6th International Symposium on Three-Dimensional Electromagnetics
Berkeley, California, March 28–30, 2017

jInv: A modular and scalable framework for electromagnetic inverse problems

Eldad Haber[1], and Patrick Belliveau
[1]University of British Columbia

Summary

Our group has developed the software package jInv, which offers a modular and extensible framework for solving pde constrained parameter estimation problems in the Julia programming language. We describe the core functionality of the package and its current electromagnetic modelling and inversion capabilities. The package offers optimization algorithms, forward modelling routines for common problems, tools for discretizing partial differential equations and interfaces to top linear algebra libraries for solving linear systems. It is also highly extensible, offering users the ability to use the discretization tools to quickly write their own forward modelling routines and connect these to the jInv optimization routines to perform inversions. We show examples of the use of jInv in DC resistivity seismic traveltime tomography joint inversion and dispersive electromagnetic forward modelling.
Keywords: 4-6 keywords, forward modeling, inversion, software, finite-volume

Introduction

Software development is a key part of geophysical electromagnetics (EM) research. Most real world problems cannot be solved analytically and computational tools are required in order to produce useful solutions. Three-dimensional (3D) EM forward modelling and inversion is particularly computationally intensive. Consequently, software developed for handling real-world 3D EM problems has been highly specialized and written in low level programming languages like C and Fortran. Examples of such work include that of Commer & Newman (2004) and Cox et al. (2010).

Historically, there has been less attention paid to researcher productivity and software development best practices in the world of EM geophysics, and in scientific computing more generally (Wilson et al., 2014). There have been two significant recent efforts that we know of in the applied geophysics community to rigorously use software development best practices to create modular, extensible general software packages for geophysical modelling and inversion. These are the SimPEG (Cockett et al., 2015) and GIMLi (Rücker et al., 2017) packages. SimPEG was written in Python, a popular dynamic, high level, easy to use programming language and GIMLi was written in a combination of Python and C++. They both provide ready made software for solving common geophysical forward modelling and inverse problems. More importantly, however, they provide libraries of easy to use building block routines that allow users to quickly write their own software to solve new problems.
These packages represent significant advances but they were not designed with high performance computing in mind. Our new software package jInv provides provides a general, modular and extensible framework for solving geophysical inverse problems. It provides similar functionality to SimPEG and GIMLi but is designed to scale to large distributed memory parallel computing systems. jInv is written in the Julia programming language (Bezanson et al., 2014). Julia is a relatively new dynamic language focused on high performance technical computing. It offers MATLAB like syntax while providing sophisticated built in support for distributed memory parallel computing.

jInv is comprised of a core set of optimization routines integrated with a system for evaluating forward problems in parallel, flexible misfit and regularization routines, tools for discretizing partial differential equations (PDEs), and interfaces to top software packages for solving sparse systems of linear equations. The package is designed to be modular and extensible. It provides a simple interface that allows users to easily add their own forward problems, optimization algorithms, and regularization functions. The core of the framework is discussed in some detail in a paper currently in preparation Ruthotto et al. (2016). The package is freely available online and released under the permissive MIT license. You can find it at `https://github.com/juliainv`.

The remainder of this abstract will give a brief overview of the core functionality of jInv and discuss its application to electromagnetic problems and electromagnetic-seismic joint inversion.

The jInv framework

At the most general level, jInv was designed to solve regularized least squares optimization problems, the general category under which most geophysical EM inverse problems fall. We formulate the inverse problem as an optimization problem. We seek the vector of model parameters $\mathbf{m}$ that minimizes a given objective function Φ. jInv supports objective functions that may be expressed as a sum of data misfit terms added to a regularization term. We write this

$$\min_{\mathbf{m}} \Phi(\mathbf{m}) = \frac{1}{2}\sum_j \Phi_d\left[\mathbf{d}_j^p(\mathbf{m}), \mathbf{d}_j^o\right] + \alpha R(\mathbf{m}) \quad (1)$$
$$\text{subject to} \quad \mathbf{m}_{\mathrm{L}} \le \mathbf{m} \le \mathbf{m}_{\mathrm{H}}.$$

Here Φ_d is a misfit function, normally a weighted sum of squares, that quantifies the discrepancy between observed data $\mathbf{d}^o$ and synthetic data $\mathbf{d}^p$. $R(\mathbf{m})$ is a regularization function and α is a tradeoff parameter that balances the importance of the misfit and regularization terms in the inversion. The vectors $\mathbf{m}_L$ and $\mathbf{m}_H$ are lower and upper bounds on the model parameters. Computation of the synthetic data for a given model usually involves many independent computations. For example, in a frequency domain airborne electromagnetic survey, a great many source locations are employed and data is collected at several frequencies for each source location. In jInv, each term of the sum in equation (1) is considered an independent forward problem that can be solved concurrently with others given sufficient computing resources. A master process then sums the contributions from each problem. In the frequency domain EM example, one might designate each frequency as a separate forward problem.
The user may specify a custom function for each forward problem that maps the model parameters used in optimization to the parameters of the pde governing that forward problem. This allows for simple implementation of joint inversion schemes. The forward problems corresponding to each modality can simply be regarded as individual, independent terms in the summation of equation (1). An example of DC resistivity and seismic traveltime tomography will be discussed later in this abstract.

Parallel performance

We examined the efficiency of our implementation of this parallelism across forward problems using a weak scaling test of the DC resistivity problem. The run time for a DC forward modelling using 10 sources executing on a single worker process was computed. Batches of 10 sources were then added on their own nodes of a cloud computing engine, up to a total of 49 batches or 490 sources with one batch per worker process and one master process coordinating the work. Parallel efficiency was then computed as a function of the number of workers. Parallel efficiency is the ratio of the time required to forward model one batch of sources divided by the time required to forward model n batches. The results of the scaling test are shown in figure 1.

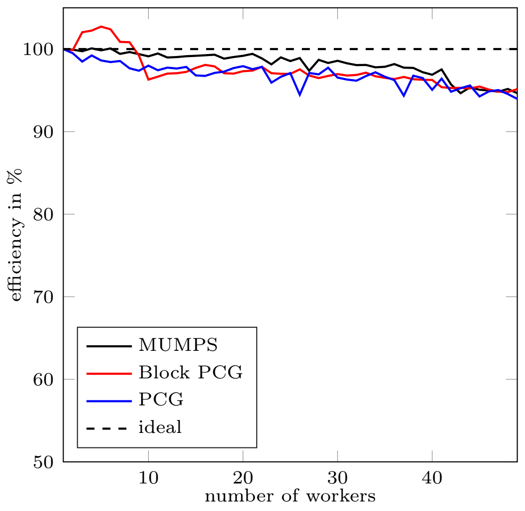

Figure 1: Parallel efficiency of DC forward modelling on a cloud computing engine.

The test was performed using three different linear solvers to solve the system of equations arising from discretization of the problem. We used the MUMPS sparse direct matrix factorization package Amestoy et al. (2001), as well as standard and block preconditioned conjugate gradient solvers implemented purely in Julia. The results were encouraging for all three solvers, with little overhead and efficiency above 90% for 49 workers.

Example applications

Joint inversion

As discussed above, jInv allows for the combination of arbitrary forward problems. In order to implement a multi-physics joint inversion, all that is needed are forward solvers for each modality and functions that relate the model parameters used in inversion to the physical parameters of each modality. We demonstrate this functionality by a DC resistivity and seismic traveltime tomography joint inversion for seismic velocity using the well known SEG/EAGE salt modelAminzadeh et al. (1997). Seismic velocity and earth conductivity are coupled by the following equation

$$\sigma(m) = \left(2 - \frac{m}{c}\right)\left(\frac{b-a}{2}\cdot(\tanh(10\cdot(c-m))+1)+a\right),$$

where σ is conductivity and m is the seismic velocity, which is used as the model parameter in the optimization procedure. The scalars a, b, and c are tuning parameters chosen to yield a conductivity contrast between the salt body (the main anomaly) and the background. The true velocity model is shown in figure 2(a).

The remaining parts of figure 2 show the DC only, traveltime tomography only and joint inversions. The DC inversion recovers shape and position of the salt body well but does a poor job of estimating the velocity. The traveltime inversion gave an excellent recovery of the velocity of the salt and the top of the body but could not recover the depth extent well. The traveltime inversion was able to combine the advantages of each approach and produce the best recovery. In jInv, moving from these two separate inversions was as simple as programming the velocity to conductivity mapping and adding the two forward problems to a list of forward problems to be evaluated during inversion.

Dispersive electromagnetic modelling

As briefly mentioned above, we have implemented frequency and time domain electromagnetics modules in the jInv framework. These modules consist of forward modelling and sensitivity calculation routines. The sensitivity routines compute the Frechet derivatives required for gradient based optimization algorithms. These routines are all mesh independent. Currently finite volume discretization on rectilinear tensor product meshes and on locally refined rectangular meshes called OcTrees, as well as finite element discretization on tetrahedral unstructured meshes. Users may make use of these discretization routines to quickly build their own mesh independent forward modelling codes.

In addition to these more standard modelling tools we are also developing modelling techniques for dispersive time-domain electromagnetic modelling for modelling time-domain induced polarization in the presence of significant EM coupling. Our modelling technique is based on the notion of stretched exponential relaxation discussed by Haber (2015). We formulate Maxwell's equations as follows

$$\begin{aligned}
&\frac{\partial \mathbf{b}}{\partial t} = -\nabla \times \mathbf{e} \\
&\nabla \times \mu^{-1}\mathbf{b} - \mathbf{j} = \mathbf{s} \\
&(1-\eta)\frac{\partial \mathbf{j}}{\partial t} + \tau^{-1}\beta t^{\beta-1}\mathbf{j} = \sigma\left(\frac{\partial \mathbf{e}}{\partial t} + \tau^{-1}\beta t^{\beta-1}\mathbf{e}\right)
\end{aligned}$$

where $\mathbf{e}$ is the electric field, $\mathbf{b}$ the magnetic flux density and $\mathbf{j}$ is electric current in the earth. The final equation defines the constitutive relation between $\mathbf{e}$ and $\mathbf{j}$ by an ordinary differential equation. The parameters in that equation define the nature of the induced polarization response of the system. η is the standard IP chargeability, τ is a time constant and $0 < \beta \leq 1$ controls the shape of the IP decay. This set of equations may be solved simultaneously to compute the time evolution of the electric field. Figure 3 shows results of simulating a gradient array with a long grounded wire source over a buried conductive block surrounded by a chargeable shell with IP decay time-constant $\tau = 0.1$. We see that the inline component of the electric field is dominated by EM induction at early times but that IP effects dominate at late times. Using the jInv framework, we were able to implement the stretched exponential forward modelling in less than 200 lines of Julia code.

Conclusions

In this abstract we have attempted to demonstrate the performance and flexibility of the jInv package. It seems to be well suited for electromagnetic simulation and inversion. As mentioned above the project is entirely open source and we encourage anyone who is interested to download the package and explore it for themselves.

REFERENCES

Amestoy, P. R., Duff, I. S., Koster, J., & L'Excellent, J.-Y. (2001). A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM Journal of Matrix Analysis and Applications*, *23*(1), 15–41.

Aminzadeh, F., Jean, B., & Kunz, T. (1997). *3-D salt and overthrust models.* Society of Exploration Geophysicists.

Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. (2014). Julia: A Fresh Approach to Numerical Computing. *pre-print.*

Cockett, R., Kang, S., Heagy, L. J., Pidlisecky, A., & Oldenburg, D. W. (2015). SimPEG: An open source framework for simulation and gradient based parameter estimation in geophysical applications. *Computers and Geosciences*, *85*, 142–154.

Commer, M., & Newman, G. (2004). A parallel finite-difference approach for 3D transient electromagnetic modeling with galvanic sources. *Geophysics*, *69*(5), 1192–1202.

Cox, L. H., Wilson, G. A., & Zhdanov, M. S. (2010). 3D inversion of airborne electromagnetic data using

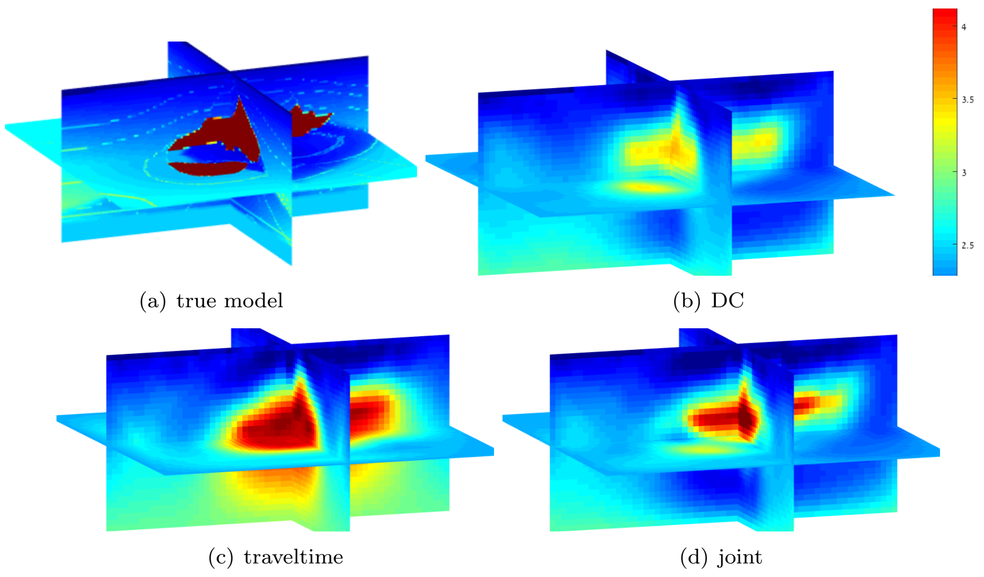

Figure 2: True SEG salt velocity model and recovered models from DC, traveltime and joint inversions. The colour scale is the same on all images.

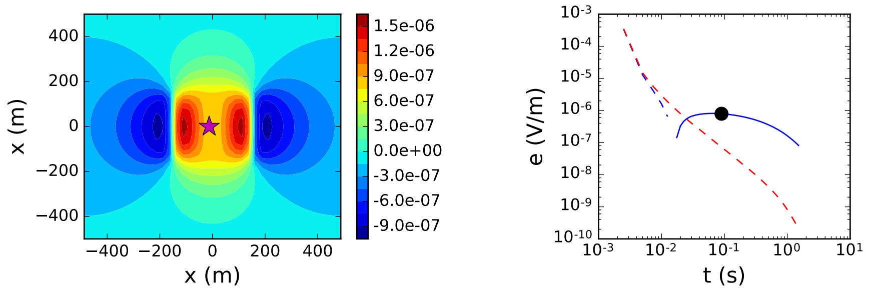

Figure 3: Time domain IP modelling results. Left figure shows the inline component of the electric field (pointing across the page) 0.1 seconds after source current step-off. The right hand figure shows the decay of the electric field at the point marked with a star. The red line shows the electric field computed when neglecting IP effects and the blue line shows the electric field computed from the full dispersive modelling. Dashed lines indicate negative.

a moving footprint. *Exploration Geophysics*, *41*(4), 250–259.

Haber, E. (2015). *Computational Methods in Geophysical Electromagnetics*. Philadelphia, PA: SIAM.

Rücker, C., Günther, T., & Wagner, F. (2017). *pyGIMLi: Geophysical Inversion and Modelling Library.*

Ruthotto, L., Treister, E., & Haber, E. (2016, jun). jInv – a flexible Julia package for PDE parameter estimation. *pre-print.*

Wilson, G., Aruliah, D. a., Brown, C. T., Chue Hong, N. P., Davis, M., Guy, R. T., et al. (2014). Best practices for scientific computing. *PLoS Biology*, *12*(1), e1001745.