ASEG-PESA 2013
'THE EUREKA MOMENT'
11-14 AUGUST 2013 • MELBOURNE, AUSTRALIA

Large Scale Joint Inversion of Geophysical Data using the Finite Element Method in *escript*

Lutz Gross
The University of Queensland, School of Earth Sciences
St Lucia, QLD, Australia
l.gross@uq.edu.au

Carina Kemp
Geoscience Australia
Symonston ACT, Australia.
Carina.Kemp@ga.gov.au

SUMMARY

The program package *escript* is a module in *python* for solving mathematical modelling problems. It is based on the finite element method (FEM) and scales on compute clusters for thousands of cores. In this paper we will discuss an extension to *escript* for solving large-scale inversion problems, in particular the joint inversion of magnetic and gravity data. In contrast to conventional inversion programs *escript* avoids the assemblage of the -in general- dense sensitivity matrix which is problematic when it comes to large-scale problems. Moreover, we will show how the FEM approach can easily be used to solve the adjoined forward problems required for the gradient calculation of the cost function. We will demonstrate the application of the algorithm for field data using hundreds of cores.

Key words: geophysical inversion, joint inversion, finite elements, parallel computing.

INTRODUCTION

The problem of the inversion of geophysical data is finding the minimum of a suitable cost function subject to constraints defined by partial differential equations (PDEs). The cost function measures the data misfit and the variation of the physical property across the domain. In general, this problem is equivalent to a system of coupled PDEs for three unknowns – the physical properties (e.g. density and susceptibility), the observation (e.g. gravity and magnetic field) and the Lagrangean multipliers introducing the PDE constraints. Therefore it is appropriate to use well established PDE solution methods such as the finite element method (FEM) (Zienkiewicz et al. 2005) to solve inversion problems.

It is common practice to state the inversion problem as a quadratic programming problem using Green's functions to explicitly calculate the sensitivity matrix of the observations from the physical property, for instance see (Li and Oldenburg, 1996). A particular limitation of the usage of the Green's functions approach is the fact that its application is limited to linear inversion problems and cannot be applied to more advanced forward models as for instance required for large values of susceptibility, see Lelivre and Oldenburg 2006. In this paper we discuss how the FEM can be applied to solving inversion problems using the adjoint-state method, see Plessix 2006. We will in particular discuss the application case of joint inversion of gravitational and magnetic data for which the inversion problem becomes non-linear. The key idea is to formulate the solution process in terms of PDEs, PDEs solution and an appropriate inner product rather using a linear algebra formulation. A second key component is the usage of the FEM to solve the PDEs. We will show that the FEM method works with the adjoint-state method in a natural way. Moreover, FEM can easily parallelized using the domain decomposition approach distribution cells across different compute nodes. In contrast to the commonly used approach to split the work across compute nodes through data tiling domain decomposition is applied during the PDEs solution process which leads to a more efficient inversion process.

PROBLEM

The geophysical properties within a domain are described by the level set function $m=(m_0,m_1)=(\rho,k)$ where ρ and k described the gravitational and magnetic properties, respectively. The task is to find the level set function m for which the cost function

$$J(m)=J^{reg}(m)+J^{grav}(\rho)+J^{mag}(k)+J^{cross}(m_0,m_1)$$

takes the minimum. In this cost function we use the H^1-regularisation:

$$J^{reg}(m)=\int \|\nabla m_0\|^2+\|\nabla m_1\|^2\ dx$$

where the integral is calculated over the region of interest and $\|.\|$ denotes the Euclidean norm. For the sake of simpler presentation L^2-regularisation and weighting factors are dropped but can be considered in the approach discussed in this paper. To align the contours for density and susceptibility we use the cross-gradient gradient term

$$J^{cross}(m_0,m_1)=\int \|\nabla m_0\|^2\|\nabla m_1\|^2-(\nabla m_0\cdot\nabla m_1)^2\ dx$$

see Gallardo et al.(20005). The miss-fit functions for the gravity $\hat{g}$ and magnetic data $\hat{B}$ are given as

$$J^{grav}(g)=\int \left(w^g\cdot(g-\hat{g})\right)^2 dx$$

$$J^{grav}(B)=\int \left(w^B\cdot\left(B-\hat{B}\right)\right)^2 dx$$

with spatially variable weighting factors w^g and w^B allowing for the location, error and direction of measurements. The gravity field g and magnetic field B are given as the negative gradient of the gravity and magnetic scalar potentials

ϕ and ψ , respectively. Both potentials are given as the solution of a PDE with appropriate boundary conditions. When using the FEM the PDEs are solved as variational equations. For the gravity potential ϕ this takes the form

$$\int \nabla v \cdot \nabla \phi \quad dx = \int v \rho \ dx$$

which needs to be fulfilled for all smooth test functions v with appropriate boundary conditions. Similarly with the background magnetic field B^b the magnetic potential ψ is given as the solution of

$$\int \nabla v \cdot \nabla \psi \ dx = \int k \nabla v \cdot B^b \ dx$$

for all smooth test functions v .
The domain of the inversion covers the subsurface where density and susceptibility are calculated as well as the region above the surface where measurements have been taken. In practice it is assumed that the region is bounded so it needs to be chosen sufficiently large to avoid boundary effects on the inversion results for the region of interest.

SOLUTION METHOD

We use the quasi-Newton scheme in form of the the limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) method, see Nocedal (1980), to solve the minimisation problem for m . The method requires the gradient ∇J of the cost function and an appropriate inner product $\langle .,. \rangle$ such that for a level set increment δm

$$J(m + \alpha \cdot \delta m) = J(m) + \alpha \cdot \langle \delta m, \ \nabla J(m) \rangle + o(\alpha)$$

for scalar $\alpha \to 0$. For the cost function J the directional derivative is given in the form

$$\langle \delta m, \ \nabla J \rangle = \int Y_0 \delta m_0 + Y_1 \delta m_1 + X_0 \cdot \nabla \delta m_0 + X_1 \cdot \nabla \delta m_1 \ dx$$

with suitable coefficients Y_0 , Y_1 , X_0 and X_1 which are functions of the level set function and of their locations within the region of interest. From the regularisation and cross gradient term one can easily show that

$$X_0 = \left(1 + \|\nabla m_1\|^2\right)\nabla m_0 - (\nabla m_1 \cdot \nabla m_0)\nabla m_1$$

$$X_1 = \left(1 + \|\nabla m_0\|^2\right)\nabla m_1 - (\nabla m_1 \cdot \nabla m_0)\nabla m_0$$

The gradient of gravity data misfit ∇J^{grav} and magnetic data misfit function ∇J^{mag} define the coefficient Y_0 and Y_1 , respectively. We use the adjoint-state method to calculate these gradients, see Plessix (2006): With $\hat{Y}_0 = \left(w^g \cdot (g - \hat{g})\right) w^g$ the directional derivative of the gravity component is given as

$$\left\langle \delta m_0, \nabla J^{grav} \right\rangle = \int \hat{Y}_0 \cdot \nabla \delta \phi \quad dx$$

where $\delta\phi$ is given as the solution of the variational equation

$$\int \nabla v \cdot \nabla \delta \phi \quad dx = \int v \ \delta m_0 \ dx$$

for all smooth test function *v*. In order to get an explicit representation of ∇J^{grav} increment $\delta\phi$ for the gradient potential needs to be translated into a increment δm_0 for the level set function. To do this we solve the adjoint problem to calculate the function Y_0 from $\hat{Y}_0$:

$$\int \nabla v \cdot \nabla Y_0 \ dx = \int \hat{Y}_0 \cdot \nabla v \ dx$$

for all smooth test function *v*. From this equation and the definition of the scalar potential increment $\delta\phi$ we get

$$\int Y_0 \ \delta m_0 \ dx = \int \nabla Y_0 \cdot \nabla \delta \phi \quad dx = \int \hat{Y}_0 \cdot \nabla \delta \phi \quad dx$$

and finally

$$\left\langle \delta m_0, \nabla J^{grav} \right\rangle = \int Y_0 \ \delta m_0 \ dx \ .$$

Similarly, one gets for the magnetic component

$$\left\langle \delta m_1, \nabla J^{mag} \right\rangle = \int \delta m_1 Y_1 \ dx$$

where $Y_1 = \nabla Y_1 \cdot B^b$, $\hat{Y}_1 = \left(w^B \cdot (B - \hat{B})\right) w^B$ and

$$\int \nabla v \cdot \nabla Y_1 \ dx = \int \hat{Y}_1 \cdot \nabla v \ dx$$

for all smooth test function *v*.

We also need to provide an approximation of the Hessian operator of the cost function:

$$\nabla J(m + \delta m) - \nabla J(m) \approx H \ \delta m$$

For the L-BFGS one needs to provide the level set increment δm for the difference of two cost function gradients as the solution of the equation above. Again the equation is solved in variational form:

$$\langle v, H \ \delta m \rangle = \int Y_0 v_0 + Y_1 v_1 + X_0 \cdot \nabla v_0 + X_1 \cdot \nabla v_1 \ dx$$

for all smooth test function *v* where coefficients Y_0 , Y_1 , X_0 and X_1 represent the difference of two cost function gradients. In theory the Hessian operator can be determined easily by calculating the partial derivatives of the coefficients with respect to the level set function m . For coefficient Y_0 and Y_1 this is not a straightforward procedure as the adjoint problems come into the way. However, as it is sufficient to

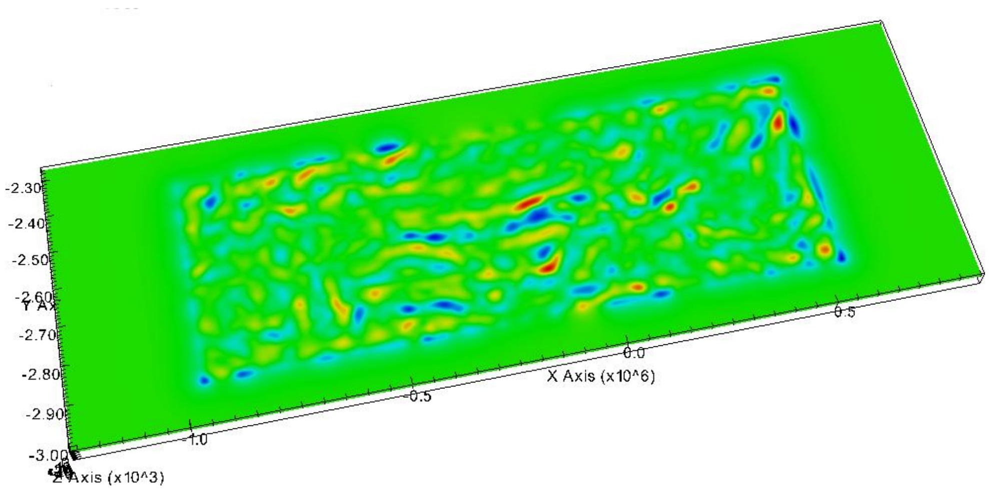

Figure 2: Result from gravity inversion of the Willowra data set. Colour shows the density distribution at horizontal plain just below ground level.

show the distribution of the susceptibility. Colour shows the density distribution. Both distributions are shown at horizontal plain just below ground level.

provide an approximation of level set increment δm we ignore contributions from the data misfit to the cost function. For this assumption we get

$$\left\langle v, H\ \delta m\right\rangle = \int A_{piqj} \frac{\partial v_p}{\partial x_i} \frac{\partial \delta m_q}{\partial x_j} dx$$

where summation over the indices *p*, *i*, *q* and *j* is performed and A_{piqj} is the partial derivative of X_{pi} with respect to $\frac{\partial m_q}{\partial x_j}$. In most practical applications weighting of cost function components will be chosen to emphasise the data misfit. For these cases dropping contributions from the data misfit in the Hessian operator will lead to an inferior search direction for L-BFGS and as a consequence to a larger number of iteration steps. In principle it is possible to include contributions from data misfit. This however would require the solution of several coupled differential equations. At this point we assume that the additional solution costs for this would exceed compute time reductions through a decrease of the number of L-BFGS iterations.

IMPLEMENTATION

We use the *escript* modelling environment (Gross at al. 2007) in *python* (van Rossum et al 2001) to implement the L-BFGS based joint inversion. *escript* is built to solve complex mathematical models. Its core is a *python* class to define a template for linear, second order PDEs which in variational form for a single PDE and a scalar solution *u* is given as

$$\int A_{ij} \frac{\partial v}{\partial x_i} \frac{\partial u}{\partial x_j} + B_i \frac{\partial v}{\partial x_i} u + C_i v \frac{\partial u}{\partial x_i} + Dvu \quad dx = \int X_i \frac{\partial v}{\partial x_i} + Yv \quad dx$$

for all smooth test function *v*. Summation over the indices *i* and *j* is performed. The functions A_{ij} , B_i, C_i, *D*, X_i and *Y* are the PDE coefficients. For the general form we refer to Gross *et.al* (2010).

Users write *python* scripts to define the coefficients as expressions of parameters, input data and solutions of other PDEs. The latter allows for the solution of time-dependent, non-linear problems and for coupling various models in a single simulation script. At run time the coefficients are evaluated and passed on to the FEM solver returning an (approximate) solution of the PDE. It is particular strength of *escript* that the user does not have to be aware of how values for the coefficients and solution are stored. Data structures and – on parallel computers – distribution of data and meshes (or grids) are not visible to the user on the *python* level. This allows developing complex models without programming skills and running them on parallel computers without time-consuming program modifications.

The L-BFGS based inversion process as outlined above can easily implemented using *escript*. For each evaluation of the cost function *J* the gravity and magnetic scalar potentials ϕ and ψ are calculated using the *escript* PDE class. For the calculation of the cost function gradient the corresponding adjoint problems are solved in variational form using *escript*. As shown above the (approximate) evaluation of the inverse of the Hessian operator can be expressed as the solution of PDE in variational form which again can be easily implemented using *escript*'s PDE solver. Notice that for this a system of tightly coupled PDE needs to be solved.

When running *escript* across different compute nodes in a compute cluster (Bischof 2008) *escript* distributes the grid cells (or mesh in case of more complex geometries) across the available compute nodes to achieve an equal distribution of work load and to minimise the costs for data exchange between compute nodes as required during the PDE solves. The PDE solution as well as the update of the physical properties is performed in parallel across all cells.

CASE STUDY

In this section we show some preliminary results from a joint inversion of gravity and magnetic data from the Willowra region in central Australia. Data are given at a resolution of about 800m covering a region of about 540km by 1500km. Vertically the domain extends to a depth of 40km below and height of 6km above ground level. Figure 1 shows the results for a joint inversion of the data using about 27 Million cells. The inversion used a total 384 cores on a SGI Altix ICE 8200 EX compute cluster for about 1.5 days. The work was distributed across 48 compute nodes while on each compute additional parallelization across local cores has been used via threading, see Gross et.al (2010). At this point we are still in the process to identify appropriate weighting factors for the various components of the cost function in particular for the cross gradient term controlling the alignment of density and susceptibility contours. Figure 2 shows the results form the inversion of the gravity data only. For this inversion 54 Million cells have been used. The visualisation shows the additional padding area added to the region covered by the data in order to reduce the impact of boundary effects onto the interior region.

CONCLUSIONS

The FEM approach and its implementation in *escript* based on the adjoint-state method can easily be extended to other inversion problems such as seismic and MT data. When it comes to solve large scale problems the Hessian operator which inverse plays the role of a preconditioner is of critical importance to make inversions feasible even when more compute power is used. Our experiments with very large data set indicate that dropping contribution from the data misfit in the Hessian operator approximation can cause problems for convergence in particular if a good inversion result requires weighting factors for the data misfit. However at this point it is not clear which in the general is the most efficient approach Hessian operator approximation.

ACKNOWLEDGMENTS

The authors thank Cihan Altinay and Azadeh Salehi Gharibi at The University of Queensland, School of Earth Sciences, for their contributions to the software development and data processing required to prepare the paper. The project work has been funded by the AuScope National Collaborative Research Infrastructure Strategy (NCRIS) and the Australian Geophysical Observatory System (AGOS) under the Education Investment Fund (EIF) of the Australian Commonwealth.

REFERENCES

C. Bischof, 2008, Parallel Computing: Architectures, Algorithms, and Applications, IOS Press.

L.A Gallardo, M.A. Meju, and M.A. Prez-Flores, 2005, A quadratic programming approach for joint image reconstruction: mathematical and geophysical examples. Inverse Problems, 21(2):435.

L. Gross, A. Cihan, A. Amirbekyan, F. Joel Fenwick, L.M. Olsen-Kettle, K . Steube, L. Graham., H.B. Muhlhaus, 2010, esys-Escript User’s Guide: Solving Partial Differential Equations with Escript and Finley. Release 3.2. The University of Queensland, Australia, available at http://espace.library.uq.edu.au/view/UQ:253350.

L. Gross, L. Bourgouin, A.J. Hale, H.B. Muhlhaus, 2007, Interface modeling in incompressible media using level sets in escript. Physics of the Earth and Planetary Interiors 163, 23–34.

L. Gross, J. Fenwick, L. Gao and D. Green, 2010, A comparison of clusters with Intel "Nehalem" and "Harpertown" processors using an FEM Benchmark, 4th eResearch Australasia Conference, Gold Coast, Qld., Australia, 2010.

R.E. Plessix, 2006, A review of the adjoint-state method for computing the gradient of a functional with geophysical applications, Geophys. J. Int., 167(2), 495–503.

P.G. Lelivre and D.W. Oldenburg, 2006, Magnetic forward modelling and inversion for high susceptibility , Geophys. J. Int. 166, 76–90 .

Y. Li and D.W. Oldenburg, 1996, 3-D inversion of magnetic data, Geophysics, 61, 394–408.

J. Nocedal, 1980, Updating quasi-Newton matrices with limited storage. Math. Comp., 35(151), 773–782.

G. van Rossum and F.L. Drake (eds), 2001, Python Reference Manual, PythonLabs, Virginia, USA. Available at http://www.python.org.

O.C. Zienkiewicz, and R.L Taylor and J.Z Zhu, 2005, The Finite Element Method: Its Basis and Fundamentals, 6th edn. Elsevier, Oxford.