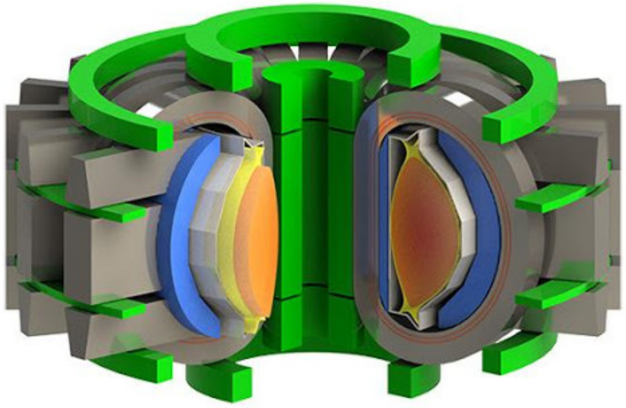
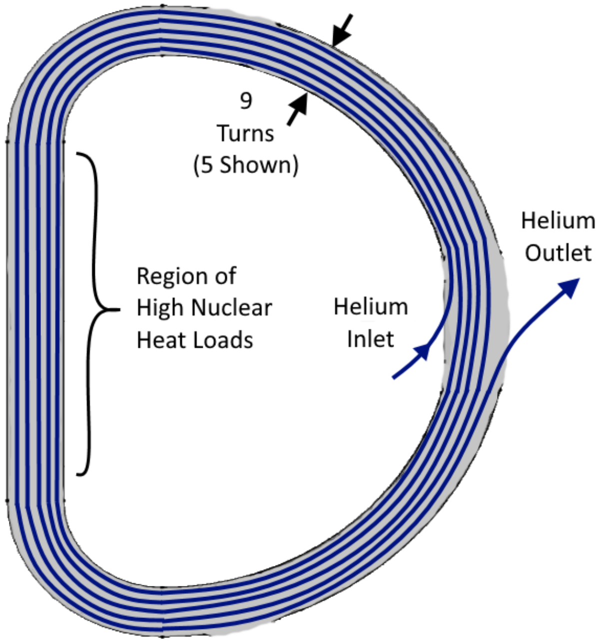
# Investigation into Numerical Stability for the simulation of supercritical helium flows in superconducting magnets

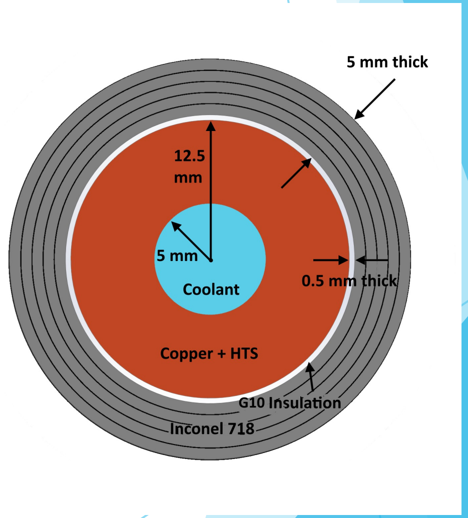
Ben Hamilton

# What and Why

- Supercritical helium is used to maintain superconducting magnets cold enough to sustain large electrical currents without loss
- Fusion systems require these magnets to confine their plasma





# Supercritical Fluid

- Ideal Gas
  - Simple, algebraic equation of state
  - Compressible flow
  - Constant transport properties
  - Coupling of energy, momentum, mass equations that may be non-linear but are entirely algebraic
- Incompressible Fluid
  - No change in density
  - Divergence-free velocity field
  - Energy equations decoupled from momentum equations
- Supercritical fluid
  - Equation of state complicated
  - Large relative changes in properties with temperature
  - Full coupling of energy, momentum and mass equations with possibly non-algebraic relations (worse than non-linear!)

# Governing Equations for a 1-D flow

- Mass

$$-\frac{\partial \rho}{\partial t} = \rho\frac{\partial v}{\partial x} + v\frac{\partial \rho}{\partial x}$$

- Momentum

$$\frac{\partial v}{\partial t} = -v\frac{\partial v}{\partial x} - \frac{1}{\rho}\frac{\partial p}{\partial x} - \frac{fv|v|}{2D_h}$$

- Energy

$$\frac{\partial h}{\partial t} = -v\frac{\partial h}{\partial x} - \frac{3v^2}{2}\frac{\partial v}{\partial x} - \frac{v^3}{2\rho}\frac{\partial \rho}{\partial x} + \frac{1}{\rho A}\frac{\partial q}{\partial x}$$

- Also

$$p = p(h, \rho) \qquad \frac{\partial q}{\partial x} = g(T, \rho, v, \mu, \Pr)$$

$$T = T(h, \rho) \qquad f = f(\rho, v, \mu, D_h)$$

Boundary Conditions:

$$p(0,t) = p_0$$

$$\begin{cases} T(0,t) = T_0 & v(0,t) > 0 \\ \left.\frac{\partial T}{\partial x}\right|_{(0,t)} = 0 & v(0,t) < 0 \end{cases}$$

$$p(L,t) = p_e$$

$$\left.\frac{\partial T}{\partial x}\right|_{(L,t)} = 0$$

Initial Conditions:

$$T(x,0) = T_0$$

$$p(x,0) = \frac{x}{L}p_e + (1 - \frac{x}{L})p_0$$

# Original Solution Method

- All spatial derivatives are 2$^{nd}$ order centered space (except at boundaries)
- FTCS method for the advection equation are unstable without adequate diffusion, so need at least semi-implicit method:
  - System is non-linear, and not all equations are known algebraic functions. So we need an iterative method
  - Jacobi's method with modified SUR for implicit time scheme (1$^{st}$ order convergence)
- Check the CFL condition

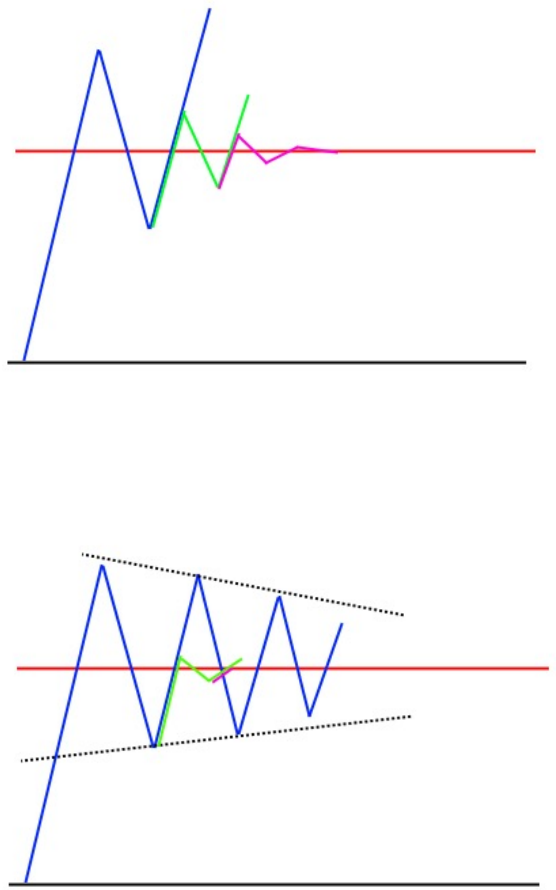
$$\frac{v_{\max}\Delta t}{\Delta x} = \frac{20 \cdot 0.001}{0.2} = 0.1$$

# Modified SUR

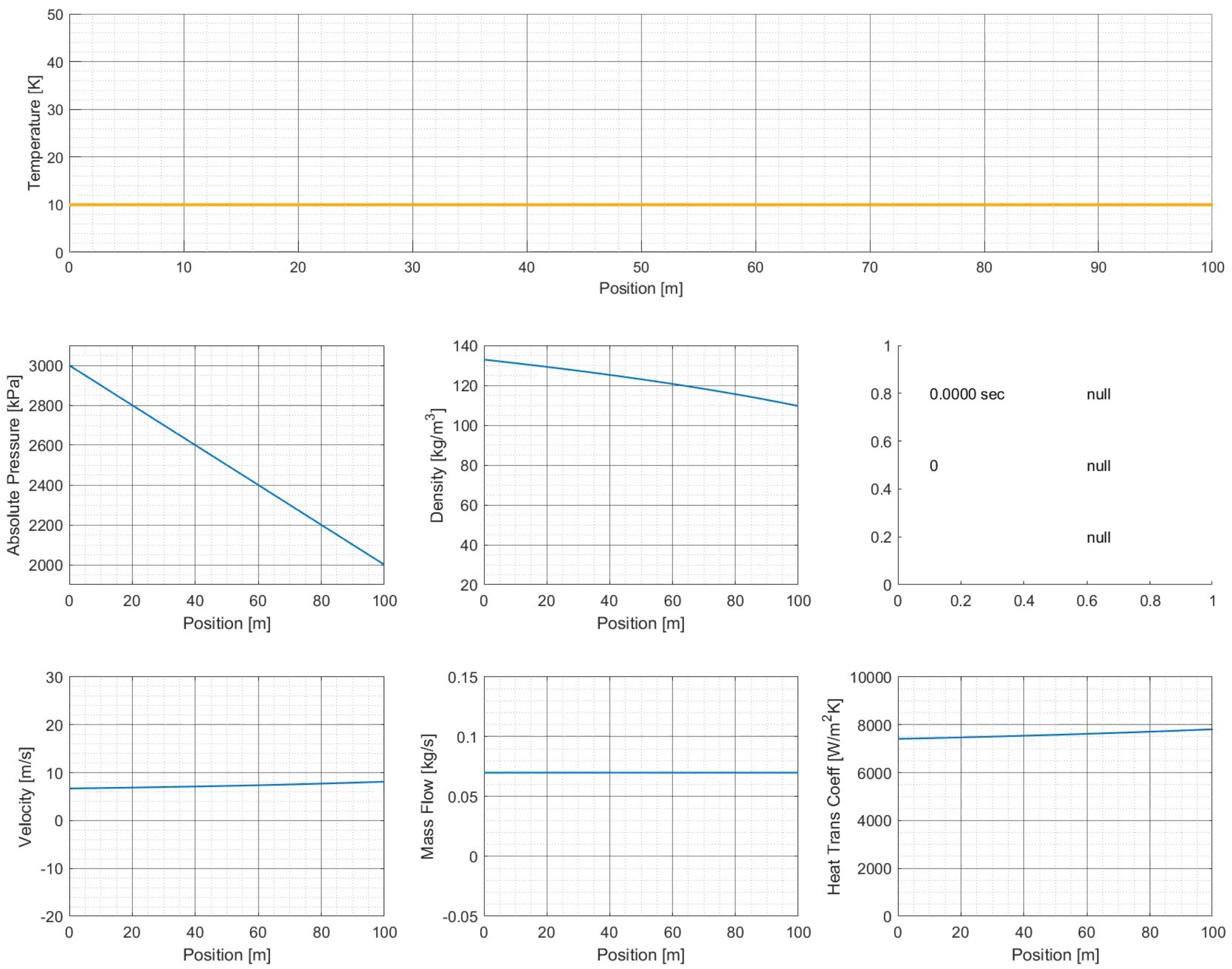
- Goal of successive relaxation is to reduce spectral radius of B (reduce impact of minor variations blowing up) with the parameter w
  - w ~ 1: Nearly a full step, but spectral radius of B may be too large and solution blows up
  - w small: Slight increment to final solution, many iterations, long solution time.
  - How can we balance the two?

# Modified SUR

- Start with w ~ 1
  - If the solution starts to get worse, cut w by a factor (0.5)
  - Slowly increase w by another factor each step (about 1.05) to help rate of convergence
- If solution converging slowly (i.e. due to oscillations)
  - Cut w by a factor (0.5)
- Algorithm may get too aggressive in reducing w, so stop when w reaches a certain min value and call it a loss
  - Goal is to tune these parameters to achieve convergence at almost all time steps



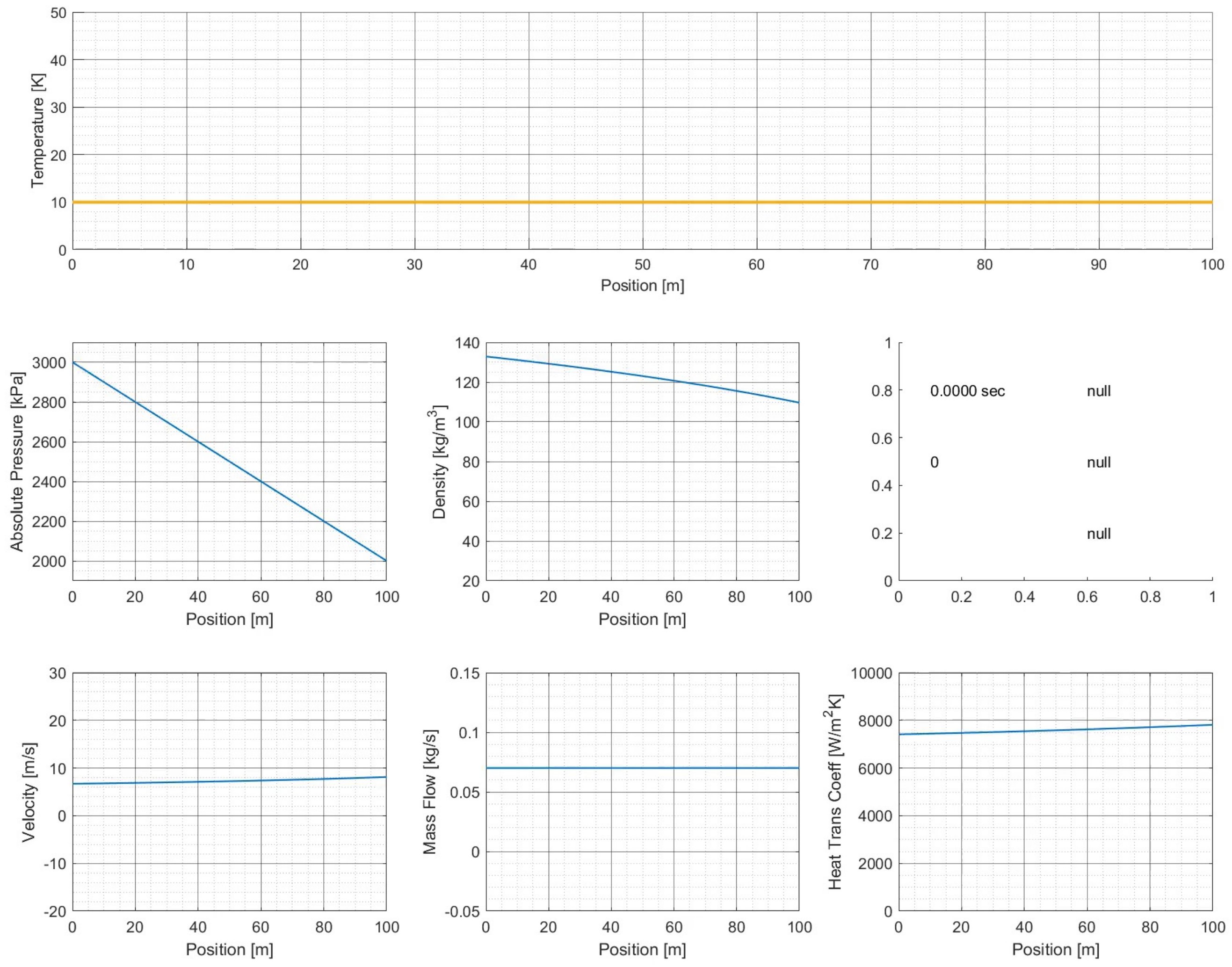
# Original Solution



# Improved Scheme: C-N

- The solution method is an iterative implicit method, so we get knowledge of the time derivatives at step k as well as at step k+1 without any additional computation.
- Implement the C-N scheme for $2^{nd}$ order time accuracy
  - Is it stable? The equations are non-linear so von-Neumann analysis makes no promises
  - Unfortunately, it is not stable enough
    - Bug in code or simply need the diffusion from the purely implicit method

# C-N Implementation



# Final Thoughts

- Highly coupled non-linear system of PDEs, even in 1 spatial dimension, is difficult to solve
  - Needed implicit method for advection instability
  - Needed small time steps to ensure stability from non-linear terms
- Use Jacobi's method with a modified relaxation method to solve
  - Why not a root-finding algorithm? Time. The jacobian matrix (or even its estimate using FD) was neither sparse nor small.
  - Ease of handling changing boundary conditions
  - SUR method tunable to get best chance of stability
- Future work
  - Implement faster convergence method
  - Improve the C-N scheme so that it is stable

# Questions?