TU Delft

PROCEEDINGS OF

EUROSIM 2001

SHAPING FUTURE WITH SIMULATION

The 4th International EUROSIM Congress, in which is incorporated the 2nd Conference on Modelling and Simulation in Biology, Medicine and Biomedical Engineering.

June 26 - 29, 2001
Aula Conference Centre
Delft, The Netherlands

Editors: A.W. Heemink
L. Dekker
H. de Swaan Arons
I. Smit
Th.L. van Stijn

ISBN: 90-806441-1-0

USING AUTOMATIC DIFFERENTIATION FOR THE MINIMAL *P*-NORM SOLUTION OF THE BIOMAGNETIC INVERSE PROBLEM

H. Martin Bücker, Roland Beucker, Christian H. Bischof

USING AUTOMATIC DIFFERENTIATION FOR THE MINIMAL *P*-NORM SOLUTION OF THE BIOMAGNETIC INVERSE PROBLEM

H. Martin Bücker[1], Roland Beucker[2], Christian H. Bischof [1]

[1]Institute for Scientific Computing, Aachen University of Technology,
D–52056 Aachen, Germany
{bischof,buecker}@sc.rwth-aachen.de
http://www.sc.rwth-aachen.de
[2]Megis Software GmbH,
Wessobrunner Str. 10, D–82166 Gräfelfing, Germany
rbeucker@besa.de
http://www.besa.de

ABSTRACT

Given the measurements of a magnetic field induced by the electrical activity of the brain, the mathematical model to localize the electrical activity on the human cortex is given by an inverse problem. The minimum-norm approach is among the common reconstruction techniques to localize the brain activity. Here, the standard approach is to minimize the Euclidean norm of the current distribution of the underlying dipole moments. A generalization from the Euclidean norm to general p-norms with $1 < p \leq 2$ is attractive because the reconstructions appear more focal as p approaches 1. Rather than using reweighted least-squares algorithms with their potential numerical instabilities, a gradient-based optimization algorithm is investigated. More precisely, a Newton-type algorithm is used where the required gradient of the cost function is either accurately computed by automatic differentiation or approximated by finite differences. Numerical results are reported illustrating that accurate gradients computed by the so-called reverse mode of automatic differentiation are more efficient than approximations based on finite differences.

KEYWORDS

Magnetoencephalography, p-Norm Minimization, Automatic Differentiation, Reverse Mode, Adifor.

1 INTRODUCTION

In magnetoencephalography (MEG) a weak magnetic field induced by the electrical activity of the brain is recorded at little distance outside the head using up to 300 so-called Super Conducting Quantum Interference Devices (SQUIDs). The mathematical model to localize the underlying electrical activity on the human cortex from the measurements is given by an inverse problem. In 1853, Helmholtz showed that there exists no unique reconstruction of a current density distribution in a volume conductor for a given electrical potential distribution on the surface of the conductor. Consequently, there are many reconstruction methods to localize the brain activity.

The minimum-norm approach (Hämäläinen and Ilmoniemi 1984) is a common method to localize electrical activity in the brain. A vector field is calculated on a pre-determined grid where each vector represents a current dipole. In general, many more source locations than sensors are used, making the

problem highly underdetermined. The current distribution with the minimal Euclidean norm is taken as the possible solution. The major disadvantage of the Euclidean norm is that the reconstructions appear very smeared. In (Wagner *et al.* 2000), however, it was observed that p-norms produce more focal reconstruction as p approaches 1.

In (Beucker and Schlitt 1996) it is shown that the solution space of the underdetermined system can be described by an overdetermined system of linear equations with full column rank. Using that formulation the minimization problem was solved by a reweighted least-squares algorithm. The major drawback of this algorithm is its numerical instability for p near 1 when a large number of dipoles is considered (Späth 1992). Therefore, in this note, the p-norm minimization problem is tackled with a gradient-based optimization method. Automatic differentiation is used for the efficient evaluation of the underlying gradient of the cost function.

The structure of this note is as follows. In Section 2, the underlying p-norm minimization problem is derived. For its efficient solution, a Newton-type algorithm needing derivatives of the cost function is employed. The gradient is not approximated by numerical differentiation such as finite differences but is computed accurately up to machine precision using automatic differentiation which is sketched in Section 3. To demonstrate the sharpening effect when using p-norms rather than the Euclidean norm, the resulting current distributions are visualized in Section 4 on a rendered cortex for different values of p with $1 < p < 2$.

2 MINIMUM *P*-NORM APPROACH

For simplicity, the human head is modeled by a spherical volume conductor K. The magnetic field $\mathbf{B}$ at a point $\mathbf{r}$ around the conductor K produced by the electrical current $\mathbf{J}$ in K is given by the Biot–Savart law

$$\mathbf{B}(\mathbf{r}) = \frac{\mu_0}{4\pi} \iiint_K \frac{\mathbf{J}(\mathbf{r}') \times (\mathbf{r} - \mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|^3} \, \mathrm{d}V', \tag{1}$$

where the integral is over all points $\mathbf{r}'$ of the volume V' of K; see (Jackson 1975) for details. The current density $\mathbf{J}$ in K is usually written as the sum of the so-called primary and volume currents (Hämäläinen *et al.* 1993). In case of a spherical volume conductor, only the tangential part of the primary current contributes to $\mathbf{B}$ around the conductor (Hämäläinen *et al.* 1993). If the primary current in K is assumed to be the sum of s current dipoles $\mathbf{Q}_i$ at locations $\mathbf{r}_{\mathbf{Q}_i}$, the expression

$$\mathbf{J}^p(\mathbf{r}'_i) = \sum_{i=1}^{s} \mathbf{Q}_i \cdot \delta(\mathbf{r}'_i - \mathbf{r}_{\mathbf{Q}_i})$$

is used for $\mathbf{J}(\mathbf{r}')$ in (1) where δ denotes the Dirac delta function. Then the problem is completely determined by the radial component of the magnetic field (Hämäläinen *et al.* 1993). Let $\mathbf{b} \in \mathbb{R}^m$ denote the vector whose entries represent the scalar components of the magnetic field in sensor direction at m locations around K and let $\mathbf{q} = (\mathbf{Q}_1^T, \mathbf{Q}_2^T, \ldots, \mathbf{Q}_s^T)^T \in \mathbb{R}^{3s}$ denote the vector whose entries are the scalar components of the ensemble of s current dipoles $\mathbf{Q}_i$ in K. Under this scenario, the discrete form of (1) is given by

$$\mathbf{b} = L\mathbf{q}, \tag{2}$$

where L is called the leadfield matrix whose dimension is $m \times 3s$. The entries of the leadfield matrix describe the effect of each dipole at the sensors. Given an ensemble of s current dipoles on a grid within a pre-determined source space in K, the resulting magnetic field at m points around K can be calculated from (2).

We now reverse this process and assume that there is a magnetic field around K whose radial components $\mathbf{b}$ are given from measurements at m sensors. We are then interested in finding an ensemble of s current dipoles on a grid producing that magnetic field. Note that, in general, the number of sensors is much less than the number of locations where current dipoles are assumed, i.e., $m \ll s$, so that the system (2) is highly underdetermined. To perform a stable minimization, the Tikhonov regularization is employed. More precisely, the following formulation presented in (Hansen 1998) is used:

$$\text{Find} \quad \mathbf{q} \in \mathbb{R}^{3s} \quad \text{to minimize} \quad f(\mathbf{q}) \tag{3}$$

where

$$f(\mathbf{q}) := \left\| \begin{pmatrix} L \\ \lambda I \end{pmatrix} \mathbf{q} - \begin{pmatrix} \mathbf{b} \\ \mathbf{0}_{3s} \end{pmatrix} \right\|_p \tag{4}$$

in which $\lambda \in \mathbb{R}$ is a scalar parameter, $1 < p < 2$, the symbol I is the $3s \times 3s$ identity matrix, and $\mathbf{0}_{3s}$ denotes the zero vector of length $3s$. For the solution of the p-norm minimization problem (3), a Newton-type method is used requiring the gradient of the cost function f with respect to $\mathbf{q}$ denoted by $\nabla f \in \mathbb{R}^{3s}$ hereafter.

3 AUTOMATIC DIFFERENTIATION

Derivatives are not only useful in optimization algorithms but also play an important role in a variety of scientific computing applications including solution of nonlinear equations, sensitivity analysis, parameter identification, and inverse problems. Automatic differentiation (AD) (Griewank 2000) is a powerful technique for accurately and efficiently evaluating derivatives of a function f given in the form of a program in a high-level programming language, here Fortran. The program is treated as a potentially very long sequence of a small set of elementary operations such as $+, -, *, /, \sin, \cos$, or $\exp$ for which derivatives are known. AD tools systematically apply the chain rule of differential calculus over and over again to these elementary derivatives eventually accumulating the derivatives of f.

Compared with other techniques to evaluate derivatives, AD offers the following significant advantages. Numerical differentiation by, say, finite differences only gives an approximation to derivatives whereas AD-generated derivatives are accurate up to machine precision. Moreover, the performance of AD-generated code usually exceeds that of the corresponding finite difference code, yet often rivaling the performance of code developed by hand. AD also requires less human effort because it eliminates the time for code development and debugging. Furthermore, AD can be used to validate simulation results based on numerical differentiation. The main disadvantage of finite differences is that they crucially depend on a suitable step size typically making indispensable a number of experiments varying the step size. There is always the inherent dilemma of finite differences that, on the one hand, the step size should be small in order to decrease the truncation error and that, on the other hand, the step size should be large to avoid cancellation errors in finite-precision arithmetic.

In the context of automatic differentiation, a program input variable is called *independent* if derivatives with respect to that variable are to be computed. A program output variable is called *dependent* if its derivative with respect to the independent variables are desired. There are two basic implementations of automatic differentiation differing in operations counts and storage requirement. The so-called forward mode of AD is usually preferred when the number of independent variables is less than the number of dependent variables because its cost of floating point operations and storage—relative to the corresponding cost of evaluating the function being differentiated—scales with the number of independent variables and does not depend on the number of dependent variables. In contrast, the so-called reverse mode of AD is more efficient if the number of dependent variables is less than the number of independent variables because its relative cost of floating point scales with the number of dependent variables and does *not* depend on the number of independent variables. However, the reverse mode may exhibit unpredictable

Table 1. Ratio $T_{F'}/T_F$ of the execution times of F' and F

Forward mode	Reverse Mode	Finite differences
2974	233	1111

storage requirement because many intermediate values need to be stored and/or recomputed. A detailed discussion of the forward and reverse mode is given in (Griewank 2000).

In the present work, we use version 3.0 of the Adifor (Bischof *et al.* 1996) tool supporting both forward and reverse mode of automatic differentiation. More precisely, suppose that there is a Fortran code F evaluating the cost function f given by (4) at a particular input $\mathbf{q}$, i.e., invoking F yields

$$y = f(\mathbf{q}).$$

Then, Adifor automatically generates a Fortran code F' by specifying $\mathbf{q}$ as the independent variable and y as the dependent variable. The code F' still evaluates f at $\mathbf{q} = (q_1, q_2, \ldots, q_{3s})^T$ but also evaluates ∇f at the same point, i.e., invoking F' computes the pair

$$y = f(\mathbf{q}) \qquad \text{and} \qquad \widehat{y} = \nabla f|_{\mathbf{q}}$$

where $\widehat{y}$ is an object storing the elements of the gradient $\partial f/\partial q_i$ for $i = 1, 2, \ldots, 3s$.

4 NUMERICAL EXPERIMENTS

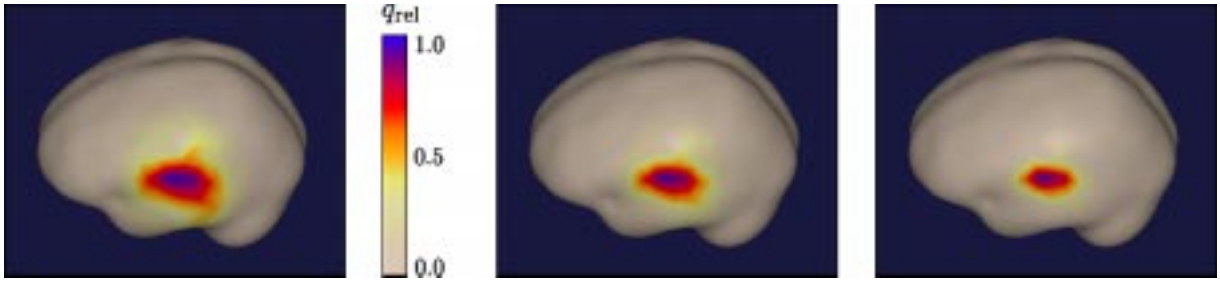
In the numerical experiments reported below, the measurements of a magnetic field at m sensors are simulated by the computation of $\mathbf{b}$ in (2) where it is assumed that a current dipole $\mathbf{Q}$ is put in the left hemisphere at location $\mathbf{r_Q} = (x, y, z)^T = (-0.6, 0.0, 0.2)^T$ of the unit sphere K with moment $\mathbf{Q} = (0, 1, 0)^T$; that is, with unit moment in y-direction. The corresponding leadfield matrix L is calculated using the BESA 2000 ® software package (Megis Software GmbH) for a BTI 148-magnetometer array. In the experiments, there are $s = 366$ grid points and $m = 148$ sensors. The scalar parameter λ in (4) is set to 0.01.

In Table 1, the ratios of the total execution times of F' and F are given for various approaches calculating the gradient. Recall that F is the code to evaluate the cost function $f(\mathbf{q})$ whereas F' is the program evaluating the pair $f(\mathbf{q})$ and $\nabla f|_{\mathbf{q}}$. Thus the ratio $T_{F'}/T_F$ of the execution times of the two programs F' and F is a measure of the cost for computing the gradient that is of length $3s = 1098$ here. On an SGI Onyx equipped with 195 MHz MIPS R10000 CPUs, the time spent in evaluating the cost function is $T_F = 2.7 \cdot 10^{-3}$s. The ratios $T_{F'}/T_F$ when the gradient in F' is calculated using the forward and reverse mode of automatic differentiation are given in Table 1. The table also presents the corresponding ratio when using finite difference approximations

$$\frac{\partial f}{\partial q_i} \approx \frac{f(\mathbf{q} + h_i \mathbf{e}_i) - f(\mathbf{q})}{h_i} \qquad \text{for } i = 1, 2, \ldots, 3s \tag{5}$$

where $\mathbf{e}_i \in \mathbb{R}^{3s}$ is the ith Cartesian unit vector. Automatic differentiation not only eliminates the need for experimenting with different step sizes h_i but is by far more efficient than finite difference approximations when the reverse mode is employed. These experimental results agree with the theory stating that $T_{F'}/T_F$ scales with the number of independent variables, $3s = 1098$, in the forward mode and the fact that $3s + 1 = 1099$ function evaluation are necessary using finite difference approximations (5). In contrast, using the reverse mode the ratio $T_{F'}/T_F$ does not depend on $3s$ at all.

For the solution of the p-norm minimization problem (3), the routine E04KZF of the NAG Fortran Library (The Numerical Algorithms Group Limited 1999) implementing a Newton-type algorithm is



$p = 2.0$ $p = 1.75$ $p = 1.5$

Figure 1. Current dipole distribution for different values of p

used allowing the user to supply a routine to calculate the pair $f(\mathbf{q})$ and $\nabla f|_{\mathbf{q}}$. In Figure 1, the resulting current dipole distribution $\mathbf{q} = (\mathbf{Q}_1^T, \mathbf{Q}_2^T, \ldots, \mathbf{Q}_s^T)^T$ is visualized using the BESA 2000® software package (Megis Software GmbH). Here, different p-norms are involved in the minimization of the cost function (4). More precisely, the quantity at grid point i identified by color is

$$q_{\text{rel}} := \frac{|\mathbf{Q}_i|}{\max\limits_{1 \le j \le s} |\mathbf{Q}_j|} \qquad \text{for } i = 1, 2, \ldots, s.$$

Comparing the distributions for various values of p starting with the Euclidean norm, $p = 2.0$, we see that the reconstructions appear more focal as p approaches 1.5.

5 CONCLUDING REMARKS

Given a magnetic field around the human head induced by the electrical activity of the brain, the mathematical model to localize the electrical activity on the human cortex is given by an inverse problem. Among the various reconstruction techniques to localize the brain activity, the minimum p-norm approach minimizes a suitable cost function with respect to the current distribution of the underlying dipole moments. The main advantage of this approach compared to standard approach involving the Euclidean norm is the availability of the parameter p that can be used to let the reconstructions appear more focal as p approaches 1.

From a numerical point of view, a Newton-type algorithm is used to carry out the actual optimization of the cost function. Here, automatic differentiation not only enables the evaluation of the required gradient accurately up to machine precision. It also increases performance considerably in comparison to an approximation of the gradient by finite differences. To be more precise, the reverse mode of automatic differentiation is used to compute a gradient of a scalar-valued cost function with respect to a large number of independent variables, here 1098. However, the number of floating point operations for evaluating the cost function and the gradient in a combined way does *not* depend on the number of independent variables using the reverse mode.

The formulation of the minimization problem used in this study is based on a regularization involving a scalar parameter λ. Directions for future work includes the investigation of the influence of λ on the simulation results.

ACKNOWLEDGEMENTS

The authors are grateful to Astrid Kallmeier who is responsible for the composition of the graphics.

REFERENCES

Beucker, R. and H. A. Schlitt (1996). On minimal l_p-norm solutions of the biomagnetic inverse problem. Technical Report KFA–ZAM–IB–9614, Research Centre Jülich, Jülich, Germany.

Bischof, C., A. Carle, P. Khademi, and A. Mauer (1996). ADIFOR 2.0: Automatic differentiation of Fortran 77 programs. *IEEE Computational Science & Engineering 3*(3), 18–32.

Griewank, A. (2000). *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. Philadelphia: SIAM.

Hämäläinen, M., R. Hari, R. J. Ilmoniemi, J. Knuutila, and O. V. Lounasmaa (1993). Magnetoencephalography — theory, instrumentation, and applications to noninvasive studies of the working brain. *Rev. Mod. Phys.* **65**, 413–497.

Hämäläinen, M. S. and R. J. Ilmoniemi (1984). Interpreting measured magnetic fields of the brain: Estimates of current distributions. Technical Report TKK–F–A559, Helsinki University of Technology, Department of Technical Physics, Helsinki, Finland.

Hansen, H. C. (1998). *Rank-Deficient and Discrete Ill-Posed Problems*. Philadelphia: SIAM.

Jackson, J. D. (1975). *Classical Electrodynamics* (second ed.). New-York: Wiley.

Späth, H. (1992). *Mathematical Algorithms for Linear Regression*. San Diego: Academic Press.

The Numerical Algorithms Group Limited (1999). *NAG Fortran Library Manual, Mark 19*. Oxford: The Numerical Algorithms Group Limited.

Wagner, M., H.-A. Wischmann, M. Fuchs, T. Köhler, and R. Drenckhahn (2000). Current density reconstructions using the L1 norm. In C. J. Aine, Y. Okada, G. Stroink, S. J. Swithenby, and C. C. Wood (Eds.), *Biomag 96: Proceedings of the Tenth International Conference on Biomagnetism*, Volume 1, pp. 393–396. New York: Springer.