the region and the right superscript $2M \times M$ represents "the number of Fourier components" × "the number of independent solutions." For the last region, one can shoot inwardly with the boundary conditions at ψ_L, i.e., by specifying χ at the edge, to get M independent solutions. Likewise, the M independent solutions can be used to form the independent solution matrix at the other end ψ_{L-1}: ${}^L\mathcal{U}^{M\times 2M}_{inward}$. Here, the subscript "inward" has been introduced to indicate the shooting in the last region is made inwardly. For the internal regions $(l = 2, \ldots, L-1)$, there are $2M$ independent solutions in each region. To construct the independent solutions in these regions, for example, region l, one can obtain the $2M$ independent solutions by specifying the independent boundary conditions at the lower end ψ_{l-1} and shooting upwardly. The $2M$ independent boundary conditions at the lower end can be simply the $2M$ columns in the identity matrix: $\mathcal{I}^{2M\times 2M}$. The $2M$ independent solutions at the upper end of each internal region can be used to form the independent solution matrices: ${}^l\mathcal{U}^{2M\times 2M}$.

Similarly, one can construct the specific solution vectors. In difference from the homogeneous solutions, there is only one set of solutions with $2M$ elements ${}^l\mathbf{s}^{2M\times 1}$ in each region. They can be obtained by specifying the initial conditions at the lower end as the null vector $0^{2M\times 1}$ (vector with all elements being zero) and shooting to the upper end, except the last region. The inward shooting is carried out in the last region.

With the independent solution matrices and the specific solution vectors in each region obtained, one can match them to obtain the global solutions. There are $L-1$ regional interfaces, and on each interface, there are $2M$ matching conditions. Note that, since the boundary conditions at axis and plasma edge have been applied, only M constants in each of these two regions remain to be determined. They are represented in vectors of M rows: ${}^1\mathbf{c}^{M\times 1}$ and ${}^L\mathbf{c}^{M\times 1}$. In the internal regions, however, there are $2M$ constants in each region, which are denoted as ${}^l\mathbf{c}^{2M\times 1}$ $(l = 2, \ldots, L-1)$. Therefore, the $2M(L-1)$ matching conditions determine fully the constants ${}^l\mathbf{c}^{M_l}$, where $l = 1, 2, \ldots, L$ and $M_1 = M_L = M$, $M_l = 2M$ for $l \neq 1$ and L. The matching conditions can be expressed as follows:

$$\mathcal{Y}\begin{pmatrix} {}^1\mathbf{c}^{M\times 1} \\ {}^2\mathbf{c}^{2M\times 1} \\ \vdots \\ {}^{L-1}\mathbf{c}^{2M\times 1} \\ {}^L\mathbf{c}^{M\times 1} \end{pmatrix}^{2M(L-1)\times 1} = \begin{pmatrix} -{}^1\mathbf{s}^{2M\times 1} \\ -{}^2\mathbf{s}^{2M\times 1} \\ \vdots \\ -{}^{L-2}\mathbf{s}^{2M\times 1} \\ {}^L\mathbf{s}^{2M\times 1}_{inward} - {}^{L-1}\mathbf{s}^{2M\times 1} \end{pmatrix}^{2M(L-1)\times 1}, \tag{16}$$

where

$$\mathcal{Y} \equiv \begin{pmatrix} {}^1\mathcal{U}^{2M\times M} & -\mathcal{I}^{2M\times 2M} & 0^{2M\times 2M} & \cdots & 0^{2M\times M} & 0^{2M\times M} & 0^{2M\times M} \\ 0^{2M\times M} & {}^2\mathcal{U}^{2M\times 2M} & -\mathcal{I}^{2M\times 2M} & \cdots & 0^{2M\times M} & 0^{2M\times M} & 0^{2M\times M} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0^{2M\times M} & \cdots & 0^{2M\times 2M} & \cdots & {}^{L-2}\mathcal{U}^{2M\times 2M} & -{}^{L-1}\mathcal{I}^{2M\times 2M} & 0^{2M\times M} \\ 0^{2M\times M} & \cdots & 0^{2M\times 2M} & \cdots & 0^{2M\times M} & {}^{L-1}\mathcal{U}^{2M\times 2M} & -{}^L\mathcal{U}^{2M\times M}_{inward} \end{pmatrix}^{2M(L-1)\times 2M(L-1)}.$$

Matrix $\mathcal{Y}$ is a band matrix. By inverting it, one can obtain the solution of Eq. (16)

$$\begin{pmatrix} {}^1\mathbf{c}^{M\times 1} \\ {}^2\mathbf{c}^{2M\times 1} \\ \vdots \\ {}^{L-1}\mathbf{c}^{2M\times 1} \\ {}^L\mathbf{c}^{M\times 1} \end{pmatrix}^{2M(L-1)\times 1} = \mathcal{Y}^{-1}\begin{pmatrix} -{}^1\mathbf{s}^{2M\times 1} \\ -{}^2\mathbf{s}^{2M\times 1} \\ \vdots \\ -{}^{L-2}\mathbf{s}^{2M\times 1} \\ {}^L\mathbf{s}^{2M\times 1}_{inward} - {}^{L-1}\mathbf{s}^{2M\times 1} \end{pmatrix}^{2M(L-1)\times 1}. \tag{17}$$

With the constants obtained from Eq. (17), the solutions in each region are then simply

$${}^l\mathbf{u} = \sum_{k=1}^{M_l} {}^lc_k\,{}^l\mathbf{u}^k + {}^l\mathbf{u}^s, \quad (l = 1, \ldots, L). \tag{18}$$

These give the numerical scheme being implemented in the ATEQ code to solve the Grad–Shafranov equation.

IV. NUMERICAL PROCEDURE AND RESULTS

In this section, we describe how to implement the numerical scheme in Sec. III. This leads to the development of ATEQ code. The computational flow chart of ATEQ is given in Fig. 1. To be more specific to describe the computational flow, we use an ITER-like equilibrium as an example. The major radius 6.2 m, minor radius 2 m, elongation 1.78, triangularity 0.4, the vacuum magnetic field at the geometric center of plasma column is 6 T, the total current 15.9 MA, and the volume average beta value is 3.371%. Figure 2 shows the cross section with the "a" part showing the initial grid setup and the "b" part showing the magnetic surfaces computed by the ATEQ code.

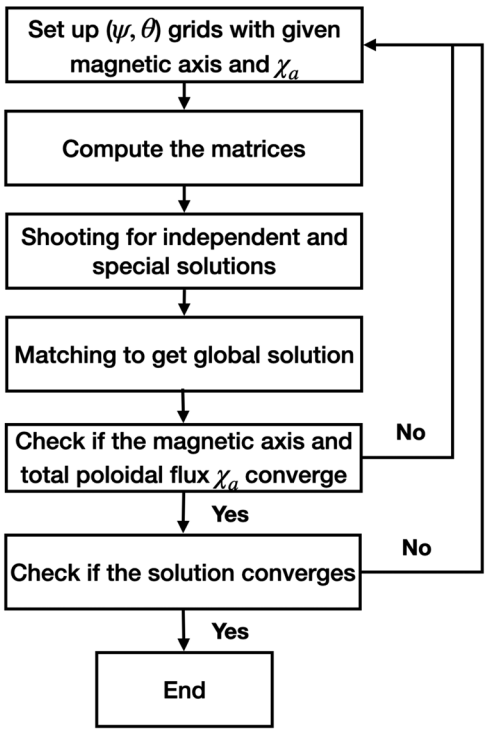

FIG. 1. Computational flow chart of the ATEQ code.

The case will also be used for the benchmark studies with the TOQ code. Further details about the equilibrium will be described there.

First, one needs to set up radial and poloidal grids (ψ, θ) as shown in Fig. 2(a). The grids are constructed to surround the magnetic axis (x_{axis}, z_{axis}). Because the magnetic axis is unknown beforehand, iteration is needed. The value of the previous step (n) is used to construct the grids to advance to the next step $n+1$. Following the iteration scheme in Eq. (12), the source term on the right hand side of Eq. (12) is evaluated by using the solution for poloidal flux $\mathbf{u}^{(n)}(\psi^{(n)}, \theta^{(n)})$ in the previous step. Note that the pressure and current profiles are prescribed by the normalized poloidal flux. The total poloidal flux χ_a needs also to be determined iteratively. At the first step, the quantities at the previous step are prescribed by initial guessing. The matrices $\mathcal{F}$ and $\mathbf{s}$ can then be computed with previous step grids as described in Sec. III B. Using the splines, the matrices $\mathcal{F}$ and $\mathbf{s}$ are made to be radially continuous functions.

Here, it is noted that the proper choice of initial (ψ, θ) grids can affect how many Fourier components are required. For the usual equilibria without X points included the choice is rather arbitrary, i.e., a wide range of grid choices can work well. For the equilibria with X points included proper choice of initial grids is important. In the ATEQ code, the initial grids are specified as follows. First, the grids with ellipticity k and triangularity δ are set up inside the specified plasma-vacuum boundary according to the formula

$$X = x_{axis} + r(\cos\theta - \delta \sin^2\theta), \tag{19}$$

$$Z = z_{axis} + kr\sin\theta, \tag{20}$$

with $r = [(X - x_{axis})^2 + (Z - z_{axis})^2]^{1/2}$. Here, k and δ can be polynomial functions of ψ. This means that one can adjust the ellipticity and triangularity from the axis to the outmost surface. In most cases, the linear dependence is sufficient. Next, the difference between the specified plasma boundary and the outmost surface given by Eqs. (19) and (20) is distributed radially. The distribution can be adjusted through an exponential multiplier of ψ. Also, the ψ grids can be packed near the axis and boundary. In our experience, with these flexibilities, roughly 100 Fourier sidebands are sufficient to get a good equilibrium solution with X points included. It is using this type of initial grid setting that the Solovév solution with X points to be described later is reproduced numerically. There is always a possibility to use the (ψ, θ) solution at step n for the grids at step $n+1$. Nevertheless, it can

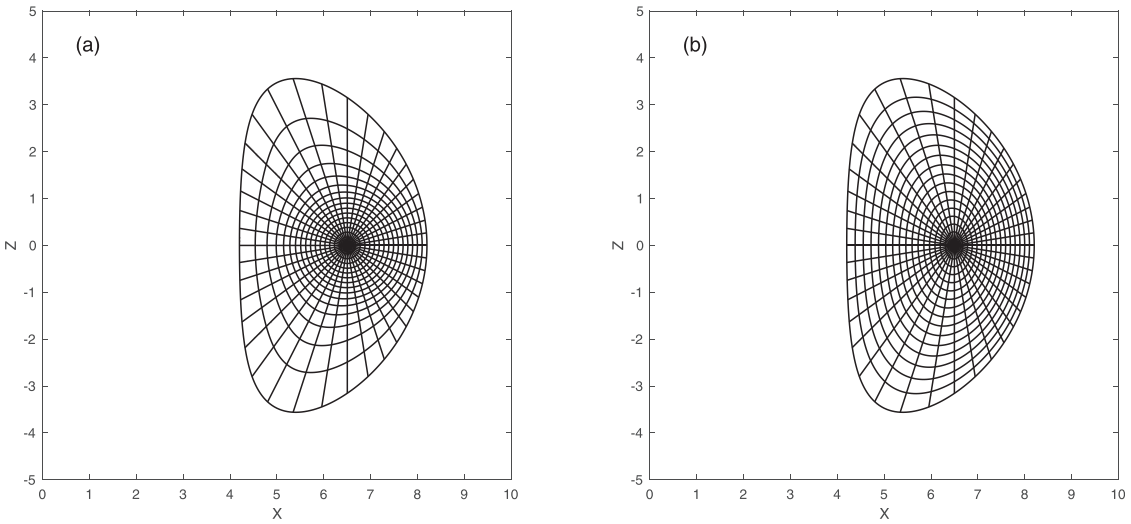

FIG. 2. Equilibrium results for the ITER-geometry-like case. (a) The initial (ψ, θ) grids. (b) The converged magnetic flux surfaces.

only be used if the solution at step n is sufficiently smooth and well-behaved.

Next, the whole radial domain is split into L regions. As described in Sec. III D, adaptive shooting is implemented to get the independent solution matrices in each region. By solving for ${}^{l}\mathbf{c}^{M_l}$ using Eq. (17), one can construct the global solution through Eq. (18). At this step, we first check if the magnetic axis (x_{axis}, z_{axis}) and total poloidal flux χ_a converge. Usually, total poloidal flux converges in one or two steps, using the following formula for prediction:

$$\chi_a^{(n+2)} = \sqrt{\chi_a^{(n+1)}\chi_a^{(n)}}.$$

Instead, to find the magnetic axis (x_{axis}, z_{axis}), one needs a few iterations. The code shoots outwardly from the assumed magnetic axis (x_{axis}, z_{axis}). After achieving the solution, the minimum of poloidal flux χ is determined. The location of this minimum is used as (x_{axis}, z_{axis}) for the shooting in the next step. This process is repeated until the starting (x_{axis}, z_{axis}) matches the location of the χ minimum computed to a required accuracy.

Figure 3 shows the iteration process for determining the magnetic axis to get the final solution in Fig. 2(b). The dashed curve in Fig. 3(a) shows the poloidal magnetic flux on the mid-plane computed with a guess value of magnetic axis in an earlier step. By searching for the minimum of the poloidal flux, a new magnetic axis location is found as shown by the solid vertical line. It is iterated until the magnetic axis coordinates (x_{axis}, z_{axis}) converge. Figure 3(b) shows the converged result. Because the iteration for magnetic axis and the iteration for the overall solution of poloidal magnetic flux are implemented simultaneously, the overall solution is often converged as the axis searching converges. The requirement for the number of poloidal Fourier components is also verified. The example shown in Fig. 2 uses 50 sidebands.

With the magnetic axis (x_{axis}, z_{axis}) and total poloidal flux χ_a being converged, one can further iterate to get the converged solution $\chi(\psi, \theta)$. With this solution, one can obtain the numerical solution for the poloidal magnetic flux $\chi(X, Z)$. The magnetic surfaces with $\chi(X, Z) = const$ are plotted in Fig. 2(b).

V. BENCHMARK STUDIES AND COMPARISON WITH EXISTING CODES

This section describes the benchmark studies. We begin with the analytical Solovév equilibrium with the X points included.[39] Next, the comparison with the existing numerical equilibrium code TOQ[16] is detailed. We also build a backward substitution module to double check the numerical equilibrium solutions. It simply substitutes the solution $\chi(X, Z)$ back into the Grad–Shafranov equation to check if the equation is satisfied to a sufficient accuracy.

The benchmark with the Solovév solution is not a trivial task. This is because the X-points are present in the Solovév equilibrium. The equilibrium computation with the X-points included is challenging because much more Fourier components are needed. The analytical Solovév solution is given as follows:

$$\chi_{Solovev} = \frac{1}{2}\left(bX_0^2 + c_0X^2\right)Z^2 + \frac{1}{8}(a - c_0)(X^2 - X_0^2)^2, \tag{21}$$

where the parameters are given as follows in the benchmark studies: $X_0 = 10$, $a = 1$, $b = -0.83$, and $c_0 = 0.92$. This solution corresponds to the pressure and poloidal current flux profiles given as follows:

$$-p' = a \quad \text{and} \quad -gg'/X_0^2 = b. \tag{22}$$

As pointed out in Ref. 39, the second-order solution in Eq. (21) is actually an exact solution of the Grad–Shafranov equation.

The numerical procedure for the benchmark studies to the analytical Solovév equilibrium solution is as follows. From the Solovév solution in Eq. (21), one can determine the last closed flux surface. The last closed flux surface is then used as the plasma boundary condition in the ATEQ code. The same pressure and poloidal current flux profiles as given in Eq. (22) are used to compute the numerical

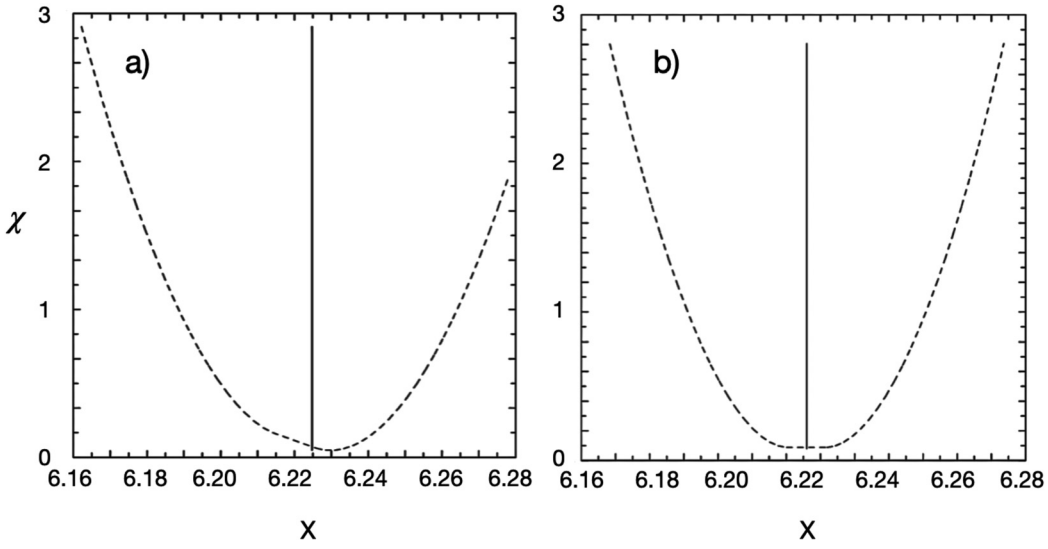

FIG. 3. The iteration process to determine the magnetic axis. (a) The initial guessing. (b) The converged result. Dashed curves indicate the poloidal flux and the vertical solid lines indicate the proposed magnetic axis at the respective iteration step.

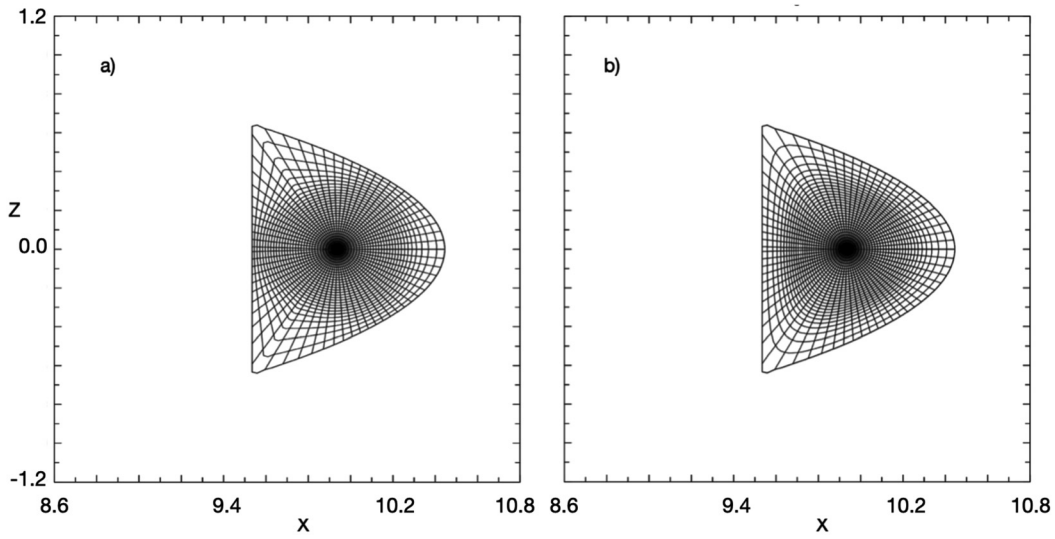

FIG. 4. Equilibrium results for Solovév solution. (a) The initial (ψ, θ) grids. (b) The converged magnetic flux surfaces.

solution $\chi(X, Z)$ with the ATEQ code. The solution is then compared with the analytical Solovév solution in Eq. (21).

Figure 4(a) gives the initial (ψ, θ) grids and Fig. 4(b) shows the converged magnetic flux surfaces computed by the ATEQ code. The number of Fourier sidebands is 102. The process just follows the chart given in Fig. 1. The numerical results agree well with the analytical solution in Eq. (21). To show the agreement, the poloidal magnetic flux at the mid-plane on the low field side both from the analytical solution in Eq. (21) (solid curve) and from the computational result by ATEQ (dashed curve) are plotted in Fig. 5. Two curves completely overlap, although the initial guessing as shown in Fig. 4(a) deviates dramatically from the actual solution in Fig. 4(b) in the ATEQ computation.

Comparisons with the existing equilibrium codes are also performed. Here, we describe a benchmark example with the TOQ equilibrium code.[16] A typical case is described as follows. A TOQ sample initiation file with *equiltype* =′ *ffprime*′ is taken. To be more specific to compare with TOQ, here the same numerical parameter notations as in the TOQ manual are used to describe the equilibrium parameters. The shape of boundary type is specified by *ishape* = 2, which is described as follows:

$$X = rzero + rmax * (\cos\theta - xshape * \sin^2\theta), \qquad Z = eshape * rmax * \sin\theta, \tag{23}$$

where the basic parameters are specified as follows: the major radius *rzero* = 6.2 m, the minor radius *rmax* = 2 m, the elongation *eshape* = 1.78, and the triangularity *xshape* = 0.4. This leads the equilibrium cross section to be given in Fig. 2.

The pressure gradient (p') and poloidal current flux parameter (gg') profiles are specified, respectively, by setting *modelp* = 3 and *modelf* = 1, which are described as follows:

$$p' = 1 - 0.4\hat{\chi} + 0.4\hat{\chi}^2 - \hat{\chi}^3, \tag{24}$$

$$gg' = 1 - \hat{\chi}. \tag{25}$$

Note here that the profiles are specified with the normalized poloidal magnetic flux $\hat{\chi}$, varying from 0 to 1 from the magnetic axis to plasma boundary. The tolerance is set to be $toleq = 10^{-5}$ in the TOQ iteration with successively increasing grid densities. Here, we have used the nonlinear pressure profile in Eq. (24), which is different from the TOQ sample initiation file, in order to avoid the linear profile case considered in the Solovév case. The pressure and poloidal current flux

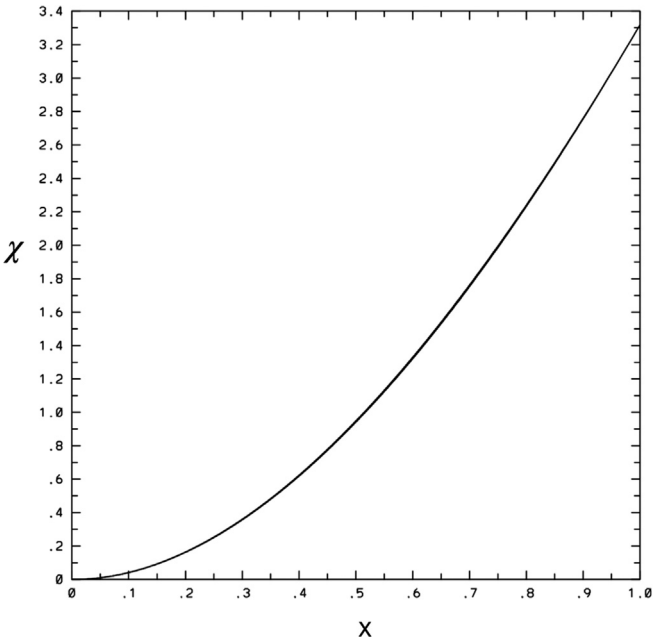

FIG. 5. The poloidal magnetic flux at mid-plane on the low field side both from the analytical Solovév solution in Eq. (21) (solid curve) and the ATEQ computational result (dashed curve). Two curves completely overlap.

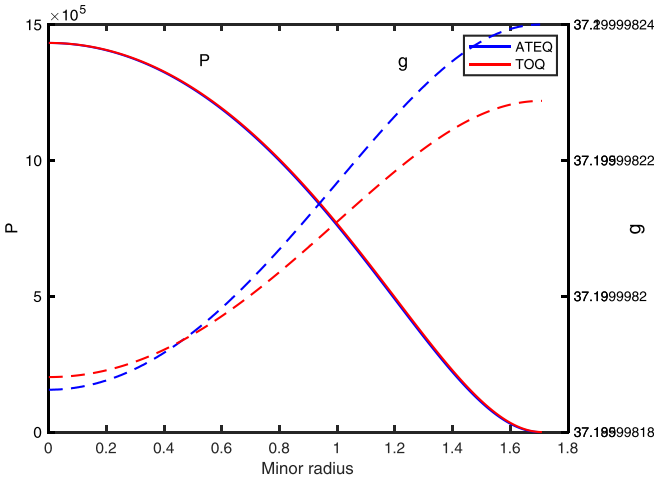

FIG. 6. The equilibrium pressure and poloidal current flux profiles vs the minor radius on the outer mid-plane for the benchmark case between TOQ (red) and ATEQ (blue).

profiles are given in Fig. 6 with respect to the minor radius on the outer vertical mid-plane. Although the p' and gg' are the same as specified in Eqs. (24) and (25) for TOQ and ATEQ codes, the pressure (p) and poloidal current flux (g) profiles can be slightly different since they are given in the minor radius, instead of the normalized poloidal flux. The slight difference of poloidal magnetic flux solution as discussed later can cause the difference. The volume average beta is 3.371%, the normalized beta is 2.54, and $l_i = 0.730$ in this equilibrium.

Figure 2(b) shows the equilibrium magnetic flux surfaces by the ATEQ code. The slight difference between TOQ and ATEQ results is not perceivable in the flux surface plot. Figure 7 is introduced to show the poloidal magnetic flux χ and the safety factor q vs the minor radius, which are computed, respectively, by the TOQ (red) and ATEQ (blue) codes. One can see that both χ and q solutions agree rather well. The slight difference results from the different accuracies for TOQ and ATEQ codes as discussed later on in the backward substitution check. Note that the red (TOQ) and blue (ATEQ) curves in Figs. 6 and 7 terminate roughly at the same minor radius. This indicates that the Shafranov shifts computed by the two codes agree.

In passing, it is pointed out that the region number L is about $30 - 40$ to recover the Solovév solution. For the case without X points, the required number L is less. It usually does not work if $L = 1$, i.e., shooting all the way from the axis to the plasma edge. Some Fourier components become extremely larger, while some others are very small. This feature makes the final matching matrix at the edge in poor condition. The multiple region matching solves the difficulty. Because the matrix size in ATEQ is determined by the number of regions, instead of the radial grid points, and the number of regions is much less than the grid points, adding some more regions does not cause many difficulties.

To further check the computation results, we implement the backward substitution check both for TOQ and ATEQ. In this checking procedure, the numerical solution χ is substituted back to the Grad–Shafronov equation, Eq. (5), to compute the relative errors at each grid point. Because the χ is determined, p and g become one dimensional. The solution (χ, θ) are used as the coordinates for checking. The five-point differential scheme is used to evaluate the derivatives. This check is done surface by surface. The relative error for each grid point is defined by the difference between the left and right values divided by the larger one between them. The surface-averaged relative errors are plotted in Fig. 8 vs the normalized magnetic flux. Because of the adaptive numerical scheme, high accuracy or low relative error is achieved by the ATEQ calculation. For TOQ computation, a very low tolerance $toleq = 10^{-5}$ has actually been imposed. The TOQ code does exit with the converged results. The convergence criterion in the TOQ code is based on the comparison between two consecutive steps,

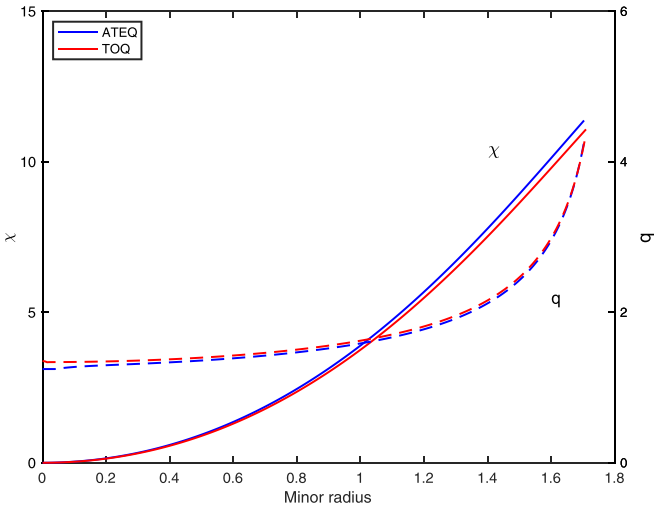

FIG. 7. The equilibrium poloidal magnetic flux χ and safety factor profiles vs the minor radius on the outer mid-plane for the benchmark case as computed by TOQ (red) and ATEQ (blue).

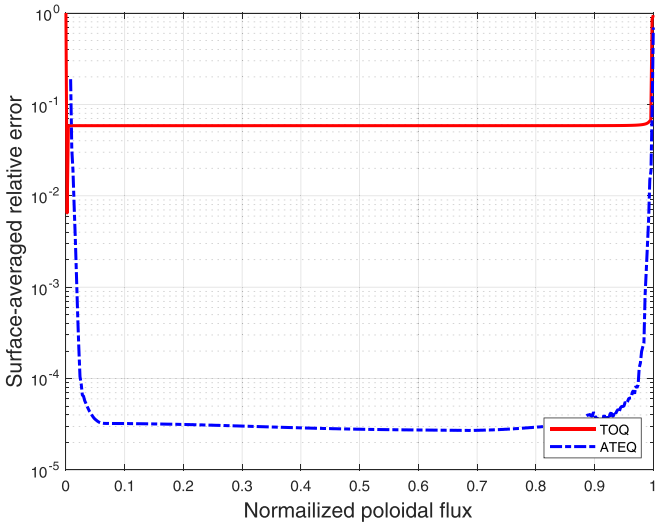

FIG. 8. The surface-averaged relative errors vs the normalized magnetic flux with the backward substitution check, respectively, for TOQ (red) and ATEQ (blue) numerical results.

instead of the backward substitution check as in the ATEQ code. This explains the larger surface-averaged relative error as compared to the ATEQ code in the backward substitution check. We especially want to emphasize that this does not necessarily imply that TOQ is not good, but only shows that different convergence criteria can yield different solutions. If TOQ used the backward substitution method to determine the convergence, TOQ could possibly also get good results. Also, the ATEQ code is based on an adaptive numerical scheme, and better convergence can be expected. The backward substitution check of ATEQ results further verifies its numerical procedure.

We also performed checks with other codes, for example, VMEC and EFIT. ATEQ achieves satisfactory results, generally giving better convergence in the backward substitution check. The backward substitution method thus confirms the validity of the ATEQ code.

VI. CONCLUSIONS AND DISCUSSION

We have presented a new, radially adaptive numerical scheme that solves the Grad–Shafranov equation for axisymmetric MHD equilibrium. This numerical scheme represents the solution through a sum in terms of independent solutions in the radial direction and Fourier decomposition in the poloidal direction. It computes the independent solutions using an adaptive shooting scheme together with the multi-region matching technique in the radial direction. The adaptive numerical scheme improves considerably the accuracy of the equilibrium solution. We named the implementation of this scheme the adaptive toroidal equilibrium code (ATEQ).

The decomposition with independent solutions effectively reduces the matrix size for solving the magnetohydrodynamic equilibrium problem, as compared with numerical schemes based on a fixed radial grid. The adaptive numerical scheme is expected to be especially helpful to deal with stiff equilibrium problems. Our results also indicate that the backward substitution method can be necessary to obtain a reliable equilibrium solution.

Let us here further discuss the unique features of the ATEQ numerical scheme. The numerical methods for solving the Grad–Shafranov equation ultimately reduce the problem to solve the matrix equations. The matrix size then matters and reducing the matrix size in discretizing the Grad–Shafranov equation is important. In the grid-based numerical schemes both in the radial and poloidal directions (finite difference or finite element), the size of the matrix is $N_r \times N_t$. Here, N_r and N_t are, respectively, the numbers of grids in the radial and poloidal directions. In the numerical scheme based on the poloidal Fourier decomposition, the matrix size is $N_r \times N_f$. Here, N_f is the number of the poloidal Fourier components. To achieve high accuracy, especially for tough problems related to the axis, X-point, or pedestal, etc., one has to increase the N_r and N_t (or N_f). Consequently, the size of the matrices becomes large and the matrices become hard to deal with numerically. In the ATEQ numerical scheme, the radial direction is split into L regions with each region addressed by the adaptive shooting of independent solutions. It reduces the radial N_r grid problem into a L region matching problem. This cuts down the $N_r \times N_t$ (or $N_r \times N_f$) matrix problem in the conventional numerical schemes into a $L \times N_{indep}$ problem in the ATEQ numerical scheme. Here, the number of regions L is about a few tens and N_{indep} is the number of independent solutions, which is of the same order as N_f. The reduction of the matrix size is by the factor L/N_r, which is about an order of magnitude, as compared with the conventional radially grid-based numerical schemes. Also, in this ATEQ numerical scheme, no matter how accuracy in the radial direction is imposed, the size of the matrices basically does not change. Such an improvement in the order of magnitude rarely happens. It, therefore, represents a significant development in this research.

The equilibrium problem is a little bit different from the stability one. If one uses the exact flux solution as the radial grids, only a single Fourier component for the magnetic flux χ is required because it is constant on the surfaces. Therefore, the required number of the Fourier components, N_f, in principle can be somewhat optimized by setting proper radial grids. Since the matrix size is reduced in the radial direction in ATEQ, one has more flexibility to increase the number of poloidal Fourier components if it is required. This is a distinct feature of ATEQ as compared to the conventional Fourier decomposition based codes. This improvement is useful.

It is realized in this field that a good numerical equilibrium solution near the axis and plasma boundary in the presence of the X points is critical. It is a challenging issue for decades. As cited in the introduction, several efforts have been made. Our work provides another possible solution. To directly compare with other codes to treat the X point equilibrium problem will be our next task. This may require close collaboration with other teams. Equilibrium codes often need certain specific procedure to execute them. Using the backward substitution method, we found that the equilibrium accuracy varies a lot even with the same code. That is why we are wary of doing code-to-code comparisons directly without the other party involved. Each code may have their own particular features. We have compared with TOQ since the example file is in the public domain. Even in this case, we have provided additional clarifications. However, one thing we can do is to compare with the Solovév analytical solution in the presence of X points. If not at all, rather few codes have been published with such a comparison as justification. This shows the capacity of ATEQ numerical scheme and code.

ACKNOWLEDGMENTS

This research is supported by the U.S. Department of Energy, Office of Fusion Energy Science under Grant No. DE-FG02-04ER54742 and the US-Japan Joint Institute for Fusion Theory (JIFT) collaboration program.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Linjin Zheng: Writing – original draft (equal); Writing – review and editing (equal). **M. T. Kotschenreuther:** Writing – original draft (equal); Writing – review and editing (equal). **F. L. Waelbroeck:** Writing – original draft (equal); Writing – review and editing (equal). **Y. Todo:** Writing – original draft (equal); Writing – review and editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are available within the article.

REFERENCES

[1]H. Grad and H. Rubin, in *Proceedings of 2nd International Conference on the Peaceful Uses of Atomic Energy* (United Nations, Geneva, 1958), Vol. 31, pp. 190–197.
[2]V. D. Shafranov, Zh. Eksp. Teor. Fiz. **33**, 710–722 (1957); Sov. Phys. JETP **6**, 545–554 (1958; available at http://jetp.ras.ru/cgi-bin/e/index/e/6/3/p545?a=list.
[3]R. J. Buttery, J. M. Park, J. T. McClenaghan, D. Weisberg, J. Canik, J. Ferron, A. Garofalo, C. T. Holcomb, J. Leuer, P. B. Snyder, and Atom Project Team, Nucl. Fusion **61**, 046028 (2021).
[4]P. B. Snyder, K. H. Burrell, H. R. Wilson, M. S. Chu, M. E. Fenstermacher, A. W. Leonard, R. A. Moyer, T. H. Osborne, M. Umansky, W. P. West, and X. Q. Xu, Nucl. Fusion **47**, 961–968 (2007).
[5]L. J. Zheng, M. T. Kotschenreuther, and J. W. Van Dam, Phys. Plasmas **17**, 056104 (2010).
[6]L. L. Lao, H. St. John, R. D. Stambaugh, A. G. Kellman, and W. Pfeiffer, "Reconstruction of current profile parameters and plasma shapes in tokamaks," Nucl. Fusion **25**, 1611 (1985).
[7]J. M. Moret, B. P. Duval, H. B. Le, S. Coda, F. Felici, and H. Reimerdes, Fusion Eng. Des. **91**, 1–15 (2015).
[8]E. Fable, C. Angioni, A. A. Ivanov, K. Lackner, O. Maj, S. Yu Medvedev, G. Pautasso, G. V. Pereverzev, W. Treutterer, and ASDEX Upgrade Team, Plasma Phys. Controlled Fusion **55**, 074007 (2013).
[9]Y. Huang, B.-J. Xiao, and Z.-P. Luo, Chin. Phys. B **26**, 085204 (2017).
[10]X. N. Yue, B. J. Xiao, Z. P. Luo, and Y. Guo, Plasma Phys. Controlled Fusion **55**, 085016 (2013).
[11]B. C. Lyons, C. Paz-Soldan, O. Meneghini, L. L. Lao, D. B. Weisberg, E. A. Belli, T. E. Evans, N. M. Ferraro, and P. B. Snyder, Phys. Plasmas **25**, 056111 (2018).
[12]R. Miller, M. Chu, R. Dominguez, and T. Ohkawa, Comments on Plasma Phys. Controlled Fusion **12**, 125–132 (1989).
[13]T. Takeda and S. Tokuda, J. Comput. Phys. **93**, 1–107 (1991).
[14]J. DeLucia, S. C. Jardin, and A. M. M. Todd, J. Comput. Phys. **37**, 183–204 (1980).
[15]S. P. Hirschman and J. C. Whitson, Phys. Fluids **26**, 3553–3568 (1983).
[16]Originally written by Bob Miller of General Atomics, see https://fusion.gat.com/THEORY/toq/overview.html for "the TOQ code description."
[17]J. A. Holmes, Y. K. M. Peng, and S. J. Lynch, J. Comput. Phys. **36**, 35–54 (1980).
[18]K. M. Ling and S. C. Jardin, J. Comput. Phys. **58**, 300–335 (1985).
[19]R. Gruber, R. Iacono, and F. Troyon, J. Comput. Phys. **73**, 168–182 (1987).
[20]H. Lütjens, A. Bondeson, and O. Sauter, Comput. Phys. Commun. **97**, 219–260 (1996).
[21]J. A. Crotinger, L. LoDestro, L. Don Pearlstein, A. Tarditi, T. A. Casper, and E. B. Hooper, "Corsica: A comprehensive magnetic-fusion devices simulation final report to the LDRD program," Report No. UCRL-ID-1262S4, 1997.
[22]G. T. A. Huysmans, J. P. Goedbloed, and W. Kerner, *Proc. CP90 Conf. Comput. Phys. Proc.* (World Scientific, 1991), pp. 371–376.
[23]B. Turkington, A. Lifschitz, A. Eydeland, and J. Spruck, J. Comput. Phys. **106**(2), 269–285 (1993).
[24]L. Greengard and J. Y. Lee, J. Comput. Phys. **125**, 415–424 (1996).
[25]G. O. Ludwig, Plasma Phys. Controlled Fusion **39**, 2021–2037 (1997).
[26]S. C. Jardin, J. Comput. Phys. **200**, 133–152 (2004).
[27]L. Guazzotto, R. Betti, J. Manickam, and S. Kaye, Phys. Plasmas **11**, 604–614 (2004).
[28]P. A. Gourdain, J. N. Leboeuf, and R. Y. Neches, J. Comput. Phys. **216**, 275–299 (2006).
[29]A. Pataki, A. J. Cerfon, J. P. Freidberg, L. Greengard, and M. O'Neil, J. Comput. Phys. **243**, 28–45 (2013).
[30]M. D. Boyer, D. J. Battaglia, D. Mueller, N. Eidietis, K. Erickson, J. Ferron, D. A. Gates, S. Gerhardt, R. Johnson, E. Kolemen, J. Menard *et al.*, Nucl. Fusion **58**, 036016 (2018).
[31]T. Sánchez-Vizuet and M. Solano, Comput. Phys. Commun. **235**, 120–132 (2019).
[32]R. F. Schmitt, L. Guazzotto, H. Strauss, G. Y. Park, and C. S. Chang, Phys. Plasmas **18**, 022502 (2011).
[33]H. Li and P. Zhu, Comput. Phys. Commun. **260**, 107264 (2021).
[34]X. Li, L. E. Zakharov, and V. V. Drozdov, Phys. Plasmas **21**, 012505 (2014).
[35]L. J. Zheng and M. T. Kotschenreuther, J. Comput. Phys. **211**, 748–766 (2006).
[36]L. J. Zheng, M. T. Kotschenreuther, and J. W. Van Dam, J. Comput. Phys. **229**, 3605–3622 (2010).
[37]J. P. Freidberg, *Ideal Magnetohydrodynamics* (Plenum Press, New York, 1987).
[38]R. C. Grimm, J. M. Greene, and J. L. Johnson, *Methods of Computational Physics* (Academic Press, New York/London, 1976), Vol. 9, pp. 253–280.
[39]L. S. Solovév, Sov. Phys. JETP **26**, 400–407 (1968); available at http://jetp.ras.ru/cgi-bin/e/index/e/26/2/p400?a=list.