

Comparison of neural network and polynomial models for the approximation of nonlinear and anisotropic ferromagnetic materials

H. Vande Sande and K. Hameyer

Abstract: Nonlinear magnetic problems can be solved efficiently by applying the iterative Newton–Raphson method in a finite-element framework. At the beginning of each nonlinear iteration, the magnetic reluctivity and the differential reluctivity must be determined in every element of the mesh. As a consequence, the time required for building the linear system to be solved, strongly depends on the evaluation time of the applied material models. Moreover, the accuracy and the smoothness of the material models affect the convergence rate of the Newton–Raphson method. Three methods for representing material properties are compared from a computational point of view. The magnetisation curves of nonlinear isotropic ferromagnetic materials are commonly approximated by cubic splines. However, it is observed that polynomials and feedforward neural networks have also been adopted for this purpose. It is shown that these, although having some attractive properties, should not be applied for approximating magnetisation curves. The same holds for more complex relations, such as the anisotropic reluctivity curves of grain-oriented steel. Although feedforward neural networks become more appealing for these types of mappings, they do not offer a computational advantage compared with the bicubic spline representation.

1 Introduction

Within the finite element framework, nonlinear magnetic problems are often solved using the nonlinear Newton–Raphson iteration scheme. Therefore at each iteration, the contribution of all elements to the stiffness matrix K and the Jacobian matrix J must be computed, as explained in Silvester *et al.* [1]. For the isotropic case, the elementary stiffness matrix is proportional to the reluctivity v [Am/Vs], while the elementary Jacobian matrix is proportional to the derivative of v with respect to the square of the flux density B [Vs/m²]. For the anisotropic case, the reluctivity and the differential reluctivity are second-order tensors having two or three independent entries, depending on the dimension of the problem. As a consequence, the time required for building the complete stiffness and Jacobian matrices strongly depends on the speed by which these can be evaluated. Obviously, this is related to the mathematical representation of the magnetic properties. Moreover, the quality of the material model also influences the convergence of the nonlinear iteration scheme, as nonsmooth representations might cause numerical oscillations in the convergence process.

Polynomials have since long been used for modelling nonlinear functions. However, they fail to give suitable approximations for strongly nonlinear functional mappings. Carefully trained feedforward neural networks provide an alternative in those cases. Over the past decade, the popularity of feedforward neural networks for modelling

nonlinear phenomena has increased. They have been adopted for modelling magnetisation curves in Patecki *et al.* [2]. In the paper, it is shown that polynomials and feedforward neural networks do not offer any improvement compared with the commonly applied cubic spline interpolation technique from a computational point of view. It holds particularly for the representation of the magnetisation curve of an isotropic material, but even for modelling anisotropic properties it is not encouraged. Within this context, the properties of feedforward neural networks, polynomials and cubic spline interpolants are compared.

2 Material data

The models are compared for two distinct cases:

- The nonlinear magnetisation curve of an isotropic steel. It is a smooth univariate functional mapping $v = f(B^2)$.
- The nonlinear and anisotropic reluctivity curves of a grain-oriented steel (Fig. 1), according to Shirkoohi *et al.* [3]. It is a smooth bivariate mapping $v = f(B, \theta)$ with θ the magnetisation angle [°].

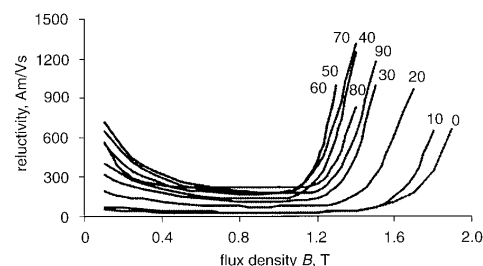


Fig. 1 Nonlinear and anisotropic reluctivity of grain-oriented steel

© IEE, 2002

IEE Proceedings online no. 20020622

DOI: 10.1049/ip-smt:20020622

Paper first received 14th March 2002 and in revised form 11th July 2002

The authors are with the Dept. ESAT, Div. ELECTA, Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, Heverlee-Leuven 3001, Belgium

For accuracy, the reluctivity is transformed in advance by a base-ten logarithm, followed by a normalisation of all variables.

3 Material models

3.1 Polynomials

The nonlinear magnetisation curve $v=f(B^2)$ can be modelled by a polynomial of order N

$$v = \sum_{n=0}^N a_n (B^2)^n \quad (1)$$

This polynomial is most efficiently evaluated by applying Horner's rule, as explained in Press *et al.* [4]. Its evaluation requires $2N$ floating-point operations (FLOPs). The model has $N+1$ degrees of freedom (DOFs). The derivative of (1) is evaluated within $2N-2$ flops. The number of flops is a crude measure for the computation time, as it ignores subscripting, memory traffic and other overheads while executing a program. It is mentioned here as it gives a first indication of an algorithm's efficiency.

The nonlinear and anisotropic magnetisation curves $v=f(B, \theta)$ can be modelled by a two-dimensional polynomial of order N

$$v = \sum_{n=0}^N \left(\sum_{m=0}^{N-n} a_{n,m} B^m \right) \theta^n \quad (2)$$

This polynomial is evaluated similar to Horner's rule within N^2+4N-1 FLOPs and it contains $(N+1)(N+2)/2$ DOFs.

3.2 Cubic splines

In computational magnetics it is convenient to represent magnetisation curves by cubic splines. Using splines avoids the necessity of higher-order polynomials over the entire input range. In between two successive data points or nodes, the cubic spline is a third-order polynomial, having coefficients that depend on the value of B^2 . If the spline is evaluated using the algorithm provided by Press *et al.* [4], 21 FLOPs are performed. The computation of its derivative requires 21 FLOPs as well.

Bivariate functions such as the anisotropic reluctivity curves can be modelled by bicubic splines, as in Dierckx [5]. They can be considered as a one-dimensional cubic spline having coefficients which are themselves cubic splines, depending on the other dimension. Hence, their evaluation is performed by repeatedly evaluating a set of one-dimensional cubic splines. Five cubic spline evaluations are required for evaluating a bicubic spline.

3.3 Feedforward neural networks

As explained in Bishop [6], two-layer feedforward neural networks can approximate any continuous functional mapping arbitrary well. Among the large amount of feedforward neural networks that can be applied for regression purposes, the two-layer perceptron with differentiable sigmoid activation functions is selected for comparison. For both problems, this type of perceptron is mathematically expressed as

$$v = \sum_{j=1}^M q_{1j} \phi(p_{j1} B^2 + p_{j0}) + q_{10} \quad (3)$$

and

$$v = \sum_{j=1}^M q_{1j} \phi(p_{j2} B + p_{j1} \theta + p_{j0}) + q_{10} \quad (4)$$

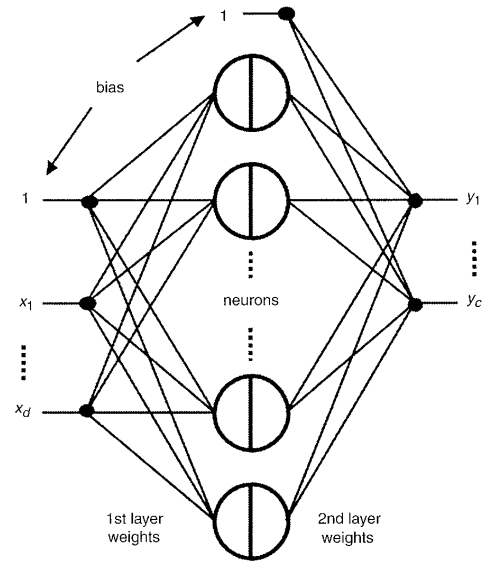


Fig. 2 Graphical interpretation of feedforward neural network

respectively, with M the number of neurons, p_{ji} the first-layer weight connecting input i with neuron j and q_{ij} the second-layer weight connecting neuron j with the output (Fig. 2). The activation function ϕ is given by a logistic sigmoid

$$\phi(a) = \frac{1}{1 + e^{-a}} \quad (5)$$

and represents the nonlinear transformation units or neurons in the network. Equation (3) has $3M+1$ DOFs and is evaluated in $7M$ FLOPs. Equation (4) is evaluated in $9M$ FLOPs and contains $4M+1$ DOFs. The derivative of (3) is given by

$$\frac{dv}{dB^2} = \sum_{j=1}^M q_{1j} p_{j1} \phi(a_j) [1 - \phi(a_j)] \quad (6)$$

$$- a_j = p_{j1} B^2 + p_{j0}$$

requiring $9M-1$ FLOPs to be evaluated.

4 Determination of coefficients

The coefficients of the polynomials in (1) are obtained by solving an over-determined system of equations using the least squares method, without constraints, as described in Golub *et al.* [7]. Measurement errors are smoothed out automatically. However, by applying constraints on the shape of the polynomials one can further improve the smoothness at the expense of a higher sum-of-squares error.

A direct method for determining the cubic spline coefficients is explained in Press *et al.* [4]. More intelligent algorithms allow for optimising the number and the location of the nodes for a given set of data points. They can be smoothed as well, as explained in Dierckx [5]. Pahner *et al.* [8] present a stochastic method for smoothing splines such that the convergence rate of the Newton-Raphson algorithm is optimised.

The perceptron is trained by minimising the sum-of-squares error

$$E(p, q) = \sum_{d=1}^D (v_d^{meas}(p, q) - v_d^{neur})^2 \quad (7)$$

with respect to the network weights. Here, v_d^{net} is the network output and v_d^{meas} the measured value for measurement d . The sum-of-squares error represents a surface in a multidimensional parameter space. Due to the nonconvexity of the activation function the error surface is nonconvex too. Hence, the minimisation algorithm may get trapped in a local minimum of E . Therefore several networks are trained and the best one is retained for the following discussion.

5 Comparison

The polynomial and perceptron models have been determined for both data sets $r=f(B^2)$ and $r=f(B, \theta)$ without imposing constraints on the smoothness of the curves. A measure for the smoothness of the model, as presented in Bishop [6], is given by

$$C = \sum_{d=1}^D \sum_{i=1}^I \left(\frac{\partial^2 y}{\partial x_i^2} \Big|_d \right)^2 \quad (8)$$

with I the number of inputs, y the model's output and x_i the i -th input. C is called the curvature of the model.

Fig. 3 shows the curvature of the computed isotropic reluctivity curves as a function of the remaining sum-of-squares error. The degree of the polynomials and the number of neurons in the perceptrons are indicated on the Figure. Obviously, the more degrees of freedom the lower the sum-of-squares error and the higher the curvature. However, for the same accuracy, perceptrons are much smoother than polynomials. The same behaviour is observed for the models of the anisotropic reluctivity curves in Fig. 4. This property makes neural network models attractive for modelling nonlinear properties.

In Figs. 5 and 6 the comparison is performed from a computational point of view. For both data sets, the real time for evaluating perceptron neural networks and polynomials is plotted against the remaining sum-of-squares error, with the number of neurons and the degree of the polynomial indicated as a parameter. These timing experiments have been done in MATLAB, on a HP B1000 workstation. Both Figures reveal an interesting feature, as for the same accuracy, perceptrons require much more computational effort than polynomials. From Figs. 7 and 8 it follows that this conclusion would not be drawn from an analysis of the number of flops. This is basically due to the facts that every neuron must carry out a time-consuming exponential function evaluation and that the code for evaluating a perceptron is much longer than the code for evaluating a polynomial.

On Figs. 5 and 6, the real time for evaluating a cubic spline and a bicubic spline is also indicated. The accuracy of a spline can be improved by increasing the number of intervals. The spline evaluation time is only weakly

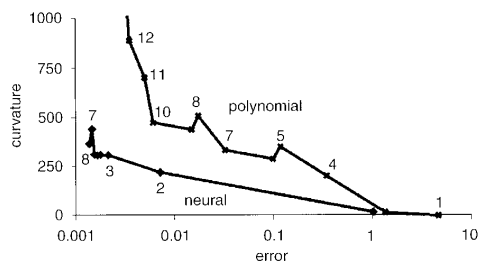


Fig. 3 Curvature of models for isotropic magnetisation curve as function of sum-of-squares error

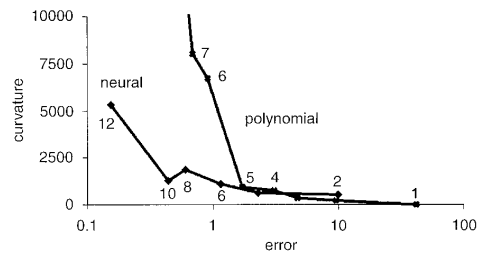


Fig. 4 Curvature of models for anisotropic magnetisation curve as function of sum-of-squares error

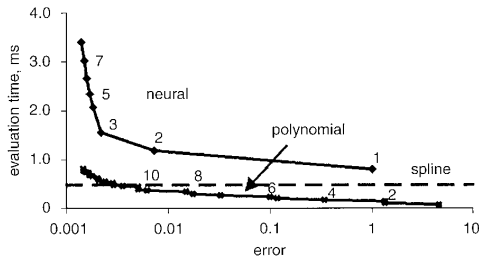


Fig. 5 Evaluation time of models for isotropic magnetisation curve as function of sum-of-squares error

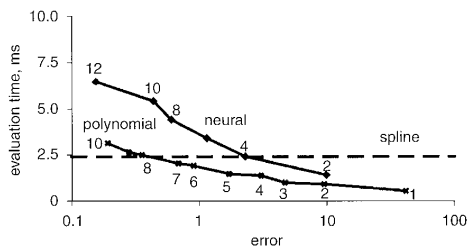


Fig. 6 Evaluation time of models for anisotropic magnetisation curve as function of sum-of-squares error

dependent on the number of nodes determining the spline, because the latter only influences the number of comparisons required for finding the evaluation interval. These comparisons involve no floating-point operations. Therefore the evaluation time is plotted as a horizontal line. For the univariate case, it is concluded from Fig. 5 that even the smallest perceptron cannot be evaluated more quickly than the cubic spline interpolant. For the bivariate case in Fig. 6, only the smallest perceptrons seem to be competitive to the bicubic spline from a computational point of view. However, their use is not encouraged, as this computational advantage only holds for the less accurate neural network models. Obviously, the same conclusion is valid for polynomial models.

Moreover, the quality of a model depends to a considerable extent on its smoothness. It may deteriorate or even preclude the convergence of the Newton-iteration scheme. By imposing constraints on the curvature of the models, as e.g. in Dierckx [5], Bishop [6], Pahner *et al.* [8] and Vande Sande *et al.* [9], it is possible to influence the smoothness. Unfortunately, the curvature decreases to the expense of a higher sum-of-squares error. Hence, in Figs. 5

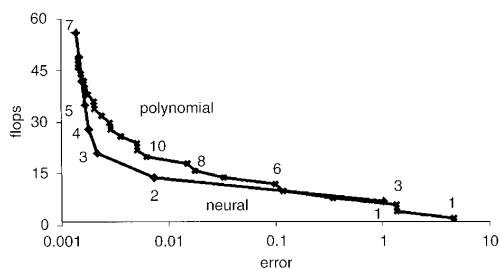


Fig. 7 Number of floating-point operations of models for isotropic magnetisation curve as function of sum-of-squares error

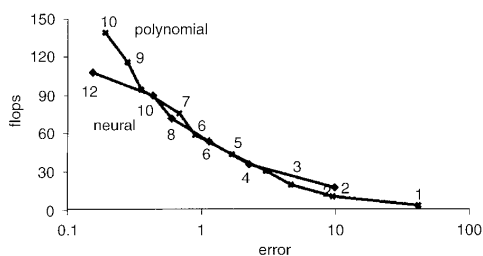


Fig. 8 Number of floating-point operations of models for anisotropic magnetisation curve as function of sum-of-squares error

and 6, this causes a shift of the polynomial and perceptron characteristics to the right. On the other hand, the cubic or bicubic spline characteristics only slightly shift upwards. This illustrates even more that polynomials and perceptron neural networks are not competitive to the commonly applied spline representations.

6 Conclusions

A comparison between polynomial, spline and perceptron neural network models for representing nonlinear isotropic or anisotropic ferromagnetic material properties has been presented. The models were compared for their accuracy,

smoothness and evaluation time, as these have a major impact on the system building and the convergence rate of the nonlinear Newton–Raphson scheme. Smooth models can be obtained by imposing constraints on the model's output. For polynomial and perceptron models, this causes a decrease in accuracy. Splines can be smoothed without accuracy loss by increasing the number of intervals. This has no significant impact on the evaluation time. Timing experiments reveal that, when taking these observations into account, polynomials and perceptron models are not competitive to the commonly used cubic splines, from the computational point of view.

7 Acknowledgments

The authors are grateful to the Belgian "Fonds voor Wetenschappelijk Onderzoek Vlaanderen" (project G.0420.99) for its financial support of this work and the Belgian Ministry of Scientific Research for granting the IUAP P5/34 on Coupled Problems in Electromagnetic Systems.

8 References

- 1 SILVESTER, P.P., and FERRARI, R.L.: 'Finite elements for electrical engineers', 3rd edn. (Cambridge University Press, Cambridge, 1996)
- 2 PATECKI, A., and STEPIEN, S.: 'Improvements of convergence BH-characteristic for the calculation of transient 3D eddy currents'. 11th International Symposium on Theoretical electrical engineering, ISTET'01, Linz, Austria, August 2001, pp. 440–444
- 3 SHIRKOOHI, G.H., and LIU, J.: 'A finite-element method for modelling of anisotropic grain-oriented steels', *IEEE Trans. Magn.*, 1994, **30**, (2), pp. 1078–1080
- 4 PRESS, W.H., TEUKOLSKY, S.A., VETTERLING, W.T., and FLANNERY, B.P.: 'Numerical recipes in C', 2nd edn. (Cambridge University Press, Cambridge, 1992)
- 5 DIERCKX, P.: 'Curve and surface fitting with splines'. (Clarendon Press, Oxford, 1995)
- 6 BISHOP, C.M.: 'Neural networks for pattern recognition', 1st edn. (Oxford University Press, New York, 1995)
- 7 GOLUB, G.H., and VAN LOAN, C.F.: 'Matrix computations', 3rd edn. (John Hopkins University Press, Baltimore, 1996)
- 8 PAHNER, U., BELMANS, R., and HAMEYER, K.: 'Optimizing ferromagnetic characteristics to reduce Newton iterations'. *COMPEL, Int. J. Comput. Math. Electr. Electron. Eng.*, 1999, **18**, (4), pp. 629–637
- 9 VANDE SANDE, H., PAHNER, U., DE GERSEM, H., HAMEYER, K., and FROYEN, L.: 'A numerical optimisation method for obtaining technically smooth magnetisation characteristics'. 11th International Symposium on Theoretical Electrical Engineering, ISTET'01, Linz, Austria, August 2001, pp. 435–439