
Lars Ruthotto
**jInv - A Flexible Framework for Parallel PDE Parameter
Estimation in Julia**

Emory University
Department of Mathematics and Computer Science
400 Dowman Drive
W408
30322 Atlanta
GA
USA
lruthotto@emory.edu
Eldad Haber

This talk presents jInv, a new framework for solving parameter estimation problems involving partial differential equations (PDEs). Parameter estimation problems of this kind arise frequently in many scientific disciplines, for example, in medical imaging, geophysical imaging, non destructive testing and economics. Parameter estimation is an inverse problem and typically ill-posed. The problem is typically formulated as an optimization problem with constraints that are given by the PDEs. The unknowns are parameters of the PDEs, which correspond to physical properties of the object to be measured. The objective is to minimize the misfit between PDE simulations and measured data plus some regularization term.

jInv is implemented in the relatively new programming language Julia [2], which offers a few advantages. First, Julia is a dynamic language therefore allows for rapid prototyping, is easy to read, change, distribute, and document. Second, Julia comes with a just-in-time compiler and therefore the runtime of the code can in many cases approach that of C or Fortran. Third, Julia provides tools that simplify parallel and distributed computing. Our code was initially developed on a standard laptop but supports also homogeneous clusters, as well as heterogeneous cloud computing services.

In this talk, we consider forward problems of the form

$$d_{ijk} = p_k^T u_{ij} + e_{ijk}, \quad \text{where} \quad A(m, \omega_i) u_{ij} = q_j, \quad (1)$$

for all $\forall i = 1, \dots, n_f, j = 1, \dots, n_s, k = 1, \dots, n_r$. Here, $A(m, \omega_i)$ is a discretization of the governing PDE for a discrete model m and additional parameter ω_i , for example, frequency. Further, p_k represent receivers, q_j the applied sources, and u_{ij} is the field caused by the i -th source and the j -th frequency. The noise, e_{ijk} is assumed to identically distributed white noise. It is important to note that computing the data requires the computation of $n_f \cdot n_s$ fields associated with n_f different linear systems and n_s right hand sides.

In practical applications, the number of sources and frequencies is generally large, and so a large number of PDE solves is required in the inversion. Thus, efficient methods for solving the PDEs are essential. In jInv, one can choose between state-of-the-art iterative or direct methods depending on the problem size and computational resources. Also, jInv supports both regular and stretched tensor meshes that can be adapted to the source locations.

For given data, sources, receivers, and frequencies, the goal of the parameter estimation problem is to estimate the model m . Therefore, we consider the bound-constrained discrete optimization problem

$$\min_m \sum_{ijk} \phi(p_i^\top A(\omega_j, m)^{-1} q_k, d_{ijk}) + \alpha R(m) \quad \text{subject to} \quad m_{\text{low}} \leq m \leq m_{\text{up}}. \quad (2)$$

Here, ϕ measures the misfit of the predicted and measured data, R is a regularizer, and m_{low} and m_{up} are lower- and upper bounds on the components of m , respectively. The parameter $\alpha > 0$ balances between the misfit and regularizer. In its current version jInv provides some commonly used misfit functions (weighted L_2 and L_1 norms) and regularization (diffusion, total variation) that can be exchanged modularly.

The optimization problem is solved using a projected Gauss-Newton method that uses a Steepest Descent step to identify the active set. The Hessian approximation of the data term is implemented in a matrix-free fashion, since it involves $A(m, \omega_i)^{-1}$ for $i = 1, \dots, n_f$, which are in general large and dense. Intermediate results of the PDE solver such as factorizations or preconditioners are re-used to accelerate computations with the Hessian. A projected PCG method is used for determining the search direction and projected Armijo condition is used for line search.

A particular focus of the talk is Julia's potential for parallel and distributed computing that is used in jInv to reflect the parallel structure of the misfit term in (2). Note that all terms can be computed independently and in parallel. Therefore a variety of parallelization strategies are supported by jInv that can be tested to exploit the given computational resources.

We use a distributed data model that divides the data, sources, receivers and frequencies among all available workers and therefore parallelizes evaluation of the misfit. One benefit is the reduced memory consumption, in particular regarding the fields and intermediate results from the PDE solves that are held in memory by the respective worker. Another benefit of this model is the reduction of communication costs, since only the model m as well as vectors of equivalent size must be communicated.

The talk will present numerical examples geophysical imaging. Particular emphasis is on the parallel implementation using Julia. We will show the scalability of the code on a single-user laptop, a large parallel workstation, and a cloud computing engine.

Bibliography

- [1] E. Haber. *Computational Methods in Geophysical Electromagnetics*. SIAM, Philadelphia, 2014.
- [2] Jeff Bezanson, Stefan Karpinski, Viral B Shah, and Alan Edelman. Julia: A Fast Dynamic Language for Technical Computing. *arXiv.org*, 2012.