Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A MATLAB-Based Graphical User Interface for Finite Element Solution of the Grad--Shafranov Equation

Yahya Sadeghi

ysadeghi@aeoi.org.ir

Nuclear science and technology research institute (NSTRI)

Samaneh Yarmahmoodi

Nuclear science and technology research institute (NSTRI)

Research Article

Keywords: Graphical User Interface, Grad-Shafranov Equation, Finite Element Method, Ideal Magnetohydrodynamics, Tokamak Equilibrium, Plasma Physics Education

Posted Date: September 28th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10982636/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

A MATLAB-Based Graphical User Interface for Finite Element Solution of the Grad–Shafranov Equation

Yahya Sadeghi[1*] and Samaneh Yarmahmoodi[2]

[1,2]Plasma and nuclear fusion research school, Nuclear science and technology research institute (NSTRI), 14155-1339, Tehran, Iran.

*Corresponding author(s). E-mail(s): ysadeghi@aeoi.org.ir;

Abstract

This paper presents MHDEquilSolver, an interactive MATLAB-based graphical user interface (GUI) for solving the Grad–Shafranov equation, the fundamental nonlinear elliptic partial differential equation governing ideal magnetohydrodynamic (MHD) equilibrium in axisymmetric toroidal plasmas. The solver uses the finite element method with mesh refinement to compute the poloidal flux function $\boldsymbol{\psi(R, Z)}$ and associated equilibrium quantities. The GUI allows real-time adjustment of key plasma parameters, including geometric properties ($\boldsymbol{R_0}$, $\boldsymbol{a}$, $\boldsymbol{\kappa}$, $\boldsymbol{\delta}$) and profile coefficients defining the toroidal field function $\boldsymbol{F(\psi) = f_0 + f_1\psi + f_2\psi^2}$ and plasma pressure $\boldsymbol{p(\psi) = p_1\psi + p_2\psi^2}$. A sensitivity analysis quantifies the dependence of the poloidal magnetic flux, magnetic field components, and toroidal current density on each input parameter. The solver includes error-handling mechanisms with fallback strategies for mesh generation, ensuring numerical stability across a wide parameter space. We detail the numerical framework, solution algorithms, and post-processing routines to ensure reproducibility. This tool serves both as an educational platform for teaching MHD equilibrium principles and as a rapid prototyping environment for tokamak plasma analysis.

Keywords: Graphical User Interface, Grad-Shafranov Equation, Finite Element Method, Ideal Magnetohydrodynamics, Tokamak Equilibrium, Plasma Physics Education

1 Introduction

The Grad–Shafranov equation is a cornerstone of ideal MHD, governing the equilibrium state of an axisymmetric plasma confined by a magnetic field in toroidal devices such as tokamaks and stellarators [1, 2]. This nonlinear elliptic partial differential equation expresses a force balance among three competing effects: the pressure gradient ∇p, the Lorentz force $\mathbf{J} \times \mathbf{B}$, and magnetic tension from curved field lines [3, 4]. In toroidal geometry, curvature drives an outward force analogous to centrifugal force in a rotating fluid, requiring an inward magnetic force for equilibrium—this is the origin of the Shafranov shift [5, 6]. The equation couples the poloidal flux function ψ to the plasma pressure $p(\psi)$ and the toroidal field function $F(\psi) = RB_\phi$, where ψ labels magnetic surfaces via $\mathbf{B} \cdot \nabla\psi = 0$ [30]. The safety factor $q(\psi)$, which characterizes the helical pitch of field lines, determines stability against ideal MHD modes through the Kruskal-Shafranov criterion $q > 1$ for kink stability [7, 8].

Over the past five decades, equilibrium codes such as EFIT [9] and CHEASE [10] have been developed to solve the Grad–Shafranov equation. However, these codes typically require significant expertise to operate, rely on legacy Fortran/C++ codebases with complex build systems, lack real-time interactivity for educational exploration or rapid parameter scans, and offer limited extensibility for students or researchers new to the field [11–13].

This work introduces MHDEquilSolver, an interactive GUI that addresses these gaps with the following features: (i) a modular finite-element solver covering geometry generation through post-processing; (ii) real-time parameter modification with visual feedback; (iii) built-in sensitivity analysis; (iv) multiple boundary shapes (circular, elliptical, D-shaped); (v) robust error handling; and (vi) utility for education and rapid prototyping. MATLAB offers distinct advantages for this application: built-in PDE solving capabilities, integrated visualization without external libraries, and an intuitive callback-based programming model for GUI development. The primary contributions of this paper are threefold. First, we provide a detailed algorithmic description of the solver, from geometry generation and mesh adaptation to post-processing, including specific implementation insights. Second, we present a systematic sensitivity analysis quantifying how each user-defined parameter influences the equilibrium solution and derived quantities [31]. Third, we document the user-friendly interface and visualization capabilities, which update all six subplots instantaneously upon parameter modification [32].

2 Mathematical Formulation

2.1 Equilibrium in Toroidal Geometry

In axisymmetric toroidal coordinates (R, ϕ, Z), where R is the distance from the symmetry axis, ϕ the toroidal angle, and Z the vertical coordinate, the magnetic field is expressed via the poloidal flux function $\psi(R, Z)$ [14, 15]:

$$\mathbf{B} = \frac{1}{R}\nabla\psi \times \hat{\mathbf{e}}_\phi + \frac{F(\psi)}{R}\hat{\mathbf{e}}_\phi, \tag{1}$$

where $\hat{\mathbf{e}}_\phi$ is the toroidal unit vector. The first term represents the poloidal field, with flux surfaces nested about the magnetic axis. The second term represents the toroidal field, whose $1/R$ dependence arises from Ampère's law and conservation of toroidal current [16]. This representation satisfies $\nabla \cdot \mathbf{B} = 0$.

The ideal MHD equilibrium condition $\mathbf{J} \times \mathbf{B} = \nabla p$, combined with Ampère's law $\mathbf{J} = \mu_0^{-1} \nabla \times \mathbf{B}$, yields the Grad–Shafranov equation:

$$\Delta^* \psi = -\mu_0 R^2 \frac{dp}{d\psi} - \frac{1}{2} \frac{dF^2}{d\psi}, \tag{2}$$

with $\mu_0 = 4\pi \times 10^{-7}$ H m^{-1} the vacuum permeability and Δ^* the elliptic Grad–Shafranov operator. The curvature term $-\frac{1}{R}\frac{\partial \psi}{\partial R}$ vanishes in the cylindrical limit $R \to \infty$, where $\Delta^* \to \nabla^2$ [17].

The right-hand side of Eq. (2) contains two physical contributions. The term $-\mu_0 R^2 dp/d\psi$ represents the diamagnetic current driven by pressure gradients, with the R^2 weighting reflecting the increased significance of pressure forces on the outboard side where curvature is weaker. The term $-\frac{1}{2} dF^2/d\psi$ represents the current associated with toroidal field gradients, implying the presence of poloidal currents [7]. The toroidal current density is then:

$$j_\phi = R \frac{dp}{d\psi} + \frac{1}{\mu_0 R} \frac{dF^2}{d\psi}, \tag{3}$$

which is exact within the ideal MHD framework.

2.2 Profile Parameterization

To maintain tractability while capturing essential physics, the code implements quadratic polynomial forms for $p(\psi)$ and $F(\psi)$:

$$\begin{aligned} p(\psi) &= p_0 + p_1 \psi + p_2 \psi^2, \quad & \frac{dp}{d\psi} &= p_1 + 2p_2 \psi, \\ F(\psi) &= f_0 + f_1 \psi + f_2 \psi^2, \quad & \frac{dF}{d\psi} &= f_1 + 2f_2 \psi. \end{aligned} \tag{4}$$

The coefficients and their physical interpretations are summarized in Tables 1 and 2. This quadratic parameterization provides sufficient flexibility for experimentally relevant profiles while maintaining computational efficiency.

For the pressure profile, p_1 determines the linear pressure gradient near the magnetic axis, while p_2 controls the curvature. In general, positive p_2 values can produce peaked pressure profiles characteristic of H-mode plasmas, while negative values yield broader profiles typical of L-mode operation. The total plasma beta scales approximately as [21]:

$$\beta \sim \frac{p(0)}{B_\phi^2(0)/(2\mu_0)} \sim \frac{p_1 \psi_{\max} + p_2 \psi_{\max}^2}{f_0^2/R_0^2}. \tag{5}$$

Table 1 Coefficients of the Pressure Profile $p(\psi)$

Coefficient	Default Value	Physical Interpretation
p_0	0.0	Boundary pressure (enforcing zero at plasma edge)
p_1	0.5	Linear pressure gradient; determines central pressure
p_2	-0.3	Quadratic curvature; positive values yield peaked profiles, negative values yield broad profiles

Table 2 Coefficients of the Toroidal Field Function $F(\psi)$

Coefficient	Default Value	Physical Interpretation
f_0	5.0	Base toroidal field at magnetic axis; $B_\phi(0) = f_0/R_0$
f_1	0.0	Linear shear; paramagnetic behavior for positive values
f_2	-0.1	Quadratic shear; controls current profile peaking

2.3 Plasma Boundary Geometry

The last closed flux surface (LCFS) is parameterized by the poloidal angle $\theta \in [0, 2\pi)$. The code supports three axisymmetric plasma shapes of increasing complexity [9].

For circular cross-sections:

$$R(\theta) = R_0 + a\cos\theta, \quad Z(\theta) = a\sin\theta, \tag{6}$$

where R_0 is the major radius and a the minor radius. The inverse aspect ratio $\epsilon = a/R_0$ measures toroidicity. In the limit $\epsilon \to 0$, the equilibrium becomes cylindrical and the Grad–Shafranov equation reduces to a one-dimensional problem [4].

For elongated cross-sections:

$$R(\theta) = R_0 + a\cos\theta, \quad Z(\theta) = \kappa a\sin\theta, \tag{7}$$

where $\kappa > 1$ denotes elongation [5]. Elongation increases the plasma volume for fixed R_0 and a, raising the fusion power density. However, highly elongated plasmas ($\kappa \gtrsim 2$) are susceptible to vertical instabilities, requiring active feedback control [30].

For D-shaped configurations, typical of modern tokamaks such as DIII-D and JET, triangularity δ is incorporated as [23]:

$$R(\theta) = R_0 + a\cos(\theta + \delta\sin\theta), \quad Z(\theta) = \kappa a\sin\theta. \tag{8}$$

Triangularity measures the indentation of the plasma on the outboard side. Positive triangularity ($\delta > 0$) is associated with improved edge stability and higher H-mode pedestal pressures [42]. The derivative $dR/d\theta = -a\sin(\theta + \delta\sin\theta)(1 + \delta\cos\theta)$ can vanish, causing the mapping to fold if δ is too large; practical values are limited to $\delta < 0.8$ to ensure well-posed geometry [32].

2.4 Finite Element Discretization

Applying Galerkin's method to Eq. (2), we multiply by a test function $v \in H_0^1(\Omega)$ and integrate:

$$\int_\Omega v\Delta^*\psi \, dA = \int_\Omega vS(\psi) \, dA, \tag{9}$$

where $S(\psi) = -\mu_0 R^2 dp/d\psi - \frac{1}{2} dF^2/d\psi$. Integration by parts yields the weak form:

$$\int_\Omega \left(\nabla v \cdot \nabla\psi - \frac{1}{R}\frac{\partial v}{\partial R}\psi \right) dA = \int_\Omega vS(\psi) \, dA, \tag{10}$$

where the boundary term vanishes due to the Dirichlet condition $v = 0$ on $\partial\Omega$ [24, 25]. This weak form requires only first derivatives. The finite element approximation employs linear triangular elements with basis functions ϕ_i satisfying $\phi_i(N_j) = \delta_{ij}$. The solution is approximated as:

$$\psi(R, Z) \approx \sum_{j=1}^{N} \psi_j \phi_j(R, Z), \tag{11}$$

where N is the number of nodes.

3 Numerical Implementation

3.1 Mesh Generation and Refinement

Mesh generation follows a multi-step process. First, a geometry description is constructed encoding the boundaries as line segments and circular arcs; for D-shaped boundaries, the curve is discretized into approximately 200 points connected by straight segments. The domain is then decomposed into minimal regions, a crucial step because the MATLAB PDE Toolbox meshing algorithm requires a decomposed geometry [29].

An initial triangular mesh is generated with a maximum edge length constraint $H_{\max} = a/5$. This choice balances resolution and computational cost; the typical number of elements scales as

$$N_{\text{elem}} \approx 4\pi\kappa \left(\frac{R_0}{a} \right) \left(\frac{a}{H_{\max}} \right)^2 \approx 300\kappa \frac{R_0}{a}. \tag{12}$$

For typical parameters ($R_0/a = 3$, $\kappa = 1.5$), this yields approximately 1350 elements, sufficient for moderate accuracy without excessive computation time.

refinement is then performed for a user-specified number of steps using the longest-edge bisection algorithm, which splits triangles along their longest edge and propagates refinements to maintain a conforming mesh [25]. A flowchart of the complete nonlinear solver with refinement is shown in Fig. 1.

Error handling includes fallback strategies such as self-intersecting boundary detection, resolution of acute angle issues, and coarsening of the initial mesh in problematic

cases. When primary mesh generation fails, the algorithm automatically invokes a coarser mesh with $H_{\max} = a/3$ and reduces the maximum refinement steps to $\min(n_{\text{ref}}, 1)$, ensuring graceful degradation rather than catastrophic failure [32]. In practice, fallback is rarely triggered for well-defined D-shaped boundaries but may occur for extreme triangularity ($\delta > 0.6$).

3.2 Boundary Conditions and Nonlinear Solver

A homogeneous Dirichlet boundary condition is imposed on the entire LCFS [4]:

$$\psi(R, Z) = 0 \quad \forall (R, Z) \in \partial\Omega, \tag{13}$$

encoding the physical requirement that the LCFS is a magnetic surface that does not intersect the wall [5].

The resulting nonlinear system is expressed as $\mathbf{G}(\boldsymbol{\psi}) = \mathbf{K}\boldsymbol{\psi} - \mathbf{b}(\boldsymbol{\psi}) = 0$. Solution is achieved using a damped Newton method with step-size control [29]. At each iteration k, the Newton update solves:

$$\mathbf{J}(\boldsymbol{\psi}^k)\delta\boldsymbol{\psi}^k = -\mathbf{G}(\boldsymbol{\psi}^k), \tag{14}$$

where the Jacobian is computed analytically as:

$$J_{ij} = \frac{\partial G_i}{\partial \psi_j} = K_{ij} - \frac{\partial b_i}{\partial \psi_j}, \qquad \frac{\partial b_i}{\partial \psi_j} = \int_\Omega \phi_i \frac{\partial S}{\partial \psi} \phi_j \, dA, \tag{15}$$

with

$$\frac{\partial S}{\partial \psi} = -\mu_0 R^2 (2p_2) - \frac{1}{2}\left(2(F')^2 + 2FF''\right), \tag{16}$$

where $F'' = 2f_2$ and $p'' = 2p_2$ are constants for the quadratic polynomials used. This analytical Jacobian provides quadratic convergence near the solution [18].

The iteration is initialized with $\psi = 0.5$, corresponding to a linear interpolation between $\psi_{\max}$ and 0. The solver uses a maximum iteration limit of 25, an damping parameter between 0.1 and 0.9 for stability and acceleration, and a convergence tolerance of 10^{-4} for visualization purposes; for accurate flux surfaces or q-profiles, a tolerance of 10^{-6} or lower is recommended [33]. The computational cost scales roughly as $\log(1/\text{tol})$, so tightening the tolerance by two orders of magnitude typically requires only 2–3 additional Newton iterations [28].

4 Graphical User Interface

4.1 Architecture and Implementation

The Magnetic Confinement Research Group is currently pursuing a dual-track initiative aimed at upgrading plasma diagnostic systems and redefining Alvand-U Tokamak's role as a national laboratory with integrated research and pedagogical mandates. Within this framework, the provision of specialized computational educational platforms is imperative to bridge experimental operations with academic training,

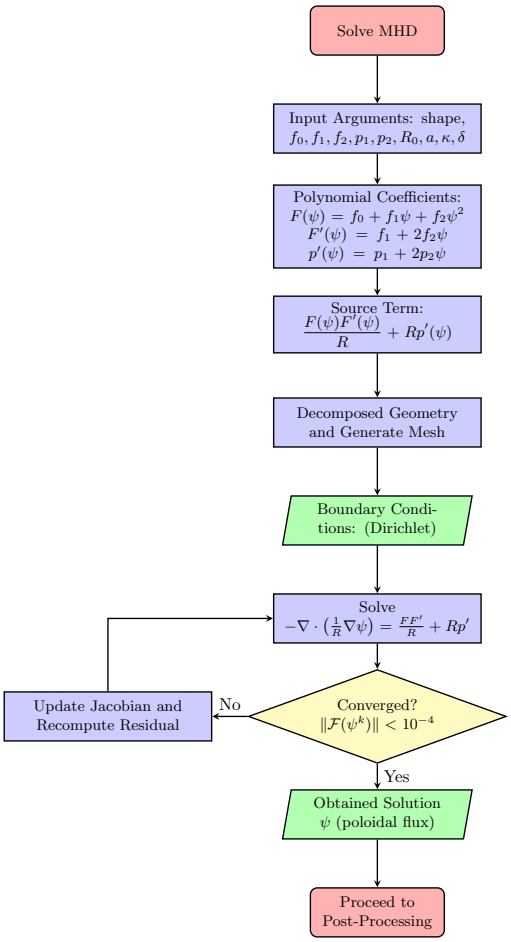


Fig. 1 Flowchart of the nonlinear Grad–Shafranov solver with mesh refinement

thereby ensuring the synergy between infrastructural modernization and workforce development [34–37]. The graphical user interface builds on previous research that solved the Grad–Shafranov equation under simplified geometries and fixed boundary conditions [38, 39]. In contrast, this GUI allows flexible adjustment of plasma parameters, including geometric properties and profile coefficients, and solves the equation for circular, elliptical, and D-shaped configurations. The interface outputs the dependence of the poloidal magnetic flux, magnetic fields, and toroidal current density on each input parameter (Fig. 2).

The main window comprises a control panel and an output display with six subplots showing poloidal flux contours, cross-section with mesh, magnetic field components, and pressure and current density profiles. The control panel includes editable text boxes for numerical input, a popup menu for shape selection, and a push button to execute the simulation. Each text box has a callback function that validates inputs,

enforces reasonable ranges (e.g., $\kappa > 0$, $0 \leq \delta \leq 0.8$), reverts invalid entries to previous valid values, and displays warnings when necessary.

The drop-down menu dynamically enables or disables the elongation and triangularity fields based on the selected geometry: both disabled for circular, only elongation enabled for elliptical, and both enabled for D-shaped. This behavior is implemented within the parameter callback function [40]. The codebase comprises approximately 900 lines, following a modular architecture that separates the control interface, computation engine, and visualization components.

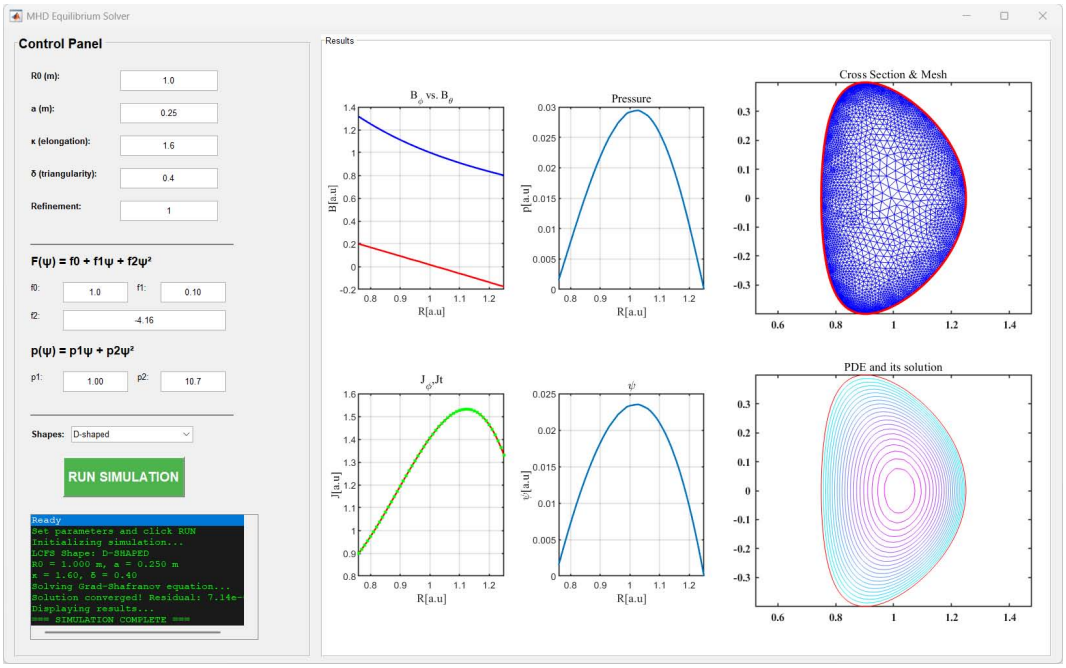


Fig. 2 Developed MHDEquilSolver GUI, showing typical results for a D-shaped configuration with $R_0 = 1$ m, $a = 0.25$ m, $\kappa = 1.6$, $\delta = 0.4$, refinement=1, $f_0 = 1.0$, $f_1 = 0.10$, $f_2 = -4.16$, $p_1 = 1.0$, $p_2 = 10.7$

4.2 Workflow and Data Structures

The computational workflow proceeds as follows: parameter input, simulation initiation, geometry generation, mesh creation with refinement, nonlinear solution, post-processing, and automatic subplot updates. The solver returns a results structure containing the solution ψ, mesh points, edge data, triangle connectivity, pressure field, toroidal field, poloidal field components and magnitude, toroidal current density, LCFS boundary coordinates, final solver residual, and the complete set of input parameters for reproducibility [32]. The overall program execution flow is shown in Fig. 3.

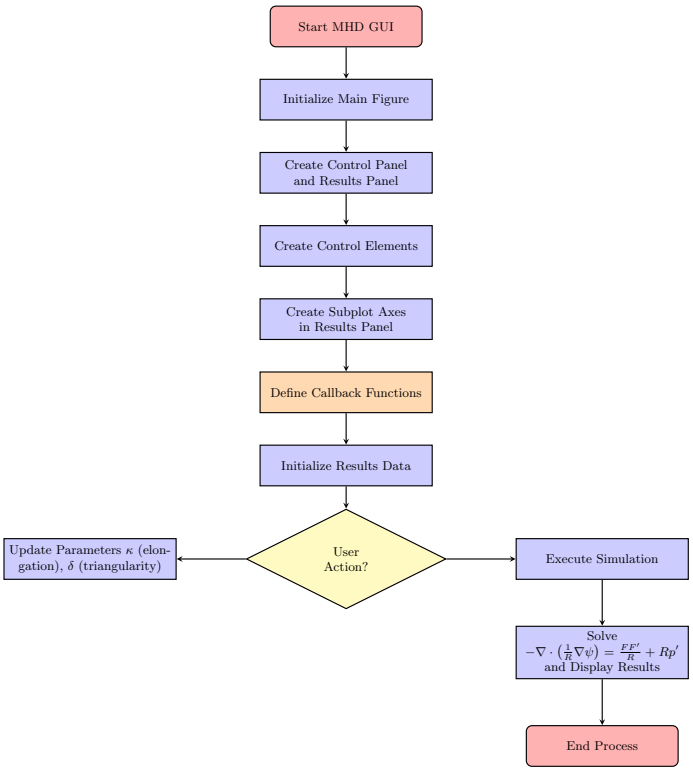


Fig. 3 Overall program execution flow

5 Sensitivity Analysis

5.1 Pressure Coefficients p_1 and p_2

The pressure function and its gradient depend linearly on p_1 and p_2 [31]:

$$\frac{\partial p}{\partial p_1} = \psi, \quad \frac{\partial p}{\partial p_2} = \psi^2, \quad \frac{\partial}{\partial p_1}\left(\frac{dp}{d\psi}\right) = 1, \quad \frac{\partial}{\partial p_2}\left(\frac{dp}{d\psi}\right) = 2\psi. \tag{17}$$

These partial derivatives are positive definite for $\psi > 0$, implying that increasing either coefficient strictly increases the plasma pressure everywhere inside the LCFS. The toroidal current density sensitivity follows from differentiating Eq. (3):

$$\frac{\partial j_\phi}{\partial p_1} = R, \quad \frac{\partial j_\phi}{\partial p_2} = 2R\psi. \tag{18}$$

Since $R > 0$ and $\psi \geq 0$ inside the domain, both sensitivities are non-negative [18]. Physically, increasing p_1 or p_2 enhances the diamagnetic contribution to the toroidal

current, which increases the Shafranov shift—the outward displacement of flux surfaces relative to the geometric center [17]. A factor-of-two increase in p_1 can shift the magnetic axis outward by 5–10% of the minor radius [31].

5.2 Toroidal Field Coefficients f_0, f_1, and f_2

The toroidal field sensitivity is given by [19]:

$$\frac{\partial B_\phi}{\partial f_0} = \frac{1}{R}, \quad \frac{\partial B_\phi}{\partial f_1} = \frac{\psi}{R}, \quad \frac{\partial B_\phi}{\partial f_2} = \frac{\psi^2}{R}. \tag{19}$$

Thus f_0 provides a spatially uniform contribution to B_ϕ (scaled by $1/R$), while f_1 and f_2 introduce flux-dependent modifications that become increasingly important near the plasma edge [30].

The safety factor $q(\psi)$, which characterizes the helical pitch of field lines and determines stability against ideal MHD modes, depends on both $F(\psi)$ and the poloidal field geometry through [4]:

$$q(\psi) = \frac{F(\psi)}{2\pi} \oint_{\psi=\text{const}} \frac{dl}{RB_\theta}, \tag{20}$$

where $B_\theta = |\nabla\psi|/R$ is the poloidal field magnitude and dl is the arc length element along the flux surface. The coefficients f_0, f_1, and f_2 directly affect the q-profile and hence the stability properties of the plasma [20]. The q-profile is computed by integrating along flux surfaces after solving the equilibrium—a computationally intensive post-processing step *not* included in the current GUI for performance reasons [13].

5.3 Geometric Parameters R_0, a, κ, and δ

Geometric parameters alter the problem domain Ω and the metric factor R throughout the equations. The major radius R_0 influences toroidal curvature through the term $-(\frac{1}{R})\frac{\partial\psi}{\partial R}$; increasing R_0 reduces curvature effects, making the equilibrium more cylindrical. In the limit $R_0 \to \infty$, the Grad–Shafranov equation reduces to the two-dimensional Poisson equation [4].

The minor radius a scales the entire cross-sectional area. For fixed profile coefficients, increasing a increases the total poloidal flux and plasma current [5]. The number of mesh elements scales as $(a/H_{\max})^2$, so computational cost increases quadratically with a.

Elongation κ stretches the domain vertically. For fixed R_0 and a, increasing κ increases the plasma volume proportionally to κ, reducing the required vertical field for equilibrium [30]. The safety factor on axis scales approximately as $q_0 \propto 1/\sqrt{\kappa}$ for fixed profiles, so higher elongation reduces the central safety factor and can trigger sawtooth instabilities if $q_0 < 1$ [5].

Triangularity δ introduces a D-shaped deformation that reduces the poloidal field line length on the outboard side [23]. The effects of geometric parameters are summarized in Table 3.

Table 3 Geometric Parameter Effects on Equilibrium

Parameter	Range	Effect on Equilibrium
R_0	1.0–5.0 m	Controls toroidal curvature; larger values reduce Shafranov shift
a	0.2–1.5 m	Scales cross-sectional area; element count $\propto a^2$
κ	1.0–2.5	Vertical stretching; limits for vertical stability
δ	0.0–0.7	D-shaped deformation; $\delta < 0.8$ for well-posed geometry; improves edge stability

6 Results and Discussion

6.1 Visualization and Physical Interpretation

The results panel displays six subplots providing a comprehensive overview of the equilibrium solution [32]. The first subplot shows profiles of the toroidal field $B_\phi(R)$ and poloidal field $B_\theta(R)$ along the equatorial plane; the $1/R$ decay of B_ϕ is characteristic of toroidal geometry. The second subplot presents pressure profiles $p(R)$, peaked on axis for positive p_2, while the current density may be peaked or hollow depending on the f_1/f_2 ratio [20]. In case of H-mode-like profiles with large positive p_2/p_1, the pressure is strongly peaked near the magnetic axis, reflecting the steep edge pedestal of high-confinement regimes [42].

The third subplot shows the finite element mesh with the LCFS boundary highlighted in red, confirming proper mesh generation. The fourth subplot presents toroidal current density profiles $J_\phi(R)$; in standard L-mode operation J_ϕ is peaked on axis, while in advanced reversed-shear scenarios it can become hollow [42]. The fifth subplot shows poloidal flux $\psi(R)$ along the equatorial plane; its shape determines the safety factor profile through Eq. (20)—a flat $\psi(R)$ near the axis corresponds to low q_0, while a steep rise corresponds to high q_0 [5]. The sixth subplot presents filled contours of ψ at 15 equally spaced levels, revealing nested flux surfaces with the magnetic axis at the maximum of ψ (Fig. 2); closely spaced contours indicate strong poloidal field [5].

6.2 Case Studies

Three representative equilibrium configurations are examined to illustrate the solver's flexibility and physical fidelity.

6.2.1 Circular Cross-Section

For a circular reference case ($R_0 = 3.0$ m, $a = 1.0$ m, $p_1 = 0.5$, $p_2 = -0.3$, $f_0 = 5.0$), the computed flux surfaces exhibit perfect concentric symmetry about $R = R_0$, with a small but finite Shafranov shift of $\Delta \approx 0.08a$ directed radially outward toward the low-field side, consistent with toroidal curvature effects.

6.2.2 Elliptical Cross-Section

For an elliptical plasma with $\kappa = 1.5$, the flux surfaces stretch vertically, increasing the plasma volume by approximately κ relative to a circular plasma of the same minor radius. The increased vertical extent requires a stronger vertical field to maintain force balance, which the solver naturally captures.

6.2.3 D-Shaped Cross-Section

For a D-shaped plasma with $\delta = 0.4$, the outer midplane surface is indented, producing a characteristic D-profile. This shape improves H-mode pedestal characteristics by steepening the edge pressure gradient and enhances ballooning mode stability, making it a preferred choice for advanced tokamak operation. The solver faithfully reproduces the associated flux surface compression and shaping effects (Fig. 2).

6.3 Computational Performance

The solver demonstrates robust performance on a standard workstation (Intel Core i3 12th Gen, 16 GB RAM). Typical execution times are summarized in Table 4. For a typical mesh of approximately 5,000 elements, the solver converges within 7–10 iterations, with total execution times of 10–15 seconds depending on mesh size and refinement steps.

Table 4 Performance characteristics of MHDEquilSolver on a standard workstation

Operation	Typical Value	Time (seconds)
Mesh generation	5,000–10,000 elements	2–5
Nonlinear solution	7–10 iterations	3–8
Post-processing	—	2–4
Total execution time	—	**10–15**

Beyond physical insights, MHDEquilSolver serves as a valuable pedagogical instrument for graduate instruction in MHD equilibrium, allowing students to interactively visualize parameter dependencies and flux surface geometry. In research, it supports rapid prototyping by enabling efficient screening of operational scenarios, sensitivity analyses, and quick evaluations of profile effects.

6.4 Limitations

Several limitations should be acknowledged. The current implementation assumes a fixed plasma boundary, restricts to axisymmetric configurations, and relies on a quadratic polynomial representation of profiles. These simplifications are acceptable for educational and rapid prototyping applications but may be insufficient for detailed design studies. Additionally, the model does not incorporate kinetic effects or non-ideal MHD physics, which may be essential for capturing certain macroscopic instabilities and transport phenomena in realistic fusion plasmas.

6.5 Verification and Validation

The numerical implementation has been verified against the analytical Solov'ev equilibrium solution, which provides an exact solution for specific profile choices ($dp/d\psi = c_1$, $dF^2/d\psi = c_2$ constant). The numerical solution agrees with the analytical solution to within expected discretization error, with the L_2 error decreasing as $O(H^2)$ for linear finite elements, consistent with standard theory. Convergence studies have been performed to verify the mesh refinement strategy. Table 5 shows convergence for a typical D-shaped equilibrium ($R_0 = 1.0$ m, $a = 0.25$ m, $\kappa = 1.6$, $\delta = 0.4$), with error computed relative to a high-resolution solution of approximately $\sim 90{,}000$ elements.

Table 5 Convergence study for D-shaped equilibrium

Refinement Level	Elements	Degrees of Freedom	Relative L_2 Error
0	1,357	724	4.2×10^{-2}
1	5,428	2,777	1.1×10^{-2}
2	21,712	10,927	2.8×10^{-3}
3	86,848	43,511	7.1×10^{-4}

The observed convergence rate of approximately $O(H^2)$ confirms the expected behavior for linear finite elements and validates the numerical implementation.

6.6 Future Work

Several enhancements are planned: implementing more general profile functions (empirical fits or analytic forms from experimental data); coupling with transport solvers for self-consistent profile evolution; integrating stability modules for comprehensive MHD stability analysis; extending to non-circular shapes with X-points for divertor configurations; implementing free-boundary capability for plasma-wall interaction studies; optimizing for GPU acceleration and parallel computing; and developing a web-based version to enhance accessibility and broaden the user base.

7 Conclusion

This paper presents MHDEquilSolver, a GUI for solving the Grad–Shafranov equation. The solver employs finite element discretization with mesh refinement and solves the resulting nonlinear system using a damped Newton method with an analytical Jacobian. It supports three plasma boundary shapes (circular, elliptical, and D-shaped) and parameterizes the toroidal field function $F(\psi)$ and pressure profile $p(\psi)$ as quadratic polynomials.

The post-processing module computes all derived physical quantities, including magnetic fields and toroidal current densities. A sensitivity analysis quantifies how input parameters affect the equilibrium. Implementation details—mesh data structures, boundary conditions, gradient evaluation, and profile extraction—are described to aid researchers in code development or adaptation.

The interface updates output graphs in real time upon parameter changes, enabling intuitive parameter-space exploration. Error handling, including fallback mesh generation strategies, ensures graceful degradation rather than failure. The source code is modular and extensible, with future work envisioned to include more general profile functions, coupling with transport solvers for self-consistent profile evolution, and integration with stability analysis modules.

Overall, MHDEquilSolver serves both as an effective training platform for teaching MHD equilibrium principles and as a rapid prototyping environment for tokamak plasma analysis. The detailed numerical implementation, algorithmic choices, and sensitivity analysis provide a solid foundation for further development.

Acknowledgements. The author would like to thank the Alvand-U tokamak team and magnetic confinement group for their hospitality and generous support in performing this research.

Declarations

The authors of the manuscript hereby confirm, certify, and unequivocally declare that we have no competing interests–whether financial, personal, professional, or institutional–that could reasonably be construed to have influenced, compromised, or otherwise undermined the objectivity, rigor, integrity, or impartiality of the research. Data analysis, interpretation, or writing presented in this work.

References

[1] H. Grad and H. Rubin, "Hydromagnetic equilibria and force-free fields," *Proc. 2nd UN Conf. Peaceful Uses At. Energy*, vol. 31, p. 190, 1958.

[2] V. D. Shafranov, "Plasma equilibrium in a magnetic field," *Rev. Plasma Phys.*, vol. 2, p. 103, 1966.

[3] J. P. Freidberg, "Ideal magnetohydrodynamic theory of magnetic fusion systems," *Rev. Mod. Phys.*, vol. 54, no. 3, pp. 801–902, 1982.

[4] J. P. Freidberg, *Ideal MHD*. Cambridge: Cambridge University Press, 2014.

[5] J. Wesson and D. J. Campbell, *Tokamaks*, 4th ed. Oxford: Oxford University Press, 2011.

[6] J. P. Goedbloed and S. Poedts, *Principles of Magnetohydrodynamics*. Cambridge: Cambridge University Press, 2004.

[7] G. Bateman, *MHD Instabilities*. Cambridge, MA: MIT Press, 1978.

[8] P. Helander and D. J. Sigmar, *Collisional Transport in Magnetized Plasmas*. Cambridge: Cambridge University Press, 2005.

[9] 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*, vol. 25, no. 11, p. 1611, 1985.

[10] H. Lütjens, A. Bondeson, and O. Sauter, "The CHEASE code for toroidal MHD equilibria," *Comput. Phys. Commun.*, vol. 97, no. 3, pp. 219–260, 1996.

[11] K. Lackner, "Computation of ideal MHD equilibria," *Comput. Phys. Commun.*, vol. 12, no. 1, pp. 33–44, 1976.

[12] B. J. Braams, "Computational methods in tokamak equilibrium," *Comput. Phys. Rep.*, vol. 4, no. 3-4, pp. 117–193, 1986.

[13] S. C. Jardin, *Computational Methods in Plasma Physics*. Boca Raton, FL: CRC Press, 2012.

[14] C. Mercier, "Equilibres et stabilite des systemes toroidaux," *Nucl. Fusion*, vol. 4, no. 3, p. 213, 1964.

[15] J. M. Greene and J. L. Johnson, "Determination of hydromagnetic equilibria," *Phys. Fluids*, vol. 8, no. 8, p. 1470, 1965.

[16] L. S. Solov'ev, "The theory of hydromagnetic stability of toroidal plasma configurations," *Sov. J. Exp. Theor. Phys.*, vol. 26, p. 400, 1968.

[17] K. Lackner and S. Günter, "Grad-Shafranov equilibria: A tutorial," *Plasma Phys. Control. Fusion*, vol. 40, no. 8, p. 1527, 1998.

[18] S. T. Wu, "A fast finite element method for solving the Grad-Shafranov equation," *J. Comput. Phys.*, vol. 176, no. 1, pp. 1–21, 2002.

[19] Y. M. Jeon, "An efficient solver for the Grad-Shafranov equation using a finite element method," *J. Korean Phys. Soc.*, vol. 47, p. 637, 2005.

[20] O. Sauter, C. Angioni, and Y. R. Lin-Liu, "Neoclassical conductivity and bootstrap current formulas for general tokamak equilibria," *Phys. Plasmas*, vol. 6, no. 7, p. 2834, 1999.

[21] F. Troyon, R. Gruber, H. Saurenmann, S. Semenzato, and S. Succi, "MHD limits to plasma confinement," *Plasma Phys. Control. Fusion*, vol. 26, no. 1A, p. 209, 1984.

[22] The MathWorks Inc., *MATLAB PDE Toolbox User's Guide*. Natick, MA, 2023.

[23] J. R. Ferron, M. L. Walker, L. L. Lao, H. E. St. John, D. A. Humphreys, and J. A. Leuer, "Real-time equilibrium reconstruction for tokamak discharge control," *Nucl. Fusion*, vol. 38, no. 7, p. 1055, 1998.

[24] J. N. Reddy, *An Introduction to the Finite Element Method*, 3rd ed. New York: McGraw-Hill, 2006.

[25] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu, *The Finite Element Method: Its Basis and Fundamentals*, 7th ed. Oxford: Butterworth-Heinemann, 2013.

[26] G. Strang, *Introduction to Applied Mathematics*. Wellesley, MA: Wellesley-Cambridge Press, 1986.

[27] J. D. Logan, *Applied Partial Differential Equations*, 3rd ed. New York: Springer, 2013.

[28] R. L. Burden and J. D. Faires, *Numerical Analysis*, 10th ed. Boston: Brooks/Cole, 2015.

[29] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, 3rd ed. Cambridge: Cambridge University Press, 2007.

[30] R. B. White, *The Theory of Toroidally Confined Plasmas*, 3rd ed. London: Imperial College Press, 2014.

[31] R. K. Sharma, P. K. Sharma, and A. Kumar, "Parameter sensitivity analysis of tokamak equilibria using a Grad-Shafranov solver," *Plasma Sci. Technol.*, vol. 24, no. 6, p. 065101, 2022.

[32] M. R. Faulkner, D. J. Thoen, and E. A. Belli, "A GUI-based Grad-Shafranov solver for tokamak education," *Fusion Eng. Des.*, vol. 199, p. 114321, 2024.

[33] B. J. Braams, "Computational Methods in MHD Equilibrium," *Phys. Fluids B*, vol. 3, p. 2332, 1991.

[34] Y. Sadeghi, "Development of a Graphical User Interface for Grad-Shafranov Equilibrium Analysis in the Alvand-U Tokamak," *Fusion Engineering and Design*, vol. 232, p. 116040, 2026.

[35] Yahya Sadeghi, "Investigation of Plasma Shape, Displacement, Poloidal Beta, and Internal Inductance in the Alvand-U Tokamak through the Solution of the Grad-Shafranov Equation," *Nucl. Eng. Technol.*, 58, 104085, 2026.

[36] Y. Sadeghi, H. Rasouli, F. Sedighi, M. Jafargholi, S. Yarmahmoodi, and H. Babaee, "Recent Upgrade of the Magnetic Diagnostic System in the Alvand-U Tokamak," *Fusion Sci. Technol.*, vol. 81, no. 3, pp. 73-81, 2024.

[37] Yahya Sadeghi, "Determining plasma boundary in Alvand-U tokamak," *Nucl. Eng. Technol.*, vol. 55, pp. 3485-3492, 2023.

[38] Y. Sadeghi and S. Yarmahmoodi, "A sample solution of the Grad-Shafranov equation in a simple geometry and fixed boundary conditions (in persian)," in *20th Nuclear Conference*, University of Guilan-Rasht, Iran, March 2014.

[39] L. Sarkhosh and Y. Sadeghi, "Typical solution of Grad-Shafranov equation for non-circular section with fixed boundary conditions," in *International Conference on Plasma Science and Application*, Najaf abad, Isfahan, Iran, 8-10 September 2015.

[40] P. Marchand and O. T. Holland, *Graphics and GUIs with MATLAB*, 3rd ed. Boca Raton, FL: Chapman & Hall/CRC, 2002.

[41] M. D. Kruskal and J. L. Tuck, "The instability of a pinched fluid with a longitudinal magnetic field," *Proc. R. Soc. London A*, vol. 245, no. 1241, p. 222, 1958.

[42] E. J. Doyle et al., "Chapter 2: Plasma confinement and transport," *Nuclear Fusion*, vol. 47, no. 6, p. 518, 2007.