Contr. Fusion and Plasma Phys. (Proc. 24th Eur. Conf. Berchtesgaden, 1997), Vol. 21A, Part II, E.P.S. (1997) 753

Solving Threedimensional Plasma Equilibria with a Neural Network Technique

B.Ph. van Milligen, J.A. Jiménez and V. Tribaldos
Asociación EURATOM-CIEMAT para Fusión,
Avda. Complutense 22, 28040 Madrid, Spain

Abstract

A new method to solve differential equations, based on neural networks, has been developed recently. The method is generally applicable to n^{th} order partial differential equations on a finite domain with boundary conditions [1]. The method avoids the use of finite differences since all derivatives are programmed explicitly. The use of a neural network guarantees a compact representation of any type of multivariate continuous solution [2, 3]. Due to this compact representation (using a low number of free parameters), and due to the fact that the minimizing process involved can be entirely formulated using analytic derivatives and these can be evaluated using a highly parallel algorithm, the technique is potentially fast, especially for problems that have to be solved repeatedly with similar boundary conditions.

In the present work, the technique is applied to the solution of the 3-D ideal magnetohydrodynamic (MHD) plasma equilibrium problem. A torsatron has been chosen for the first 3-D equilibria computed.

1. Introduction

We apply a recently introduced method for the solution of differential equations [1], based on MLP-1 type neural networks, to the solution of stellarator equilibria [4]. The method consists in representing the solution to the differential equation (i.e. some spatial potentials whose derivatives give the magnetic field) by a neural network mapping. The differential equation is solved in real space. All derivatives required are taken analytically, and the learning algorithm is intrinsically parallel. Therefore the method shows potential for the development of fast equilibrium codes.

2. Magnetic field representation

We will solve the ideal magnetohydrodynamic equilibrium problem, given by $\vec{j} \times \vec{B} = \vec{\nabla} p$. The often used representation of the magnetic field (Clebsch-Gordon):

$$\vec{B} = \vec{\nabla}\psi \times \vec{\nabla}\chi \tag{1}$$

is not directly applicable when we wish to solve magnetic equilibria by means of a neural network representation of the scalar potentials, because one of the potentials is multivalued. Neural networks cannot handle such quantities, since they are one-to-one mappings.

Therefore, we return to Helmholtz's theorem which states that any vector field can be split in a divergence-free and a rotation-free part:

$$\vec{B} = \vec{\nabla}\psi \times \vec{\nabla}\chi + \vec{\nabla}\xi \qquad (2)$$

where $\nabla^2 \xi = 0$ since the field is divergence-free.

The choice ξ = const. that leads to the Clebsch-Gordon representation is only one of an infinity of possible "gauges". We may therefore choose another "gauge" that allows us to eliminate the multivaluedness of the potentials in the plasma region. We select precisely the potential ξ that reproduces the magnetic field on the axis (i.e. $\vec{B}\big|_{ax} = \vec{\nabla}\xi$). Additionally, we require that ψ be a flux function, or: $\vec{\nabla}\psi \cdot \vec{\nabla}\xi = 0$.

3. Programmatic details

The scalar potentials are represented by a MLP-1 neural network: its inputs are the spatial coordinates (R,ϕ,Z); its outputs are the three scalar potentials $\{\psi, \chi, \xi\}$ of Section 2.

The deviation from zero of the differential equation, at a number of points in the plasma interior, as well as the deviation of the boundary conditions, at some points on the boundary, are used to define a penalty functional, E, which is minimized using an advanced gradient-descent algorithm by varying the neural network weights. This procedure is described in [1, 4]. Here we only wish to stress that the algorithm for calculating the penalty functional and its gradients is *highly parallel* (more than 97%), and important speedups are expected on massively parallel computers.

4. An equilibrium of the TJ-IU torsatron

In this section, we describe the results obtained for a low-β equilibrium of the Spanish TJ-IU torsatron (l =1, m = 6, major radius R_0 = 0.6 m, average minor radius <a> = 0.1 m, toroidal field B_T = 0.5 - 0.67 T).

First, we wished to ascertain that the representation we use is capable of reproducing the magnetic field and pressure corresponding to this equilibrium. In previous publications, we have already shown how the magnetic flux of such equilibria can be fitted [5, 6]. The equilibrium code VMEC [7] was used to produce a data file containing the magnetic field and plasma pressure for a given equilibrium on a regular grid (N = 8500 data points over one period: $0 \le \phi < \pi/3$). We then trained the network to reproduce the magnetic field and the pressure, p. The penalty functional E to be minimized consists of two terms:

$$E = \gamma_1 \sum_{n=1}^{N} \left[\vec{B} - \vec{B}_{exact}\right]^2 + \gamma_2 \sum_{n=1}^{N} \left[p - p_{exact}\right]^2, \qquad (3)$$

which are evaluated in the plasma region, Ω. The two weighting factors γ_i are chosen such that both terms contribute in about the same amount to E. To relate the pressure to the scalar potentials, we used the expression

$$p = p(\psi) = 1.5 \cdot 10^{-2} (10^{-2} - \psi). \qquad (4)$$

Table I gives the result of the training procedure for each of the field components and the pressure for networks with K = 63, 127 and 255 hidden nodes, respectively.

The errors in the magnetic field and the pressure are reasonable (of the order of or below 1%). We conclude that it is possible to represent the field of a typical torsatron equilibrium with a neural network.

Table I

Quantity	*Average*	*Spread*	*Error (K=63)*	*Error (K=127)*	*Error (K=255)*
Iterations			5000	7000	13000
B_R	0.000362	0.103	0.00159	0.000851	0.000651
B_ϕ	0.521	0.108	0.00136	0.000713	0.000494
B_Z	0.00857	0.118	0.00188	0.000864	0.000620
p	$7.44\cdot10^{-5}$	$4.34\cdot10^{-5}$	$0.119\cdot10^{-5}$	$0.0639\cdot10^{-5}$	$0.0508\cdot10^{-5}$

We then solved the equilibrium by the method described in [1, 4]. Initial network weights were taken randomly. The boundary surface ($\partial\Omega$) was specified on a 48×12 equidistant (in VMEC co-ordinates) grid in the toroidal angle ϕ and the poloidal angle θ (N_{bound} = 576). The internal points of the region Ω were selected with ϕ again in the range $0 \le \phi \le \pi/3$, their total number being N = 4250. Finally, the pressure was specified on a chord of 150 points at ϕ = 0 and again at $\phi = \pi/3$ due to stellarator symmetry. The pressure profile was the one given in Eq. (4). The total toroidal flux was set at $\Phi_0 = 1.99\cdot10^{-2}$ T m^2 and the net toroidal current was kept equal to zero. The network we used had K = 127 hidden nodes.

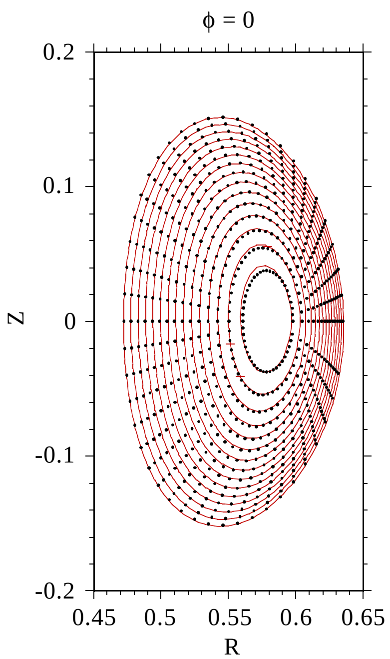

Fig. 1 - The ψ-surfaces of a low-β equilibrium of the TJ-IU stellarator at ϕ = 0°. Dots indicate the position of the flux surfaces as calculated by VMEC, and lines indicate the flux surfaces calculated by a neural network differential equation solver.

The calculations converged to a solution with an average RMS force balance error $\sqrt{\langle|\varepsilon|^2\rangle}$ of 8.5% (where $|\varepsilon| = \left|\vec{j}\times\vec{B} - \vec{\nabla}p\right| / \left|\vec{\nabla}p\right|$). Fig. 1 shows the ψ-surfaces of the equilibrium at ϕ = 0°, compared with results from VMEC. There is a significant difference with the VMEC results, which is not surprising at error levels as high as this. Fig. 2 shows the spatial distribution of the force balance error $|\varepsilon|$, while Fig. 3 shows its numerator. Observe that the force balance (i.e. $\left|\vec{j}\times\vec{B} - \vec{\nabla}p\right|$) does not show a maximum at the magnetic axis, due to the fact that the chosen magnetic field representation does not have a singularity there.

5 Conclusions

In this paper we have shown that the three-dimensional plasma equilibrium problem can in principle be solved by the method of solving differential equations with the aid of a neural network. To do so, we first introduced a new representation for the magnetic field, which differs from the usual Clebsch-Gordon representation by a choice of gauge, and in which the scalar potentials are not multivalued. The magnetic field is represented by three instead of two scalar potentials.

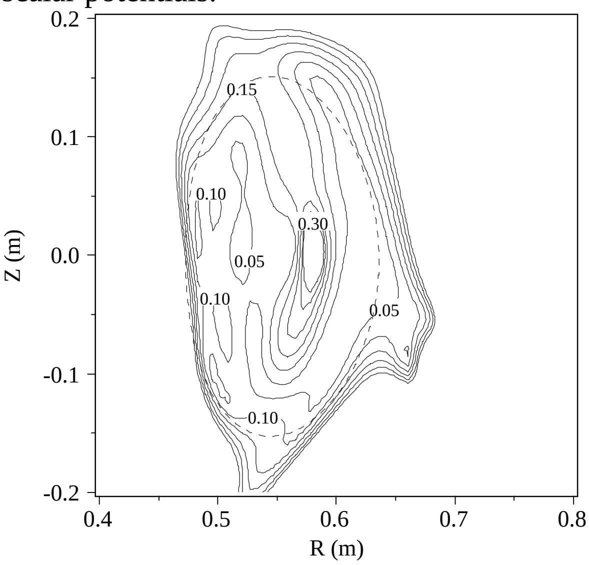

Fig. 2 - The force balance error $|\varepsilon|$ at $\phi = 0°$.

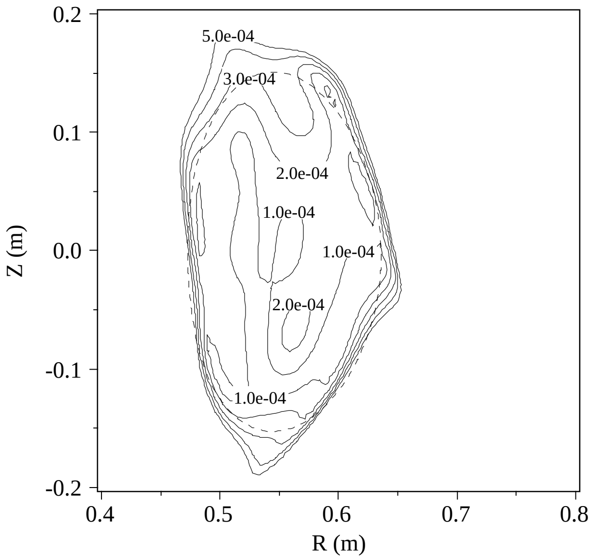

Fig. 3 - The force balance at $\phi = 0°$.

The use of a neural network to represent the scalar potentials is shown to provide reasonable accuracy in both the magnetic field and the pressure. That is to say, reasonable for the purposes of e.g. experimental data interpretation; for the purposes of the calculation of Poincaré sections (by following field lines) or stability limits, the accuracy achieved here is in general not sufficient. However, there is no limit *in principle* on the accuracy that can be reached - only a practical limit in terms of calculation time.

Finally we emphasize that, independently of the actual representation used, equilibrium solvers based on this neural network technique are by nature parallel, and large acceleration factors are to be expected on highly parallel computers.

References

[1] B.Ph. van Milligen, V. Tribaldos, J.A. Jiménez, Phys. Rev. Lett. **75**, 3594 (1995)

[2] K. Hornick, Neural Networks **4**, 251 (1991)

[3] K. Funahashi, Neural Networks **2** (3), 183 (1989)

[4] B.Ph. van Milligen, J.A. Jiménez and V. Tribaldos, *Solving differential equations using a neural network: Application to three-dimensional plasma equilibria*, Submitted to J. Comput. Physics (1997)

[5] V. Tribaldos, B.Ph. van Milligen, Nucl. Fusion **36**, 283 (1996)

[6] V. Tribaldos and B.Ph. van Milligen, Rev. Sci. Instrum. **68**, 931 (1997)

[7] S. P. Hirshman and D.K. Lee, Comp. Phys. Comm. **39**, 161 (1986)