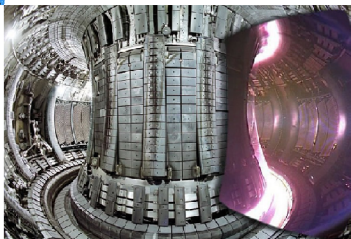


# Réseaux de neurones pour la modélisation du transport turbulent dans les plasmas de fusion.

Juan Redondo Hernan.

Superviseurs: Jonathan Citrin et Frédéric Imbeaux

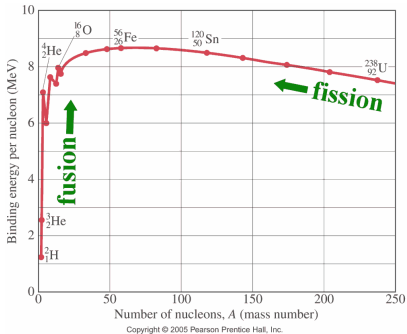


# Introduction

Réproduire l'énergie des étoiles.

Energie de liason par nucleon;

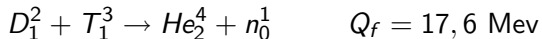
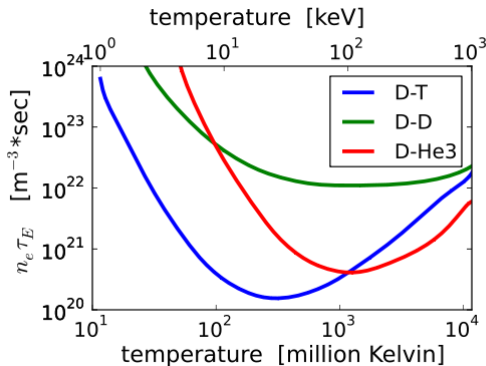
$$B_f = [Zm_p + (A - Z)m_n - m_A]c^2$$



# Introduction

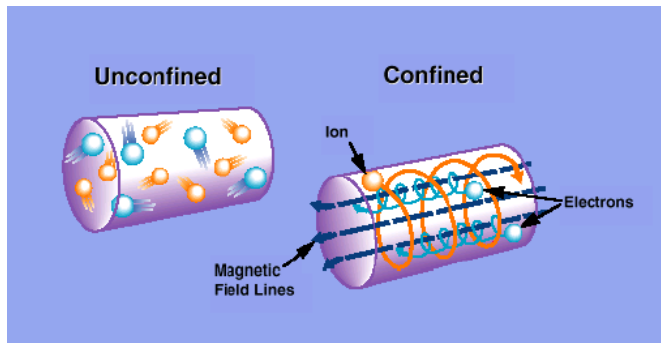
Critère de Lawson.

Condition simple pour ignition:



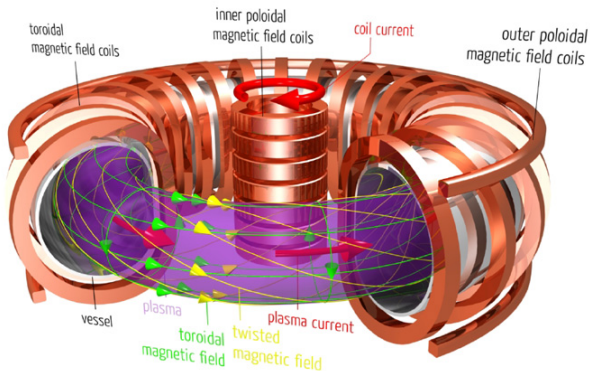
# Introduction

Confinement du plasma par des champs magnetiques.



# Introduction

Concept le plus avancé: tokamak.



Tokamak ITER en construction.

Demonstration des principes: conçu pour une puissance de fusion de 500MW en décharges de 5 minutes.

# Motivation

Transport de particules et d'énergie.

Transport de particules:

$$\frac{\partial n}{\partial t} + \text{div} \cdot \Gamma = \text{Sources} - \text{Puits}$$

$$\Gamma = -D \nabla n + n(V_c + V_t)$$

Transport d'énergie:

$$\frac{\partial(\frac{3}{2}nkT)}{\partial t} + \text{div} \cdot q = \text{Sources} - \text{Puits}$$

$$q = -n\chi \nabla T + \Gamma \frac{5}{2}kT$$

# Motivation

Transport des particules et d'énergie.

Transport de particules:

$$\frac{\partial n}{\partial t} + \text{div} \cdot \Gamma = \text{Sources} - \text{Puits}$$

$$\Gamma = -\mathbf{D} \nabla n + n(\mathbf{V}_c + \mathbf{V}_t)$$

Transport d'énergie:

$$\frac{\partial(\frac{3}{2}nkT)}{\partial t} + \text{div} \cdot q = \text{Sources} - \text{Puits}$$

$$q = -n\chi \nabla T + \Gamma \frac{5}{2}kT$$

# Motivation

Calcul des flux à une distance radiale.

- ▶ Gyrokinetic 5D
  - ▶ Non linéaire, électromagnétique.
  - ▶  $\approx 10^4 - 10^5$  CPUh.
- ▶ Quasilinear 4D (e.g. QuaLiKiz <sup>1</sup>)
  - ▶ Linéarisé, électrostatique.
  - ▶  $\approx 10$  CPUs.
- ▶ Neural network regression of database of QuaLiKiz results
  - ▶ Some CPU  $\mu$ s.
  - ▶ Possibilité de calcul en temps réel basé en premiers principes!

---

<sup>1</sup>Bourdelle C. et al 2007 Phys. Plasmas 14 112501



# Transport turbulent

## Approche quasi-linéaire et électrostatique

Equation de Vlasov (limite électrostatique):

$$\frac{\partial f_s}{\partial t} + \mathbf{v} \frac{\partial f_s}{\partial \mathbf{r}} + \frac{q\mathbf{E}}{m_s} \frac{\partial f_s}{\partial \mathbf{v}} = 0 \quad (1)$$

Quasi-neutralité:

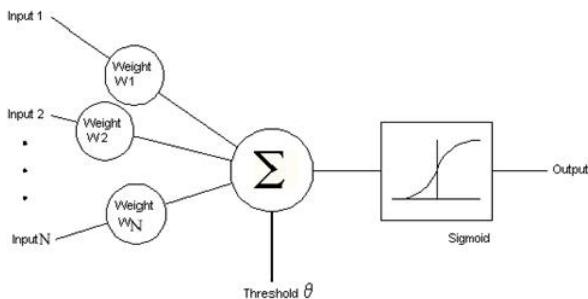
$$\sum_s q_s n_s = 0 \quad (2)$$

- ▶ Différents phénomènes physiques produisent des fluctuations de différente fréquence à l'origine du transport turbulent:
  - ▶ ITG: Ion temperature gradient.
  - ▶ ETG: Electron temperature gradient.
  - ▶ TEM: Trapped electron mode.

# Réseaux de neurones

## Perceptron

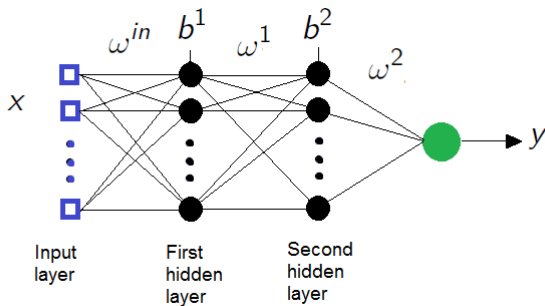
Technique de regression venant de l'intelligence artificielle.



# Réseaux de neurones

## Multilayer perceptron

Approximateur universel!



Les poids  $\omega$  déterminent la fonction de sortie:

$$y = \sum_i^N \omega_i^2 g(b_i^2 + \sum_j^M \omega_{ij}^1 g(b_j^1 + \sum_k^I \omega_{jk}^{in} x_k))$$

# Application en 4 dimensions

## Caracteristiques du réseau

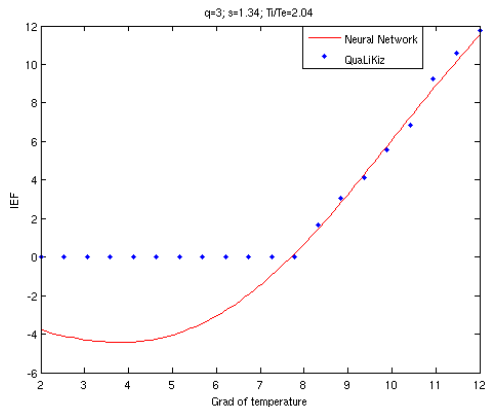
- ▶ Parametres variables:
  - ▶ Facteur de sécurité  $q$ , tournure des lignes de  $\mathbf{B}$ .
  - ▶ Magnetic shear,  $s = r \frac{q'}{q}$ .
  - ▶ Rapport des temperatures ionique et électronique,  $\frac{T_i}{T_e}$ .
  - ▶ Gradient de température: conducteur principal d'instabilités.
- ▶ Architecture du réseau:
  - ▶ Nombre de couches et des noeuds.
  - ▶ Fonction d'activation.
  - ▶ Algorithme d'apprentissage.
  - ▶ Parametre de regularisation  $\lambda$ . Les algorithmes d'apprentissage utilisent comme critère:

$$error = \lambda \frac{1}{N} \sum_{i=1}^N (\text{NN output} - \text{QuaLiKiz result})^2 + (1-\lambda) \frac{1}{n} \sum_{j=1}^n \omega_j^2$$

# Application en 4 dimensions

## Résultats

Flux d'énergie ionique.  $q_i = -n_i \chi_i \nabla T_i + \Gamma_i \frac{5}{2} k T_i$

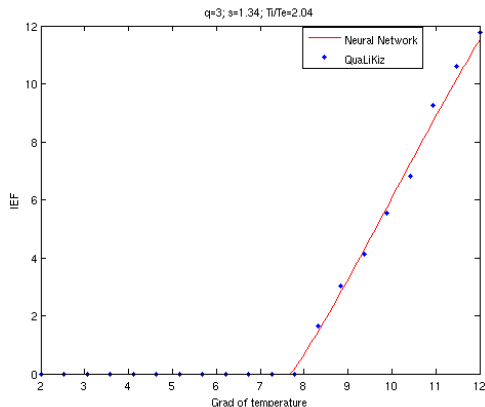


Regression seulement de la partie positive! → Seuil d'instabilité

# Application en 4 dimensions

## Résultats

Flux d'énergie ionique.  $q_i = -n_i \chi_i \nabla T_i + \Gamma_i \frac{5}{2} k T_i$

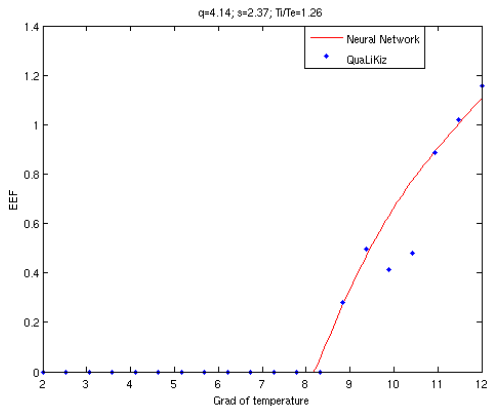


Output du réseau  $< 0 \rightarrow \text{Output}=0$

# Application en 4 dimensions

## Résultats

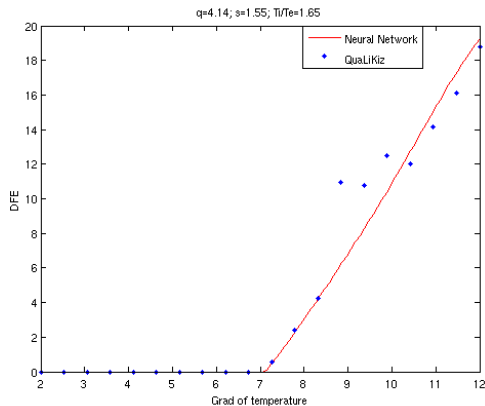
Flux d'énergie électronique.  $q_e = -n_e \chi_e \nabla T_e + \Gamma_e \frac{5}{2} k T_e$



# Application en 4 dimensions

## Résultats

Flux diffusif d'électrons.  $\Gamma_e = -\mathbf{D}_e \nabla n_e + n_e(V_c + V_t)$



Transport de particules et chaleur en temps réel!

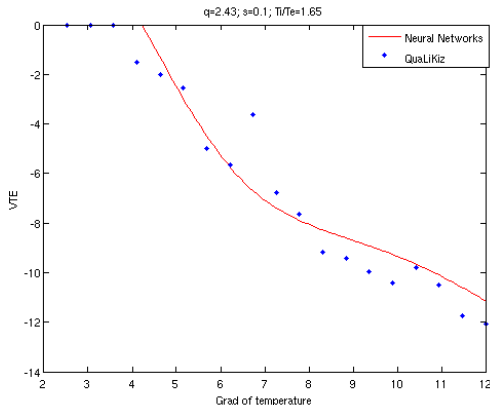


# Application en 4 dimensions

## Résultats

Flux d'électrons dû au pinch thermique.

$$\Gamma_e = -\mathbf{D}_e \nabla \mathbf{n}_e + n_e (V_c + \mathbf{V}_t)$$



Flux de chaleur ionique = 0  $\rightarrow$  Pinch thermique = 0

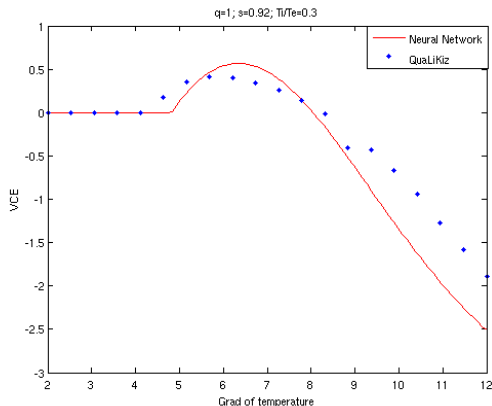
Transport de particules et chaleur en temps réel!

# Application en 4 dimensions

## Résultats

Flux d'électrons dû au pinch de compression.

$$\Gamma_e = -\mathbf{D}_e \nabla \mathbf{n}_e + n_e (\mathbf{V}_c + \mathbf{V}_t)$$

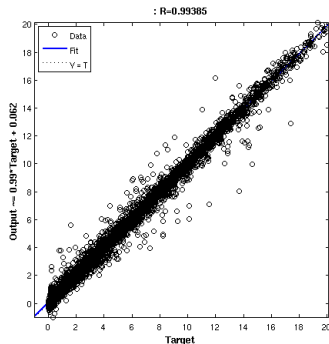


Transport de particules et chaleur en temps réel!

# Application en 4 dimensions

## Méserver la performance

$R$  = pente des points output – target.

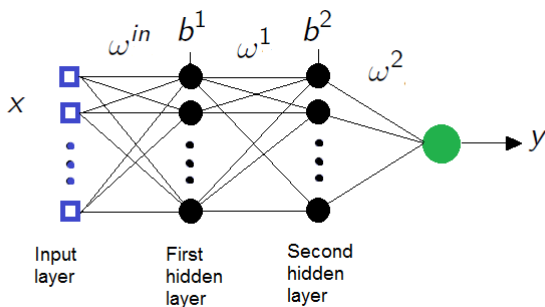


$$\text{Normalised Minimum Square Error} = \frac{1}{N} \frac{\sum (y_i - \text{target}_i)^2}{\frac{(\sum y_i)}{N} \frac{(\sum \text{target}_i)}{N}}$$

# Application en 4 dimensions

Un output

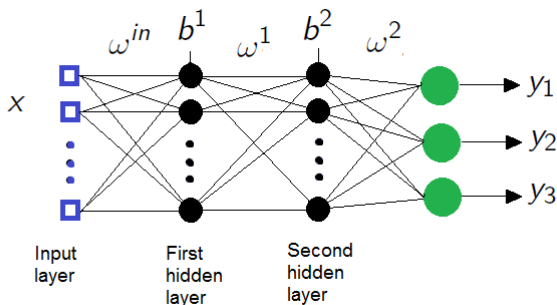
Single output NN	R	NMSE
a) Ion heat flux	0.9939	0.0072
b) Electron heat flux	0.9894	0.0155
c) Electron diffusivity	0.9908	0.0142
d) Electron thermal pinch	0.9527	0.0512
e) Electron compression pinch	0.9697	0.0734



# Application en 4 dimensions

Trois outputs

Three output NN	R	NMSE
f) Ion heat flux	0.9910	1.2696
g) Electron heat flux	0.9809	1.7102
h) Electron diffusivity	0.9911	1.8268



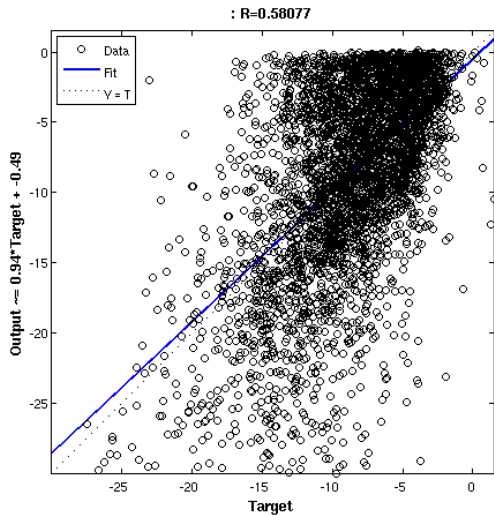
# Application en 4 dimensions

## Five outputs

Five output NN	R	NMSE
i) Ion heat flux	0.9910	1.1438
j) Electron heat flux	0.9704	1.7102
k) Electron diffusivity	0.9845	1.6575
l) Electron compression pinch	0.9754	3.1090
m) Electron thermal pinch	0.58077	1.1027

# Application en 4 dimensions

Thermal pinch in five dimensions NN



# Extrapolation à 10 dimensions

## Questions à résoudre.

Le nombre de points total est exponentiel avec la dimension  $\approx N^d$ .  
Avec cette méthodologie nous pouvons aller jusqu'à 10 dimensions.

- ▶ Contraintes techniques.
  - ▶ Heures de calcul haut performance disponibles sur HELIOS  $\approx 10^5 - 10^6 \text{ CPUh} \rightarrow \approx 2 \cdot 10^8$  points dans l'espace des parametres.
  - ▶ Optimisation de l'organisation des simulations (contrainte de mémoire).
  - ▶ Debug de la version du code QuaLiKiz sur HELIOS.
- ▶ Complexité du problème physique
  - ▶ Manque de connaissance des seuils d'instabilité (réduire le nombre de points calculés).
  - ▶ Non linéarités du problème (construction postérieure du réseau).



# Conclusions

- ▶ Transport des particules et de chaleur en temps réel pour la première fois. Future implementation avec le code RAPTOR<sup>2</sup>.
- ▶ Bons résultats à basse dimension pour des problèmes généraux de transport.
- ▶ Début du travail en 10 dimensions.
- ▶ Besoin de techniques de réduction de dimensionalité pour de dimensions plus grandes.

---

<sup>2</sup>F. Felici and O. Sauter 2012 Plasma Phys. Control. Fusion 54 025002.

# 10 paramètres

## Base de données

Début d'une base de données avec les paramètres suivants (adimensionnels):

Paramètre	# points	Interval
Gradient de température ionique	8	[2, 10]
Gradient de température électronique	8	[2, 10]
Gradient de densité électronique	9	[-5, 7]
Rapport $T_i/T_e$	7	[0.5, 2]
Densité électronique	4	[0.5, 20]
Facteur de sécurité $q$	8	[1, 6]
Magnetic shear	9	[-1, 4]
Alpha	6	[0, 0.6]
% Beryllium	4	[0, 0.75]
Distance au centre du plasma	6	[0.15, 0.9]