

# Solving Equilibria with a Neural Network

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

## Abstract

A general method to solve differential equations that was introduced recently, based on the use of MLP-1 type neural networks, is applied to the fast solution of ideal magnetohydrodynamic (MHD) plasma equilibria. Major advantages of the method are: (1) it does not presuppose any particular topology of the solution (e.g. nested flux surfaces); (2) the equations are solved in real space (without coordinate transformations); (3) finite differences are not used; and (4) it is potentially fast. We provide a proof-of-principle of the method by solving a D-shaped tokamak equilibrium, and discuss the application to three-dimensional (stellarator) equilibria. The technique especially promises to be fast for problems that have to be solved repeatedly with similar boundary conditions, and perhaps makes on-line plasma equilibrium calculations feasible for stellarators.

## Description of the method

Before discussing the plasma equilibrium problem, we briefly introduce the general method for solving differential equations mentioned in the Abstract. This material has been presented earlier in Ref. [1]. The problem to which this method is applicable is of the form:

$$D(\vec{f}(\vec{x})) = \vec{0} \quad (1)$$

where  $D$  is a differential operator and  $\vec{f}$  is a multivariate function of  $\vec{x}$ , that satisfies appropriate boundary conditions so that it is a unique solution to Eq. (1);

The new method consists in *approximating* the solution  $\vec{f}(\vec{x})$  with a neural network. The training process of the neural network is a little different from the usual procedure (which consists in training the network by presenting it with examples of a *known* solution): here, we adjust the network weights by means of a variant of the error backpropagation algorithm in order to minimize Eq. (1) on the whole domain within the given boundaries. In other words, the minimization of the sum of squares of the difference between the network output and some examples is replaced by the minimization of a penalty functional that depends on the equation to be solved.

We approximate the solution using a network of the Multilayer Perceptron type with one hidden layer (MLP-1). This type of network is capable of approximating an arbitrary continuous function to within arbitrary precision, provided sufficient hidden nodes are present [2-5].

Denoting the network inputs by  $\vec{x} = \{x_1, x_2, \dots, x_J\}$ , the hidden layer nodes by  $\vec{y} = \{y_1, y_2, \dots, y_K\}$ , and the network outputs by  $\vec{z} = \{z_1, z_2, \dots, z_L\}$ , we define

$$y_k = \sigma \left[ \sum_{j=1}^{J+1} v_{kj} x_j \right] \quad (2a)$$

$$z_l = \sum_{k=1}^{K+1} w_{lk} y_k \quad (2b)$$

where  $x_{J+1}$  and  $y_{K+1}$  are so-called "bias units" with a fixed value of 1,  $v$  and  $w$  are referred to as input and output weights, respectively, and  $\sigma(x)$  is the sigmoid function:

$$\sigma(x) = \tanh\left(\frac{x}{2}\right) \quad (3)$$

We intend that the network output  $\vec{z}$  should provide an approximation to the exact solution of Eq. (1),  $\vec{f}(\vec{x})$ . For that purpose, we define a penalty functional  $E$ :

$$E = [E_{d.e.}]^2 + [E_{b.c.}]^2 \quad (4)$$

where  $E_{d.e.} \equiv D(\vec{z})$  is the l.h.s. of the differential equation Eq. (1) applied to  $\vec{z}$ , and  $E_{b.c.}$  schematically represents a functional that is zero if and only if the boundary conditions are satisfied. For  $\vec{z} = \vec{f}$ ,  $E = 0$ , and since the solution is unique, minimizing Eq. (4) provides an approximation to  $\vec{f}$ .

$E(\vec{x})$ , as given by Eq. (4), is evaluated for a (sufficiently large) number  $N$  of values  $\{x(i), i=1, \dots, N\} \in \Omega$ , where  $\Omega$  is a finite domain in  $\mathbb{R}^J$ . The total error  $E_{tot}$  is defined as the sum of all the values of  $E(x(i))$ . Minimization of  $E_{tot}$  will then provide a solution to Eq. (1) on  $\Omega$ , satisfying the boundary conditions as specified, provided (a) the number of hidden nodes is sufficient; (b) the covering of this domain by  $\{x(i)\}$  is well-distributed so that the solution is well-described by the function values  $\{f(x(i))\}$ ; and (c) this covering is dense enough to avoid overfitting ( $N$  is larger than the number of free parameters, or network weights). The number of hidden nodes then determines the maximum attainable accuracy.

For the minimization procedure a standard quasi-Newton gradient-descent algorithm is used. Note that due to the simplicity of the network specified by Eq. (2), it is generally possible to give an analytic expression for the evaluation of the differential operator  $D$  for a given set of network weights. Thus, the gradients of  $E$  (Eq. (4)) with respect to the weights  $v$  and  $w$  can be expressed analytically, which improves and accelerates the minimization procedure.

### A D-shaped Tokamak Equilibrium

The problem of the fixed-boundary tokamak equilibrium is stated as follows: determine the poloidal flux function  $\psi(R, Z)$  such that the Grad-Shafranov equation is satisfied [6], or

$$E_1^2 = \sum_{n=1}^N \left[ \Delta^* \psi + \mu_0 R^2 \frac{\partial p}{\partial \psi} + F \frac{\partial F}{\partial \psi} \right]_{\Omega}^2 = 0 \quad (5)$$

where the Grad-Shafranov operator is defined as:

$$\Delta^* \psi = R \frac{\partial}{\partial R} \left[ \frac{1}{R} \frac{\partial \psi}{\partial R} \right] + \frac{\partial^2 \psi}{\partial Z^2}$$

and where  $p(\psi)$  is the pressure profile and  $F(\psi)$  is related to the poloidal current profile. Eq. (5) is evaluated for  $N$  interior points of the plasma cross-section ( $\Omega$ ). This equation, describing toroidally symmetric ideal MHD equilibria, has a unique solution when adequate boundary conditions are imposed and the two source profiles  $p(\psi)$  and  $F(\psi)$  are specified. To eliminate the arbitrary integration constant in  $\psi$ , and to give to  $\psi$  its physical meaning of a magnetic field integrated over a surface, we require that there is a magnetic axis,  $(R_{ax}, Z_{ax})$ , where the flux  $\psi$  attains its minimum, being equal to 0:

$$E_2^2 = [\psi(R_{ax}, Z_{ax}) - 0]^2 = 0 \quad (6)$$

It should be remarked that the location of the magnetic axis  $(R_{ax}, Z_{ax})$  is not known *a priori*. Thus, every  $N_{eval}$  iterations the minimum of the flux function as given by the network is determined and the position of this minimum is taken to be  $(R_{ax}, Z_{ax})$ .

The boundary condition for the differential equation is  $\vec{B} \cdot \vec{n} = 0$  at the plasma boundary (fixed-boundary problem with a perfectly conducting wall), where  $\vec{B}$  is the magnetic field and  $\vec{n}$  is the normal to the boundary. However, we further require that the poloidal field generated corresponds to a pre-specified toroidal plasma current  $I_p$ , or

equivalently we require that the flux on the boundary has the preset value of  $\psi_{bound}$ . Since this automatically implies  $\vec{B} \cdot \vec{n} = 0$  there is no need to impose the latter separately:

$$E_3^2 = \sum_{m=1}^{N_{bound}} [\psi - \psi_{bound}]_{\partial\Omega}^2 = 0 \quad (7)$$

which is evaluated for  $N_{bound}$  boundary points (on  $\partial\Omega$ ). The penalty functional  $E_{tot}$  for minimization is the weighted sum of the three terms of Eqs. (5-7).

To check the solution obtained with the new method we have compared it with the solution generated by the plasma equilibrium solver VMEC [7-9] for a particular case. VMEC calculates an equilibrium, assuming nested flux surfaces, from input profiles  $p(\psi)$  and  $q(\psi)$ , where the safety factor  $q$  is related to the current distribution. The equilibrium problem expressed by Eqs. (5-7) is slightly different, since it requires the profiles  $p(\psi)$  and  $F(\psi)$  as input. To obtain the same equilibrium with both methods, we have passed the VMEC output profile of  $F(\psi)$  on to the neural network solver along with the input profile of  $p(\psi)$ . Both profiles are given in terms of polynomial fit coefficients to the VMEC profiles.

We have selected a D-shaped plasma (JET-like [10]) with  $R_0 = 3.1$  m, horizontal minor radius  $a = 1.35$  m, elongation  $\kappa = 1.66$ , a safety factor at the boundary of  $q_a = 4$ , and a total average normalized pressure of  $\langle\beta\rangle = 4.26$  %.

Using a network with  $K = 31$  hidden nodes and making the identifications  $x_1 = R - R_0$ ,  $x_2 = Z$  (two input nodes) and  $z_1 = \psi$  (one output node), the neural network solver converges in 1000 iterations to a solution satisfying the force equilibrium condition,  $\vec{j} \times \vec{B} - \nabla p = 0$ , to within 7 % on average (with respect to the pressure gradient). The accompanying figure shows the flux contours as obtained from the network solver and the location of the corresponding VMEC flux surfaces. A more accurate reconstruction can be obtained, if required, by letting the iteration process continue and by increasing the number of hidden nodes,  $K$ . Force balance errors of the order of 1% are easily obtained. Nevertheless the fast reconstruction presented is already quite satisfactory for most purposes.

## Prospects

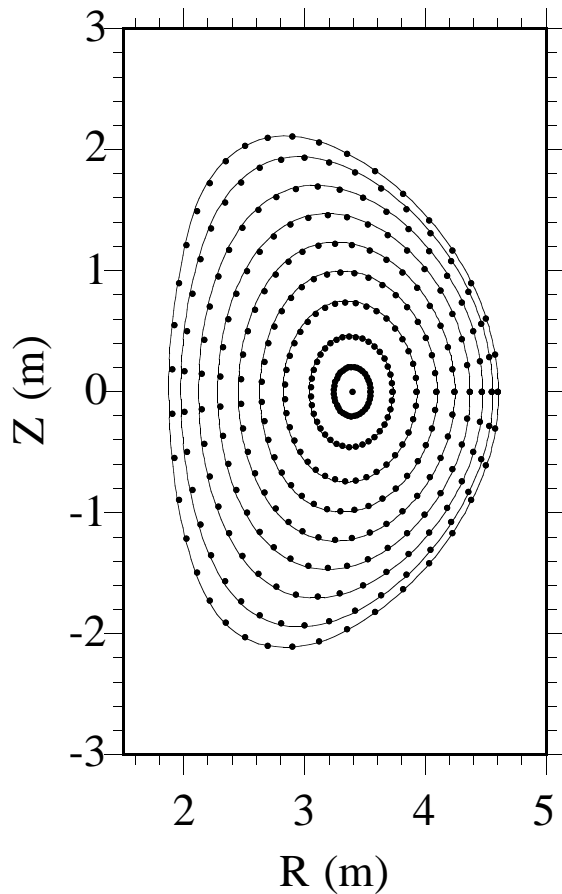
Encouraged by these results, the calculation of free-boundary equilibria and fully three-dimensional (stellarator) equilibria is planned in the near future. With respect to the latter we remark that an excellent representation of the flux of a Helic stellarator equilibrium (TJ-II) has already been obtained with only  $K = 255$  hidden nodes [11], suggesting that the three-dimensional equilibrium problem is tractable by this method. In this respect we note that, independently of this solution method, a general advantage of representing a multivariate function by means of a neural network is storage efficiency and evaluation speed.

## Conclusions

The present relatively simple cases provide a proof-of-principle of the power and possibilities of the new solution method. We stress the major advantages of the method for plasma equilibrium calculations, which are the following: (1) It is capable of solving the equations in real space, i.e. without complicated and costly coordinate transformations. This is especially advantageous for the three-dimensional problem. (2) It does not presuppose any particular topology of the solution (e.g. nested flux surfaces). Thus the possibility of easily incorporating X-points or island regions exists, insofar as permitted by the equations. (3) The use of finite differences is avoided. (4) It promises to be fast for problems that have to be solved repeatedly with similar boundary conditions.

## Acknowledgments

This work was made possible by Commission of the European Communities Bursary ERB4001GT921485. The authors would like to thank their colleagues at the Asociación EURATOM-CIEMAT for a very pleasant cooperation.



Flux surfaces as generated by the VMEC equilibrium solver and the neural network solver. Dots indicate the location of the VMEC flux surfaces; lines are contour levels of the neural network output at the same flux values. The correspondence is quite good, especially considering the small number of hidden nodes of the network ( $K = 31$ ).

## References

- [1] B.Ph. van Milligen, V. Tribaldos, J.A. Jiménez, *A Neural Network Differential Equation and Plasma Equilibrium Solver*, Submitted to Phys. Rev. Lett. (1995)
- [2] K. Hornick, M. Stinchombe, H. White, *Neural Networks* **2** (1989) 359
- [3] K. Hornick, *Neural Networks* **4** (1991) 251
- [4] K. Funahashi, *Neural Networks* **2** (3) (1989) 183
- [5] Tarun Khanna, *Foundations of neural networks*, Addison-Wesley, Reading MA (1990)
- [6] J.P. Freidberg, *Ideal Magnetohydrodynamics*, Plenum Press, New York, 1987 (Chapter 6 and references therein)
- [7] S. P. Hirshman and J.C. Whitson, *Phys. Fluids* **26** (1983) 3553
- [8] S. P. Hirshman and D.K. Lee, *Comp. Phys. Comm.* **39** (1986) 161
- [9] S. P. Hirshman and O. Betancourt, *J. Comp. Phys.* **96** (1990) 99
- [10] *The JET Project: Design proposal of the Joint European Torus*, EUR-5516e (EUR-JET-R5) Commission of the European Communities.
- [11] V. Tribaldos and B.Ph. van Milligen, *Electron Cyclotron Emission calculations for TJ-II stellarator*, Submitted for publication to Nuclear Fusion (1995); and V. Tribaldos, *ECE studies for TJ-II stellarator*, This Conference