Optimizing fusion gain through a simplified whole-device model

June 24, 2025

1 Introduction

This document describes a proposed modeling problem for the summer school hackathon. We'll first describe the physical picture and motivation, then the governing equations, and finally the configuration of software components we'll use to investigate it.

The setting is the on-axis plasma in a Z Pinch device, bounded on one end by a cathode and on the other by an anode. As the Z Pinch current ramps up, and the plasma undergoes compression, on-axis current must connect through a Langmuir sheath at both electrodes. The dynamics of the current can be modeled as a series RLC circuit: a circuit containing a resistor, an inductor, and a capacitor. The RLC circuit equations let us relate the plasma current, which can be observed from the quasi-steady solution of our plasma kinetic equations, to the plasma voltage gap; that is, the voltage gap across the plasma-facing electrodes.

The final piece of the picture is adiabatic compression. To a first approximation, we can assume that the pinch is compressing adiabatically, which gives scaling relations between the plasma current and the on-axis bulk plasma quantities, such as number density and temperature.

Putting all of these pieces together, there is a potential optimization problem to be investigated: what combination of circuit and plasma parameters maximizes a quantity of interest such as time-integrated neutron yield, or perhaps Q-scientific? If the circuit model, bulk plasma model, and sheath model can all be implemented in a differentiable program, then the optimization problem can be tackled with a derivative-based optimizer.

2 Circuit model

This section describes the series RLC circuit equations for a discharging capacitor. Let $Q(t)$ be the charge in the capacitor, C the capacitance, R the resistance, and L the total inductance of the circuit. The total current is assumed constant throughout the circuit, and is equal to the rate of change of the charge:

$$I = I_p = \dot{Q}.$$

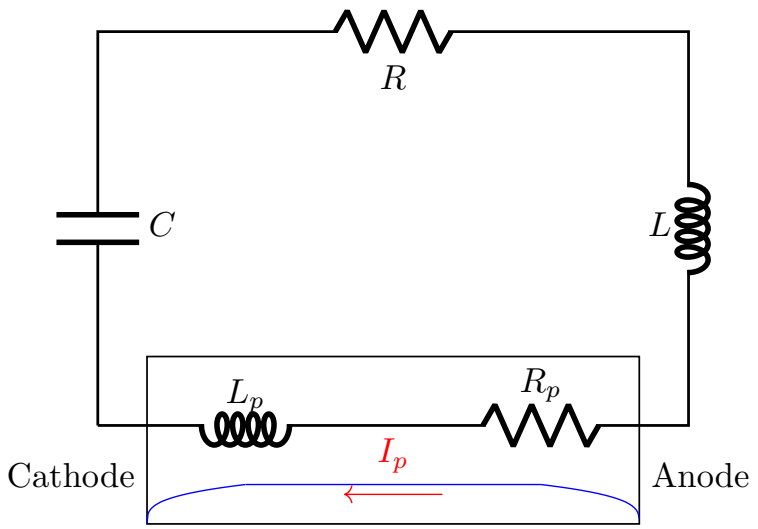
The voltages of each component are related by Kirchoff's voltage law

$$V_R + V_L + V_C - V_p = 0, \tag{1}$$

where V_p is the voltage drop across the plasma, and V_R, V_L, V_C are the voltage drops across the resistor, inductor, and capacitor respectively. The sign of V_p is determined by our chosen convention, that $V_p = \phi_A - \phi_C$, where ϕ_A, ϕ_C are the potentials at the anode and cathode, respectively.

From the definition of capacitance we have $V_C = Q(t)/C$. From Ohm's law we have $V_R = I_p R = \dot{Q}R$, and $V_L = L\dot{I} = L\ddot{Q}$. Finally, we decompose the plasma voltage into the sum of a resistive component and an inductive component:

$$V_p = V_{Rp} + L_p\ddot{Q}$$



Combining all of the voltage terms, we obtain a second-order ODE for $Q(t)$:

$$(L - L_p)\ddot{Q} + R\dot{Q} + \frac{Q}{C} = V_{Rp} \tag{2}$$

Equation (**??**) is not trivial to solve, because $\dot{Q}$ and V_p are related via a complex plasma model. For now, we may consider that we have a forward plasma model that lets us compute the current driven by any given bias voltage:

$$I_p = I_p(V_p;\ T, n, L).$$

We have included the assumed dependence on temperature, density and plasma length as parameters for illustration purposes; in what follows we omit these parameters.

An implicit Euler scheme for (**??**) is

$$\begin{bmatrix} Q^{n+1} \\ \dot{Q}^{n+1} \end{bmatrix} = \begin{bmatrix} Q^{n} \\ \dot{Q}^{n} \end{bmatrix} + \Delta t \begin{bmatrix} I_p^{n+1} \\ \frac{1}{L-L_p}\left(-Q^{n+1}/C - RI_p^{n+1} + V_{Rp}^{n+1}\right) \end{bmatrix}. \tag{3}$$

This may be solved by finding the root of the residual function

$$\mathcal{R}\left(\begin{bmatrix} Q^{n+1} \\ V_p^{n+1} \end{bmatrix}\right) = \begin{bmatrix} Q^{n+1} - Q^n - \Delta t I_p^{n+1} \\ -I_p^{n+1} + \dot{Q}^n + \frac{\Delta t}{L-L_p}\left(-Q^{n+1}/C - RI_p^{n+1} + V_{Rp}^{n+1}\right) \end{bmatrix},$$

where $I_p^{n+1} = I_p(V_p^{n+1})$, and the resistive component of the plasma voltage must be solved for:

$$\begin{aligned} V_{Rp}^{n+1} &= V_p^{n+1} - L_p\ddot{Q} \\ &= V_p^{n+1} - \frac{L_p}{L - L_p}\left(-\frac{Q^{n+1}}{C} - RI_p^{n+1} + V_{Rp}^{n+1}\right), \end{aligned}$$

giving

$$V_{Rp}^{n+1} = \left(1 + \frac{L_p}{L - L_p}\right)^{-1}\left(V_p^{n+1} - \frac{L_p}{L - L_p}\left(-\frac{Q^{n+1}}{C} - RI_p^{n+1}\right)\right).$$

The root of the residual $\mathcal{R}$ can be found by a Newton iteration. In a differentiable programming environment, such a routine is trivial to write, but it relies on the ability to evaluate the gradient of I_p. This indicates that for our circuit solve, we require an end-to-end differentiable kinetic plasma simulation.

3 Incorporating plasma compression, resistive heating, and radiative cooling

As the capacitor discharge modeled by the above RLC circuit equations proceeds, the plasma between the electrode gap is far from static. Its bulk properties (principally density and temperature) change, which drive increasing fusion yield. For the purposes of this modeling exercise, we include three effects:

- Adiabatic compression of the plasma on-axis
- Resistive heating of the plasma by the plasma current
- Radiative cooling of the plasma, principally by so-called Bremsstrahlung radiation

We incorporate these effects into an extended ODE model for current, temperature, and density, with the following time-splitting structure:

$$\frac{d}{dt}\begin{pmatrix} Q \\ I_p \\ T \\ n \end{pmatrix} = \begin{pmatrix} I \\ \ddot{Q} \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \left(\frac{dT}{dt}\right)_{rad} + \left(\frac{dT}{dt}\right)_{\eta} \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \left(\frac{dT}{dt}\right)_{comp} \\ \left(\frac{dn}{dt}\right)_{comp} \end{pmatrix}.$$

The idea is to step each term separately. We begin with an Implicit Euler step of the current terms, described above, followed by an explicit solve of the radiative and resistive terms. Finally, we assume that the compression step occurs adiabatically and reaches radial force balance much faster than a single timestep.

3.1 Resistive heating

Resistive heating of a plasma follows a volumetric analogue to Ohm's law. Heating per unit volume is given by

$$P_\eta = \eta j^2,$$

where η can be estimated by the (corrected) Spitzer resistivity [**goldstonIntroductionPlasmaPhysics1995a**]

$$\eta = \frac{1}{1.96}\frac{\sqrt{2}m_e^{1/2}Ze^2\ln\Lambda}{12\pi^{3/2}\epsilon_0^2 T^{3/2}},$$

and we will use the average current density,

$$j = \frac{I}{\pi a^2},$$

where $a = \sqrt{N/\pi n}$ is the pinch radius.

Dividing by the volumetric density, we get

$$\left(\frac{dT}{dt}\right)_\eta = \frac{P_\eta}{n} = \frac{\eta j^2}{n}.$$

3.2 Radiative losses

The primary radiative cooling of interest in a fusion plasma is bremsstrahlung radiation. This form of radiation is produced by electrons accelerating (or decelerating), and in a plasma is associated with Coulomb collisions between ions and electrons.

The approximate volumetric power density of bremsstrahlung losses in a plasma is [**goldstonIntroductionPlasmaPhysics19**

$$P_{br}[\mathrm{eVs^{-1}m^{-3}}] = 1.06\times10^{-19}Z^2 n_{e[\mathrm{m^{-3}}]}n_{i[\mathrm{m^{-3}}]}T_{[\mathrm{eV}]}^{1/2}.$$

We can assume $n_e = Zn_i$, and for a hydrogen or D-T plasma we have $Z = 1$. Once again dividing by the volumetric density, we get

$$\left(\frac{dT}{dt}\right)_{rad} = -\frac{P_{br}}{n}$$

3.3 Adiabatic compression

The principal effect of increasing the plasma current is to compress the plasma: as it seeks to maintain force balance between the confining magnetic field and its own thermal pressure, the latter increases. By making the reasonable assumption that this process is *adiabatic*, meaning that it occurs slowly enough that no shock waves appear, we can derive scaling relations between two states that are connected by a compression event. Denoting the two states by subscripts 1 and 2, [**shumlakShearedFlowStabilizedZPinch2012**] provides the following adiabatic scaling relations. We assume that the linear density N is constant.

For any plasma in radial force balance, the temperature is related to current by

$$\frac{T_2}{T_1} = \frac{I_2^2}{I_1^2}.$$

The volumetric density scales as

$$\frac{n_2}{n_1} = \left(\frac{T_2}{T_1}\right)^{\frac{1}{\gamma-1}},$$

where γ is the adiabatic constant, which we will take to be 5/3.

3.4 Putting it all together

To advance from time t^n to time t^{n+1} we use the following scheme:

$$\begin{aligned}
\begin{pmatrix} Q^{n+1} \\ I_p^{n+1} \end{pmatrix} &= IE \begin{pmatrix} Q^n \\ I_p^n \end{pmatrix}, \\
T' &= T^n + \Delta t \left(\frac{P_\eta^n}{n^n} - \frac{P_{br}^n}{n^n} \right) \\
T^{n+1} &= (I_p^{n+1}/I_p^n)^2 T^n \\
n^{n+1} &= (T^{n+1}/T')^{\frac{1}{\gamma-1}} n^n
\end{aligned}$$

where IE represents a single timestep of the Implicit Euler scheme (**??**). To tell a story about the final three equations, T' represents the temperature after a single timestep of non-adiabatic heating and cooling. Depending on whether resistive heating or radiative cooling dominates, the plasma is either over- or under-pressured at this point. It then undergoes expansion or compression until it is in radial force balance again. Its temperature in radial equilibrium is independent of T', and depends only on the ratio of currents between times t^{n+1} and t^n. However, the adiabatic expansion or compression is between states with temperatures T' and T^{n+1}, so that the ratio of volumetric densities is determined by T'.

4 Kinetic modeling of the plasma sheath

For simulating the sheath dynamics, we need to carefully design the model problem in such a way that it preserves the steady-state behavior that we want, as well as the bulk resistivity. Both the sheath structure and the plasma resistivity are key to obtaining the correct plasma current density. We use a setup similar to that described in [**skolarContinuumKineticInvestigation2023**]. The key features are:

- A domain with a length of around 256 (in units of the Debye length), bounded on both ends by an absorbing wall boundary. Inside the domain we solve the Vlasov-Poisson system, plus collisions to effect resistivity.

- The desired bias voltage V_p is applied as a boundary condition on the Poisson equation.
- A "flux source term", whose role is to replace the particles lost to the walls, is applied to keep the total particle number nearly constant. This lets us achieve a steady state solution.
- A collision operator is localized to the center of the domain. Notionally, we let the interior two-thirds of the domain correspond to the nearly 5×10^5 Debye lengths between electrodes in a physical Z Pinch. To obtain the correct resistivity, we enhance the collisionality (and thereby reduce the particle mean free path) by a ratio corresponding to the ratio between the simulation domain length and the interelectrode gap.

The governing kinetic equation for our study is the 1D1V Vlasov-Poisson-BGK equation in the "flexible plasma normalization" [**millerMultispecies13momentModel2016**]:

$$\partial_t f_s + v\partial_x f_s + (\omega_c\tau)\frac{Z_s}{A_s}E\partial_v f_s = Q(f_s) + \Gamma_s \tag{4}$$

The collision operator Q is a linearized version of the BGK collision operator about an equilibrium distribution with temperature T_0 and velocity $u = 0$:

$$Q(f_s) = \nu_s(x)(M_s(v) - f_s),$$

where

$$M_s(v) = \frac{n_s(x)}{(2\pi T_0/A_s)^{1/2}} \exp\left(-\frac{A_s|v|^2}{2T_0}\right).$$

The collision frequency $\nu_s(x)$ is spatially dependent, since we apply an increased collision frequency in the center of the domain in order to simulate equilibration with the bulk plasma. The physical mean free path is many thousands of Debye lengths, but our simulation domain is only a few hundred. Increasing ν lets us obtain a Maxwellian distribution in the center of the domain.

The term Γ_s is a source term localized to the center of the domain that serves to replenish particles lost to the absorbing wall:

$$\Gamma_s = \gamma(x)M_s(v) = \left(\frac{1}{L_s} - \frac{|x|}{L_s^2}\right)F_{out}M_s(v),$$

where F_{out} is the flux of *ions* out of the domain:

$$F_{out} = \int v\left(f_i|_{x=-L/2} + f_i|_{x=L/2}\right)\,\mathrm{d}v$$

The normalization constants appearing in Equation (**??**) are as follows:

- Z_s and A_s are the normalized charge and mass of species s, expressed in units of the proton charge and mass.
- $\omega_c\tau$ is the normalized reference proton cyclotron frequency in a field B_0 which would give unit plasma beta: $|B_0|^2/2\mu_0 = n_0T_0$.

For now we can take these normalization constants as given; we will need to concern ourselves with their definitions when we move on to translating a specific physical problem into our equation setup.

Equation (**??**) is coupled to the normalized Gauss's law,

$$\partial_x E = \frac{(\omega_p\tau)^2}{\omega_c\tau}\rho_c, \tag{5}$$

where

$$\rho_c = \sum_s Z_s \int f_s\,\mathrm{d}v \tag{6}$$

is the charge density. We will use the elliptic form of Gauss's law,

$$\partial_x^2 \phi = -\frac{(\omega_p \tau)^2}{\omega_c \tau} \rho_c, \tag{7}$$

with $E = -\partial_x \phi$.

4.1 Domain and boundary conditions

We'll use a physical domain of length L_x, which by convention extends from $-L_x/2$ to $L_x/2$.

4.1.1 Absorbing wall

The simplest boundary condition that will produce a Langmuir sheath is the absorbing wall boundary condition. At a spatial boundary x_b with outward normal vector $\mathbf{n}(x_b)$, we have

$$f_s^b(x_b, v) = \mathbf{1}_{\mathbf{n}(x_b)\cdot v < 0} f_s(x, v), \tag{8}$$

where $\mathbf{1}$ is the indicator function.

4.2 Full-tensor discretization

We use the SSPRK2 time stepping scheme:

$$\begin{aligned} f^{(1)} &= f^n + \Delta t \mathcal{V}(f^n) \\ f^{n+1} &= \frac{f^n}{2} + \frac{f^{(1)}}{2} + \Delta t \mathcal{V}(f^{(1)}), \end{aligned}$$

where $\mathcal{V}$ represents the right-hand side of the Vlasov-Poisson-BGK equation. First-order derivatives are discretized using a slope-limited finite volume scheme with direct upwinding based on the sign of the flux (v or E).

4.3 Straightforward DLR scheme

This is a multispecies, collisional extension of the single-species scheme described in [**einkemmerLowRankProjectorSplitti**
We give each species a low-rank ansatz:

$$f(x, v, t) = \sum_{ij} X_i(x, t) S_{ij}(t) V_j(v, t)$$

The time integration is accomplished with the projector-splitting integrator using a first-order Lie-Trotter splitting. Boundary conditions in x must be applied to the K basis functions, rather than X; this slightly modifies the form of the $\partial_x X$ terms in the S and L substeps. For a comprehensive description of the treatement of boundary conditions in the low-rank framework, see [**huAdaptiveDynamicalLow2022**].

4.3.1 Low-rank substeps

K step:

$$\begin{aligned} \partial_t K_j = &- \sum_{l=1}^{r} \left\langle V_j, v^+ V_l \right\rangle_v D_+ K_l(x,t) - \sum_{l=1}^{r} \left\langle V_j, v^- V_l \right\rangle_v D_- K_l(x,t) \\ &- (\omega_c \tau) \frac{Z}{A} \left[\sum_{l=1}^{r} \left\langle V_j, D_+ V_l \right\rangle_v E_+ K_l(x,t) + \sum_{l=1}^{r} \left\langle V_j, D_- V_l \right\rangle_v E_- K_l(x,t) \right] \\ &+ (\nu(x) + \gamma(x)) \left\langle V_j, M \right\rangle_v - \sum_{l=1}^{r} \nu(x) \left\langle V_j, V_l \right\rangle_v K_l(x,t) \end{aligned}$$

S step:

$$\begin{aligned}\partial_t S_{ij} = &\sum_{l=1}^{r} \left\langle V_j, v^+ V_l \right\rangle_v \langle X_i, D_+ K_l \rangle_x + \sum_{l=1}^{r} \left\langle V_j, v^- V_l \right\rangle_v \langle X_i, D_- K_l \rangle_x \\ &+ (\omega_c \tau) \frac{Z}{A} \left[\sum_{k,l=1}^{r} \langle V_j, D_+ V_l \rangle_v \langle X_i, E_+ X_k \rangle_x S_{kl} + \sum_{k,l=1}^{r} \langle V_j, D_- V_l \rangle_v \langle X_i, E_- X_k \rangle_x S_{kl} \right] \\ &- \langle X_i, \nu(x) + \gamma(x) \rangle_x \langle V_j, M \rangle_v + \sum_{k,l=1}^{r} \langle X_i, \nu(x) X_k \rangle_x \langle V_j, V_l \rangle_v S_{kl}\end{aligned}$$

L step:

$$\begin{aligned}\partial_t L_i = &- \sum_{l=1}^{r} v^+ V_l \langle X_i, D_+ K_l \rangle_x - \sum_{l=1}^{r} v^- V_l \langle X_i, D_- K_l \rangle_x \\ &- (\omega_c \tau) \frac{Z}{A} \left[\sum_{k=1}^{r} \langle X_i, E_+ X_k \rangle_x D_+ L_k + \sum_{k=1}^{r} \langle X_i, E_- X_k \rangle_x D_- L_k \right] \\ &+ \langle X_i, \nu(x) + \gamma(x) \rangle_x M - \sum_{k=1}^{r} \langle X_i, \nu(x) X_k \rangle L_k\end{aligned}$$