Tokamak 3D Equilibrium Reconstruction

A Deep Learning approach

Lorenzo Rossi

Università di Rome "Tor Vergata"

June 7, 2022

Contents

Introduzione

Tramite le equazioni MHD (*MagnetoHydroDynamics*) è possibile considerare il plasma come un fluido conduttore soggetto all'azione di un campo magnetico.

Equilibrio

L'equilibrio è quella situazione in cui tutte le forze agenti su di essa hanno risultante nulla.

La ricostruzione dell'equilibrio del plasma è necessaria al miglioramento dell'efficienza fusionistica e alla protezione delle componenti che costituiscono il Tokamak.

Dinamica del plasma

La dinamica del plasma viene descritta da:

- Continuity equation (1):$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \nu) = 0$;
- Momentum equation (3):$\rho \frac{\partial \nu}{\partial t} + \rho(\nu \cdot \nabla)\nu = J \times B - \nabla p$;
- Ideal Ohm's law (3):$E + \nu \times B = 0$;
- Faraday's law (3):$\frac{\partial B}{\partial t} = -\nabla \times E$;
- "Low frequency" Ampere's law (3):$\mu_0 J = \nabla \times B$;
- Magnetic divergence (1):$\nabla \cdot B = 0$;
- Energy (1):$\frac{d}{dt}(\frac{p}{\rho^{\gamma}})$;

Equilibrio del plasma

Assumendo che:

- Il plasma si trovi in regime stazionario: $\frac{\partial}{\partial t} = 0$
- Riferimento in v ($\nu = 0$);

Si ottiene:

- $J \times B = \nabla p$
- $\mu_0 J = \nabla \times B$
- $\nabla \cdot B = 0$

Grad-Shafranov

Per giungere infine all'equazione di Grad-Shafranov occorre supporre simmetria toroidale. In particolare:

$$\frac{\partial}{\partial \phi} = 0$$

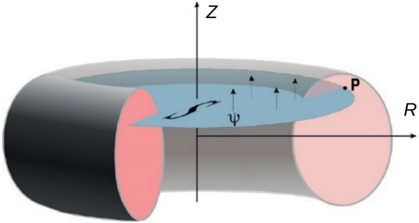


Figure: Simmetria toroidale Tokamak

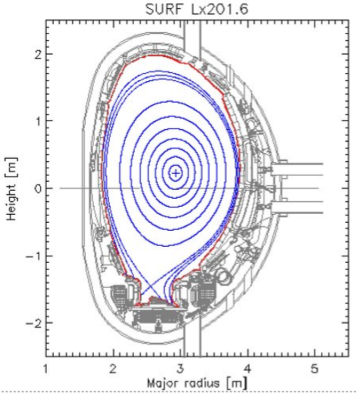
Equazione di Grad-Shafranov

$$p = f(\psi)$$

$$F = g(\psi)$$

$$\frac{\partial^2 \psi}{\partial r^2} - \frac{1}{r}\frac{\partial \psi}{\partial r} + \frac{\partial^2 \psi}{\partial z^2} = -\mu_0 r^2 \frac{dp}{d\psi} - \frac{1}{2}\frac{dF^2}{d\psi}$$

Limitazioni



Sebbene questo metodo consenta di ricostruire efficientemente l'equilibrio del plasma. Tuttavia:

- Il plasma non è sempre in regime stazionario;
- La simmetria toroidale non è sempre rispettata.

Una valida alternativa per ottenere più informazioni sul processo in questione viene fornita dal metodo Physics Informed Neural Network basati sul deep learning.

Physics Informed Neural Network

Il Physics Informed Neural Network è un metodo di deep learning basato su reti neurali per risolvere le PDE (*Partial Differential Equation*).

- Permettono di risolvere numericamente equazioni differenziali molto complesse;
- Le soluzioni delle PDE minimizzano una funzione di costo dipendente dalle equazioni fisiche;
- La funzione di costo deve essere ben modellata per aderire al modello preso in considerazione;
- Processo di training elevato;
- Si necessitano di condizioni al contorno;

Input: R, Z, ϕ

Output:p, B_r, B_z, B_ϕ

Weight Factor = α

$\nabla \cdot B = 0$

$\downarrow$

Loss1=$\frac{1}{r}\frac{\partial r B_r}{\partial r} + \frac{1}{r}\frac{\partial B_\phi}{\partial \phi} + \frac{\partial B_z}{\partial z}$

$\nabla \times B = \mu_0 J$

$\downarrow$

Loss2=$(\frac{1}{r}\frac{\partial B_z}{\partial \phi} - \frac{\partial B_\phi}{\partial z})\mathbf{r} + (\frac{\partial B_r}{\partial z} - \frac{\partial B_z}{\partial r})\phi - \frac{1}{r}(\frac{\partial (r B_\phi)}{\partial r} - \frac{\partial B_r}{\partial \phi})\mathbf{z} - \mu_0 \mathbf{J}$

$J \times B = \nabla p$

$\downarrow$

Loss3=$J_\phi B_z - J_z B_\phi - \frac{\partial p}{\partial R} + J_z B_r - J_r B_z - \frac{\partial p}{\partial \phi} + J_r B_\phi - J_\phi B_r - \frac{\partial p}{\partial Z}$

Vincoli al bordo

$\downarrow$

$p_0, B_{r0}, B_{t0}, B_{Z0}$ noti

Loss4=$\frac{mean(p-p_0)^2}{mean(p_0)^2} + \frac{mean(B_z-B_{z0})^2}{mean(B_{z0})^2}\frac{mean(B_r-B_{r0})^2}{mean(B_{r0})^2}\frac{mean(B_\phi-B_{\phi 0})^2}{mean(B_{\phi 0})^2}$

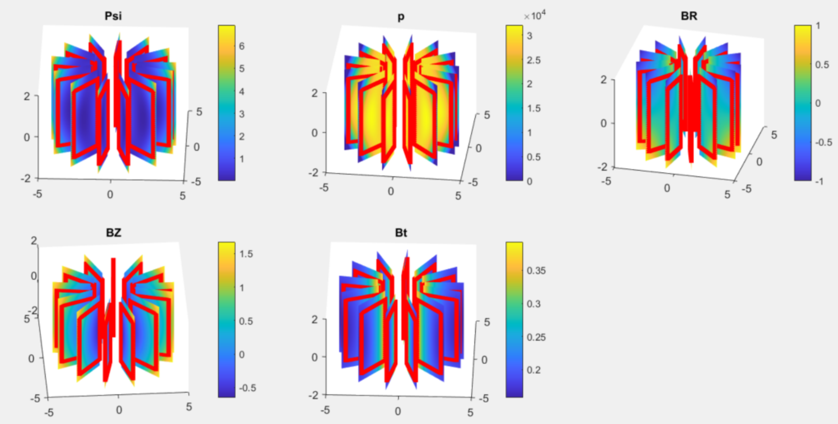
Loss=(Loss1+Loss2+Loss3+Loss4)*α

Condizioni al contorno

La soluzione delle PDE in $\phi = \{0, 2\pi\}$ potrebbe portare a soluzione diverse quando queste, per periodicità, devono essere identiche. Per evitare questo comportamento, occorre aggiungere una quinda funzione di costo:

$$Loss5 = \frac{mean(p_i - p_f)^2}{mean(p_i)^2} + \frac{mean(B_{r,i} - B_{r,f})^2}{mean(B, r_i)^2} + \frac{mean(B_{z,i} - B_{z,f})^2}{mean(B_{z,i})^2} + \frac{mean(B_{t,i} - B_{t,f})^2}{mean(B_{t,i})^2}$$

Risultato atteso:



Risultati

Risultato della rete neurale: