A Fully Implicit Kinetic Electromagnetic Scheme in XGC-1

Ben Sturdevant

Dec. 20, 2018 (Update Feb. 25, 2019)

Contents

1 Introduction

We are attempting to apply the fully implicit particle-in-cell (PIC) method developed by G. Chen, L. Chacón, and others to an electromagnetic, drift kinetic electron, gyrokinetic ion model in XGC-1. Some relevant references include:

1. G. Chen, L. Chacón and D.C. Barnes, J. Comput. Phys. 230 (18) (2011) 7018-7036
2. G. Chen and L. Chacón, Comput. Phys. Comm. 185 (2014) 2391-2402
3. G. Chen, L. Chacón, C.A. Leibs, D.A. Knoll and W. Taitano, Comput. Phys. Comm. 185 (2014) 2391-2402
4. G. Chen and L. Chacón, Comput. Phys. Comm. 197 (2015) 73-87

The remainder of this document is organized as follows. In section 2, we present the governing equations. Section 3 gives the fully implicit discretization scheme. Section 4 motivates the iterative solution method used to solve the implicit equations at each time step. Section 5 presents the fluid model used to derive the preconditioner we have been using in the iterative method. Section 6 briefly presents some of the recent improvements we have implemented in the preconditioner. Finally, Section 7 gives some initial ideas on how machine learning could be applied to solving the system of nonlinear equations from the implicit method.

2 Governing Equations

For simplicity, in this document we only describe the model for the electrons and take the ions to be described with a density composed of a fixed background part n_0 and the polarization response $n_i^{pol}(\phi) = \nabla_\perp \cdot \left(\frac{n_0 m_i}{q_i B^2}\nabla_\perp \phi\right)$, which appears in the gyrokinetic Poisson equation. This assumption shouldn't matter much for the purposes of preconditioning, since the fast time scales are due to electron physics. Furthermore, we do not discuss the use of the δf method here or subcycling.

2.1 Particle Equations of Motion

Each particle electron is described by a set of coordinates $(\mathbf{X}, v_\parallel, \mu)$, which evolve according to equations of motion coming from variational guiding center theory. See for example [Littlejohn, 1983]. The equations of motion are:

$$\dot{\mathbf{X}} = \frac{1}{D}\left[v_\parallel\left(\hat{\mathbf{b}}_0 + \delta\hat{\mathbf{b}}\right) - \frac{m_e}{eB_0}v_\parallel^2 \nabla\times\hat{\mathbf{b}}_0 - \frac{m_e}{eB_0}\hat{\mathbf{b}}_0\times\left(\frac{e}{m_e}\mathbf{E} + \mu\nabla B_0\right)\right] \tag{1}$$

$$\dot{v}_\parallel = -\frac{e}{m_e D}\left[\left(\hat{\mathbf{b}}_0 + \delta\hat{\mathbf{b}}\right) - \frac{m_e}{eB_0}v_\parallel\nabla\times\hat{\mathbf{b}}_0\right]\cdot\left(\mathbf{E} + \frac{m_e}{e}\mu\nabla B_0\right) \tag{2}$$

$$\dot{\mu} = 0, \tag{3}$$

where m_e and e are the electron mass and charge respectively, $\mathbf{E}$ is the full electric field, $\mathbf{B}_0$ is a constant background magnetic field, and $\delta\mathbf{B}$ is the time evolving perturbed magnetic field. We define the "hatted" quantities by $\hat{\mathbf{b}}_0 = \mathbf{B}_0/B_0$ and $\delta\hat{\mathbf{b}} = \delta\mathbf{B}/B_0$, where B_0 is the magnitude of $\mathbf{B}_0$. Finally, we have defined the quantity:

$$D = 1 - \frac{m_e}{eB_0}v_\parallel \mathbf{b}_0\cdot\nabla\times\mathbf{b}_0.$$

2.2 Field Equations

To advance the particle electrons in time according to Eqs.(1)–(3), we require the fields $\mathbf{E}$ and $\delta\mathbf{B}$. These fields may be expressed in terms of an electrostatic potential ϕ and the parallel component of a vector potential $A_\parallel$ as

$$\mathbf{E} = -\nabla\phi - \frac{\partial A_\parallel}{\partial t}\hat{\mathbf{b}}_0$$
$$\delta\mathbf{B} = \nabla\times\left(A_\parallel\hat{\mathbf{b}}_0\right) = \nabla A_\parallel\times\mathbf{b}_0 + A_\parallel\nabla\times\mathbf{b}_0.$$

Here ϕ and $A_\parallel$ are determined from the quasi-neutrality condition and Ampere's law, respectively. These may be written as:

$$\mathcal{L}_1\phi = \frac{q_i}{e}n_0 - n_e \tag{4}$$

$$\mathcal{L}_2 A_\parallel = -e\Gamma_{\parallel e} = j_\parallel, \tag{5}$$

where $\mathcal{L}_1 = -\frac{n_0 m_i}{q_i B_0^2}\nabla_\perp^2$ and $\mathcal{L}_2 = -\frac{1}{\mu_0}\nabla_\perp^2$. The terms on the right hand side of Eqs.(4)–(5) are obtained by depositing the particle electrons to the mesh. Since the evolution of the particle electrons is determined by the fields, we may write abstractly $n_e = n_e(\phi, A_\parallel)$ and $j_\parallel = j_\parallel(\phi, A_\parallel)$ showing Eqs.(4)–(5) to be a coupled nonlinear system of equations.

3 Discretized Equations

To advance the particle electrons forward in time from time step n to $n+1$, we use the fields $\mathbf{E}^{n+1/2}$ and $\delta\mathbf{B}^{n+1/2}$ defined by

$$\mathbf{E}^{n+1/2} = -\frac{1}{2}\left(\nabla\phi^{n+1} + \nabla\phi^{n}\right) - \frac{2}{\Delta t}\left(A_\parallel^{n+1/2} - A_\parallel^{n}\right)\hat{\mathbf{b}}_0$$
$$\delta\mathbf{B}^{n+1/2} = \nabla \times \left(A_\parallel^{n+1/2}\hat{\mathbf{b}}_0\right).$$

Note, however, that the field information at times $n+1/2$ and $n+1$ is determined by the state of the particles between times n and $n+1$. Hence, at each time step, we must find a self-consistent state for the particle and field systems. The equations which determine the unknown field information can be written as

$$\mathcal{L}_1\phi^{n+1} = \frac{q_i}{e}n_0 - n_e^{n+1} \tag{6}$$
$$\mathcal{L}_2 A_\parallel^{n+1/2} = j_\parallel^{n+1/2}, \tag{7}$$

where n_e^{n+1} and $j_\parallel^{n+1/2}$ have nonlinear dependence on ϕ^{n+1} and $A_\parallel^{n+1/2}$, via coupling with the particle system, i.e.

$$n_e^{n+1} = n_e(\phi^{n+1}, A_\parallel^{n+1/2})$$
$$j_\parallel^{n+1/2} = j_\parallel(\phi^{n+1}, A_\parallel^{n+1/2}).$$

4 Iterative Solution Method

Next, we motivate the iterative method that we use to solve the implicit equations. We start with equations Eqs.(6)–(7), writing the electron density and parallel current as functions of the unknown field variables and drop the time indices for simplicity.

$$\mathcal{L}_1\phi = \frac{q_i}{e}n_0 - n_e(\phi, A_\parallel) \tag{8}$$
$$\mathcal{L}_2 A_\parallel = j_\parallel(\phi, A_\parallel) \tag{9}$$

Suppose we have an approximation of $(\phi, A_\parallel)$ at iteration k and wish to find a small correction to better satisfy Eqs.(8)–(9). We take

$$\phi^{k+1} = \phi^k + \delta\phi \tag{10}$$
$$A_\parallel^{k+1} = A_\parallel^k + \delta A_\parallel, \tag{11}$$

where k is now an iteration index. A Newton-Raphson iterative method can be derived by plugging in and linearizing. We have

$$\mathcal{L}_1\phi^k + \mathcal{L}_1\delta\phi \approx n_0 - n_e^k - \delta n_e$$
$$\mathcal{L}_2 A_\parallel^k + \mathcal{L}_2\delta A_\parallel \approx j_\parallel^k + \delta j_\parallel,$$

where we have defined the linear displacements by

$$\delta\Psi \equiv \delta\phi\frac{\partial\Psi}{\partial\phi} + \delta A_\parallel\frac{\partial\Psi}{\partial A_\parallel}$$

for $\Psi = (n_e, j_\parallel)$. Next, we rewrite the contribution of the zeroth order terms as

$$n_0 - n_e^k - \mathcal{L}_1\phi^k = \mathcal{L}_1\tilde{\phi} - \mathcal{L}_1\phi^k = \mathcal{L}_1 R_\phi^k$$
$$j_\parallel^k - \mathcal{L}_2 A_\parallel^k = \mathcal{L}_2\tilde{A}_\parallel - \mathcal{L}_2 A_\parallel^k = \mathcal{L}_2 R_A^k,$$

where we define

$$\tilde{\phi} = \mathcal{L}_1^{-1} n_e^k$$
$$\tilde{A}_\parallel = \mathcal{L}_2^{-1} j_\parallel^k,$$

and the residuals are

$$R_\phi^k = \tilde{\phi} - \phi^k$$
$$R_A^k = \tilde{A}_\parallel - A_\parallel^k.$$

Finally, we have that the correction to the field variables at iteration k can, in principle, be obtained by solving the following linear system of equations:

$$\begin{bmatrix} \mathcal{L}_1 & 0 & I & 0 \\ 0 & \mathcal{L}_2 & 0 & -I \\ \frac{\partial n_e}{\partial \phi} & \frac{\partial n_e}{\partial A_\parallel} & -I & 0 \\ \frac{\partial j_\parallel}{\partial \phi} & \frac{\partial j_\parallel}{\partial A_\parallel} & 0 & -I \end{bmatrix} \begin{bmatrix} \delta\phi \\ \delta A_\parallel \\ \delta n_e \\ \delta j_\parallel \end{bmatrix} = \begin{bmatrix} \mathcal{L}_1 R_\phi^k \\ \mathcal{L}_2 R_A^k \\ 0 \\ 0 \end{bmatrix} \tag{12}$$

Equations Eqs.(10)–(11) with corrections obtained from solving Eq.(12) is equivalent to the Newton-Raphson method applied to Eqs.(8)–(9). We may write the iteration scheme abstractly as

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \mathcal{P}^{-1}\mathbf{r}\left(\mathbf{x}^k\right), \tag{13}$$

where $\mathbf{x} = [\phi, A_\parallel]^T$, $\mathbf{r} = [R_\phi, R_A]^T$, and $\mathcal{P}$ is an invertible operator mapping the residuals to the iteration correction. Note that the consistency of the iteration method does not depend on the particular form of $\mathcal{P}$ found in Eq.(12). In fact, any invertible $\mathcal{P}$ will produce the correct solution, provided the method converges. The choice of $\mathcal{P}$ will, however, determine if (and at what rate) the method converges. For the remained of the document, we will refer to the operator $\mathcal{P}$ as the preconditioner.

5 Fluid Preconditioner Model

In the previous section, we motivated the iteration scheme of Eq.(13) by deriving a Newton-Raphson method for solving equations Eqs.(8)–(9). Using the left hand side matrix in Eq.(12) for the preconditioner, however, is impractical for a number of reasons. First, it would be nearly impossible to derive an explicit expression for the linear responses of n_e and $j_\parallel$ to changes in the field variables ϕ and $A_\parallel$. These responses would include complicated expressions due to motions of the full particle system, particle-grid interactions, etc. Furthermore, constructing such a matrix could be quite costly, and it would have to be updated each iteration. We wish to instead choose $\mathcal{P}$ with the following properties:

- $\mathcal{P}^{-1}\mathbf{v}$ should be cheap to compute for an arbitrary vector $\mathbf{v}$.
- $\mathcal{P}$ should have similar spectral properties to the matrix on the left hand side of Eq.(12) near the solution.
- $\mathcal{P}$ should either be cheap to construct or should not need frequent updating.

Here, we consider replacing the bottom two equations in Eq.(12) with approximate linear responses coming from fluid equations. Motivated by physics, we choose a model that should accurately capture the fastest time scales in

the system - the shear-Alfvén wave and its electrostatic limit (the Ω_H mode). The starting point is the following system of equations for n_e and $j_\parallel$:

$$\frac{\partial n_e}{\partial t} - B\nabla_\parallel B^{-1}\frac{j_\parallel}{e} = 0$$
$$\frac{m_e}{e^2 n_0}\frac{\partial j_\parallel}{\partial t} - E_\parallel - \frac{T_e}{e n_0}\nabla_\parallel n_e = 0,$$

where n_0 and T_e are the background density and electron temperature respectively. We discretize the above equations in time in a consistent manner to give n_e at integer time steps and $j_\parallel$ at half integer time steps.

$$\frac{n_e^{n+1} - n_e^n}{\Delta t} - B\nabla_\parallel B^{-1}\frac{j_\parallel^{n+1/2}}{e} = 0$$
$$\frac{m_e}{e^2 n_0}\frac{j_\parallel^{n+1/2} - j_\parallel^n}{\Delta t/2} + \left(\frac{1}{2}\nabla_\parallel\left(\phi^{n+1} + \phi^n\right) + \frac{A_\parallel^{n+1/2} - A_\parallel^n}{\Delta t/2}\right) - \frac{T_e}{e n_0}\frac{1}{2}\nabla_\parallel\left(n_e^{n+1} + n_e^n\right) = 0.$$

Finally, we find the linear responses for n_e^{n+1} and $j_\parallel^{n+1/2}$ from the above discretized system to be

$$\delta n - \Delta t B\nabla_\parallel B^{-1}\delta j_\parallel/e = 0$$
$$\frac{m_e}{e n_0}\delta j_\parallel/e + \frac{\Delta t}{4}\nabla_\parallel\delta\phi + \delta A_\parallel - \frac{\Delta t}{4}\frac{T_e}{e n_0}\nabla_\parallel\delta n = 0.$$

Hence, at each iteration, we solve the following system of equations to obtain a correction:

$$\begin{bmatrix} \mathcal{L}_1 & 0 & \mathcal{M} & 0 \\ 0 & \mathcal{L}_2/e & 0 & -\mathcal{M} \\ 0 & 0 & I & -\Delta t B\nabla_\parallel B^{-1} \\ \frac{\Delta t}{4}\nabla_\parallel & I & -\frac{\Delta t}{4}\frac{T_e}{e n_0}\nabla_\parallel & \frac{m_e}{e n_0} \end{bmatrix} \begin{bmatrix} \delta\phi \\ \delta A_\parallel \\ \delta n \\ \delta j_\parallel/e \end{bmatrix} = \begin{bmatrix} \mathcal{L}_1 R_\phi^k \\ \mathcal{L}_2/e\ R_A^k \\ 0 \\ 0 \end{bmatrix}. \tag{14}$$

Here, we have included the mass matrix $\mathcal{M}$ in the top two equations, since the field equations are solved with the finite element method. By approximating the linear responses of the fluid moments, we have sacrificed the quadratic convergence typically achieved with the Newton-Raphson method, however, we have gained a practical iterative method involving a matrix with a known form that can be precomputed.

6 Recent Improvements to the Preconditioner

Here we briefly describe some recent efforts to improve the convergence rate of the iterative method.

6.1 Fixed Point Acceleration

Fixed point (Anderson) acceleration is an attempt to speed up the convergence of a fixed point iterative method by selecting linear combinations of the previous m iterates in such a way as to minimize the residual of the next iterate when applied to a linear contraction mapping. For our purposes, we can use this as a black-box, simply feeding in $\mathcal{P}^{-1}\mathbf{r}$ at each iteration and taking the output to correct the iteration. Some references include:

1. Lecture notes from University of Jyväskylä
2. N.N. Carlson and K. Miller, SIAM J. Sci. Comput. 19 (3), 728-765 (1998). See Section 9.

6.2 Particle-Mesh Interactions

After some analysis of the particle-in-cell equations, it can be shown that there are nonlocal contributions of the electric field that $j_\parallel$ and n_e respond to due to the interpolation and deposition steps in the PIC method. Furthermore, in XGC, these steps are performed in field aligned coordinates before the fluid moments are converted back to cylindrical coordinates. In effect, the fluid moments from the PIC system respond to an electric field that is smoothed along field lines. We were able to accurately model this smoothing with an operator denoted $\mathcal{S}$, which maps the electric field on the mesh to an effective electric field. Including this operator in Eq.(14), we have:

$$\begin{bmatrix} \mathcal{L}_1 & 0 & \mathcal{M} & 0 \\ 0 & \mathcal{L}_2/e & 0 & -\mathcal{M} \\ 0 & 0 & I & -\Delta t B \nabla_\parallel B^{-1} \\ \frac{\Delta t}{4}\mathcal{S}\nabla_\parallel & \mathcal{S} & -\frac{\Delta t}{4}\frac{T_e}{en_0}\nabla_\parallel & \frac{m_e}{en_0} \end{bmatrix} \begin{bmatrix} \delta\phi \\ \delta A_\parallel \\ \delta n \\ \delta j_\parallel/e \end{bmatrix} = \begin{bmatrix} \mathcal{L}_1 R_\phi^k \\ \mathcal{L}_2/e\ R_A^k \\ 0 \\ 0 \end{bmatrix}. \tag{15}$$

In Fig.(6.2), we show the effects of including this operator on the convergence rate. In addition, we show results with and without using fixed point acceleration (FPA). The test results shown use the following parameters:

Δt	4.36×10^{-8} s
n_0	1.0×10^{19} m^{-3}
T_e	2.0 keV
m_e	9.11×10^{-30} kg
m_i	1.67×10^{-27} kg

We use cylindrical geometry and an initial perturbation with toroidal mode number $n = 1$ and poloidal mode number $m = 3$. Note that the electron mass is 10 x the realistic mass in this case.

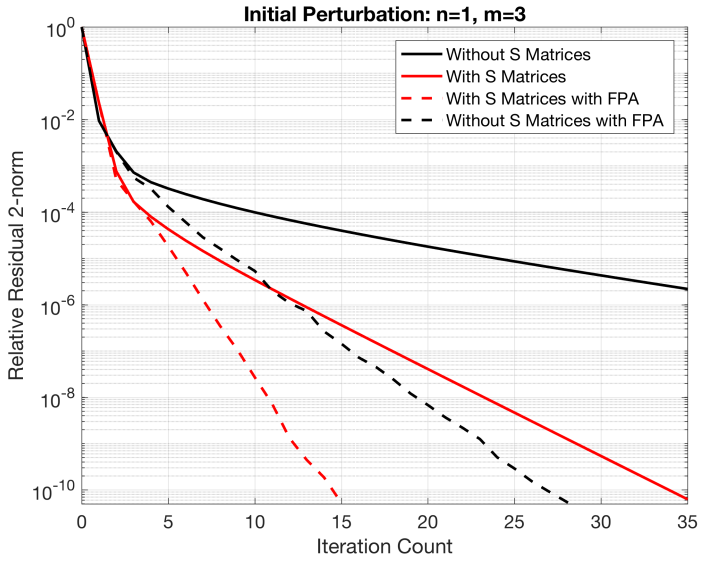

Figure 1: Convergence history comparing the preconditioner including the $\mathcal{S}$ operator to the preconditioner not including $\mathcal{S}$. For each case, we show both with and without fixed point acceleration (fpa).

6.3 Three Field Solver

During our convergence studies, we noticed the following behavior. After a few iterations, a dominant mode in the residual appears, characterized by "red-black" or "even-odd" oscillations in the toroidal direction. The preconditioner seems to struggle to reduce this component. We suspect this is due to the use of central differencing in the $\nabla_\parallel$ operators, since a "red-black" mode is in the near-null-space of this operator. In addition, we noticed that the convergence rate becomes worse when the ratio $v_{th}\Delta t/\Delta s$ was increased. In this ratio, we have the electron thermal velocity $v_{th} = \sqrt{T_e/m_e}$ and Δs is the arclength between two field following nodes. This motivated us reformulate the preconditioner equations to eliminate the first order operator in the fourth row and third column of the block matrix in Eq.(15). Simply replacing δn with $\Delta t B \nabla_\parallel B^{-1}$ yields the following 3 x 3 block matrix system

$$\begin{bmatrix} \mathcal{L}_1 & 0 & \Delta t \mathcal{M} B \nabla_\parallel B^{-1} \\ 0 & \mathcal{L}_2/e & -\mathcal{M} \\ \frac{\Delta t}{4}\mathcal{S}\nabla_\parallel & \mathcal{S} & \frac{m_e}{en_0} - \frac{\Delta t^2 T_e}{4en_0} B\nabla_\parallel^2 B^{-1} \end{bmatrix} \begin{bmatrix} \delta\phi \\ \delta A_\parallel \\ \delta j/e \end{bmatrix} = \begin{bmatrix} \mathcal{L}_1 R_\phi^k \\ \mathcal{L}_2/e\ R_A^k \\ 0 \end{bmatrix},$$

where now the temperature term contains a second order which can be central differenced without supporting a null space. In Fig.(6.3), we show the effects of this reformulation on the convergence rate. In addition, we show results with and without using fixed point acceleration (FPA). The test results shown use the following parameters:

Δt	4.36×10^{-8} s
n_0	1.0×10^{19} m^{-3}
T_e	2.0 keV
m_e	9.11×10^{-31} kg
m_i	1.67×10^{-27} kg

We use toroidal geometry and an initial perturbation with toroidal mode number $n = 1$ and poloidal mode number $m = 4$. Note that we are using the electron mass in this case.

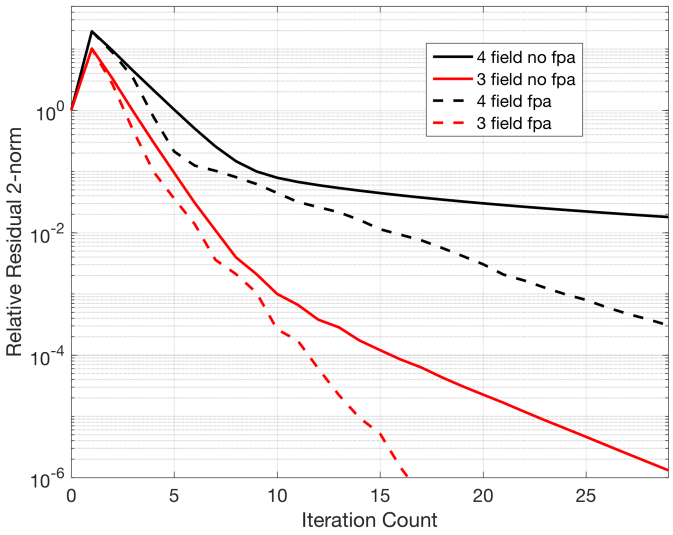

Figure 2: Convergence history comparing the three field preconditioner to the four field preconditioner. For each case, we show both with and without fixed point acceleration (fpa).

7 Possible Starting Points for Machine Learning

Here, we present two possible approaches for using machine learning to aide in the solution of the nonlinear equations resulting from the implicit discretization. Both rely on converged solutions, which can be used to generate error data for particular iterative states.

7.1 Approach 1: Using ML to Directly Predict Error from the Residual

One possible use of machine learning would be to provide data driven predictions of the error in our current iterative state given the residual of the state. In other words, we could replace $\mathcal{P}^{-1}$ in Eq.(13) with a learned nonlinear mapping from $\mathbf{r}^k = \mathbf{r}(\mathbf{x}^k)$ to an approximation of $\mathbf{e}^k$, where $\mathbf{e}^k = \mathbf{x} - \mathbf{x}^k$ with $\mathbf{x}$ such that $\mathbf{r}(\mathbf{x}) = \mathbf{0}$. The current iterative state would then be corrected by:

$$\mathbf{x} \approx \mathbf{x}^k + \tilde{\mathbf{e}}(\mathbf{r}^k), \tag{16}$$

where $\tilde{\mathbf{e}}$ is the predicted error. In generating training data, we have produced converged solutions, which we take to be $\mathbf{x}$, by taking many iterations in our current scheme to reduce the residual to a tight tolerance.

We consider a data set of error-residual pairs coming from several different time steps and iterations:

$$\mathcal{D} = \{(\mathbf{r}^{(1)}, \mathbf{e}^{(1)}), (\mathbf{r}^{(2)}, \mathbf{e}^{(2)}), ..., (\mathbf{r}^{(n)}, \mathbf{e}^{(n)})\}. \tag{17}$$

Here, we index the data set by the subscript j, not necessarily related to time step or iteration index. A simple loss function can be defined for training by:

$$L(\boldsymbol{\theta}) = \sum_{j=1}^{n} \frac{\|\mathbf{e}^{(j)} - \tilde{\mathbf{e}}(\mathbf{r}^{(j)}; \boldsymbol{\theta})\|^2}{\|\mathbf{e}^{(j)}\|^2}, \tag{18}$$

where $\tilde{\mathbf{e}}$ is the predicted error and $\boldsymbol{\theta}$ contains the model parameters to be learned. Here $\|\cdot\|$ is an L^2-like norm accounting for scaling differences between $A_{\|}$ and ϕ. In particular,

$$||\mathbf{e}||^2 = ||e_\phi||^2 + \frac{\mu_0 n_0 m_i}{B^2} ||e_A||^2. \tag{19}$$

We note that the division by $\|\mathbf{e}_j\|^2$ in Eq.(18) is to account for the large variations in size that can occur between $\mathbf{e}^{(j)}$ at different iterations.

7.2 Approach 2: Combining ML with the Fluid Preconditioner Model

In Pathak et al., Chaos 28.4 (2018): 041101, the benefit of combining machine learning with a knowledge based model for predicting spatio-temporal dynamics in nonlinear systems was demonstrated. Perhaps such an approach could be beneficial for a nonlinear equation solver as well. As an example, we could consider a learned nonlinear mapping from $\mathbf{r}^k$ to an approximation of $\Delta\mathbf{e}^k = \mathbf{e}^k - \mathcal{P}^{-1}\mathbf{r}^k$. The current iterative state would then be corrected by:

$$\mathbf{x} \approx \mathbf{x}^k + \mathcal{P}^{-1}\mathbf{r}^k + \tilde{\Delta\mathbf{e}}(\mathbf{r}^k), \tag{20}$$

where $\tilde{\Delta\mathbf{e}}$ is the predicted "error of the error".

The data set is:

$$\mathcal{D} = \{(\mathbf{r}^{(1)}, \Delta\mathbf{e}^{(1)}), (\mathbf{r}^{(2)}, \Delta\mathbf{e}^{(2)}), ..., (\mathbf{r}^{(n)}, \Delta\mathbf{e}^{(n)})\}, \tag{21}$$

and the loss function:

$$L(\boldsymbol{\theta}) = \sum_{j=1}^{n} \frac{\|\Delta \mathbf{e}^{(j)} - \tilde{\Delta}\mathbf{e}(\mathbf{r}^{(j)}; \boldsymbol{\theta})\|^2}{\|\Delta \mathbf{e}^{(j)}\|^2}. \tag{22}$$