A Computationally Efficient Levenberg-Marquardt Algorithm and Its Application to Hydrogeologic Inverse Modeling

Youzuo Lin, Daniel O'Malley, and Velimir V. Vesselinov

Los Alamos National Laboratory
Earth and Environmental Sciences Division
Los Alamos, NM 87545

May 25, 2016

Outline

1. Introduction
2. Hydrogeologic Inverse Modeling
   - Forward Problem - Groundwater Flow Equation
   - Inverse Problem - Damped Least-Squares Problem
3. Efficient Hydrogeologic Inverse Modeling Method
   - Numerical Optimization Methods
   - Levenberg-Marquardt Method
4. Computational Cost Analysis
5. Numerical Results
6. Conclusions

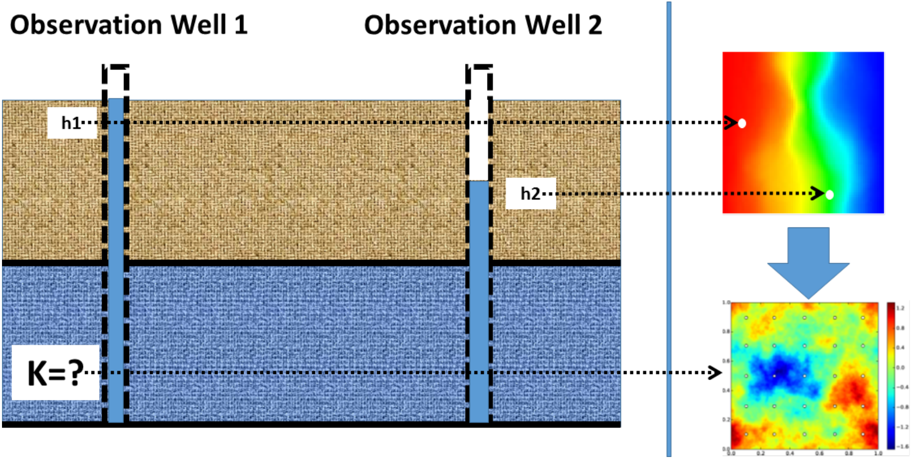
Outline

Introduction

- Inverse modeling in hydrogeology seeks the characterization of spatially distributed parameters defined over a model domain based on observations of state variables.
- We employ a gradient-based numerical optimization method, Levenberg-Marquardt method, to solve the hydrogeologic inverse modeling problem.
- The core of the Levenberg-Marquardt algorithm involves the selection of the damping parameter and the linear solve for the search direction at every iteration.
- The linear solve can be computationally intensive, which hinders the applications of LM-algorithm based inverse modeling methods to large-scale or even moderate hydrogeology models.
- We apply a computationally efficient Krylov-subspace-recycled iterative linear solver to solve the linear system at every iteration.

Outline

Hydrogeologic Inverse Modeling - Illustration



- **Input:** Measured values (hydraulic heads) at *N* observation wells.
- **Output:** Model parameter values (conductivity or transmissivity) at every grid node of the model.

Forward Problem

The forward problem of hydrogeologic inverse modeling is governed by the groundwater flow equation,

Groundwater Flow Equation

$$\nabla \cdot (T \nabla h) = g$$
$$g(x, y) = 0$$
$$\left.\frac{\partial h}{\partial x}\right|_{a,y} = \left.\frac{\partial h}{\partial x}\right|_{b,y} = 0$$
$$h(x, c) = 0, h(x, d) = 1$$

where h is the hydraulic head, T is the transmissivity and g is a source/sink (here, set to zero).

Forward Problem

Using the operator, the forward modeling problem of the hydrogeologic inverse modeling can be simplified as,

Groundwater Flow Equation - Operator Form

$$\mathbf{h} = f(\mathbf{T}),$$

where $f(\cdot)$ is the forward operator mapping from the model parameter space to the measurement space.

Inverse Problem

Correspondingly, the problem of model calibration can be posed as a damped least-squares problem,

Hydrogeologic Inverse Modeling

$$\begin{aligned}\mathbf{x} &= \arg\min_{\mathbf{x}} \{f(\mathbf{x})\}, \\ &= \arg\min_{\mathbf{x}} \left\{ \|\mathbf{d} - f(\mathbf{x})\|_R^2 + \lambda \|\mathbf{x} - \mathbf{x_0}\|_Q^2 \right\},\end{aligned}$$

where $\mathbf{d}$ represents a recorded hydraulic head dataset, $\mathbf{x}$ is the calibrated model parameter, $\mathbf{x_0}$ is the prior model parameters, $\|\mathbf{d} - f(\mathbf{x})\|_2^2$ measures the data misfit, $||\cdot||_2$ stands for the L_2 norm, and R is the covariance matrix for the data error and Q is the covariance matrix for the model parameters.

Outline

Numerical Optimization Methods - Fundamental

- Line search optimization is an iterative method usually posed as,

Line search optimization

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)}\mathbf{p}^{(k)},$$

where k is the iteration index, the vector $\mathbf{p}^{(k)}$ is the search direction and $\alpha^{(k)}$ is the step length.

- Different optimization methods are developed according to the selection of the descent direction, $\mathbf{p}^{(k)}$.
  - First-Order Method: Steepest Descent Method
  - Second-Order Method: Newton-Type Methods and **Levenberg-Marquardt (LM) Method**.

Numerical Optimization Methods - Fundamental

- Search direction of Steepest Descent Method, Newton-Type Methods and Levenberg-Marquardt (LM) Method:
  - Steepest Descent Method: $\mathbf{p}^{(k)} = -\nabla f^{(k)}$.
  - Newton-Type Methods: $\mathbf{p}^{(k)} = -((J^{(k)})'J^{(k)} + S^{(k)})^{-1}\nabla f^{(k)}$, where $J^{(k)} = J(\mathbf{x}^{(k)})$ is the Jacobian matrix for the model parameter $\mathbf{x}^{(k)}$ and $S^{(k)}$ is the higher-order term in Hessian.
  - Levenberg-Marquardt (LM) Method: $\mathbf{p}^{(k)} = -\left[(J^{(k)})'J^{(k)} + \mu\,\mathrm{diag}((J^{(k)})'J^{(k)})\right]^{-1}\nabla f^{(k)}$, where μ is the damping parameter and in the Levenberg version of the LM method, $J^{(k)} = I$.
- We choose **Levenberg-Marquardt (LM)** Method because:

Numerical Optimization Methods - Fundamental

- Search direction of Steepest Descent Method, Newton-Type Methods and Levenberg-Marquardt (LM) Method:
  - Steepest Descent Method: $\mathbf{p}^{(k)} = -\nabla f^{(k)}$.
  - Newton-Type Methods: $\mathbf{p}^{(k)} = -((J^{(k)})' J^{(k)} + S^{(k)})^{-1} \nabla f^{(k)}$, where $J^{(k)} = J(\mathbf{x}^{(k)})$ is the Jacobian matrix for the model parameter $\mathbf{x}^{(k)}$ and $S^{(k)}$ is the higher-order term in Hessian.
  - Levenberg-Marquardt (LM) Method: $\mathbf{p}^{(k)} = -\left[(J^{(k)})' J^{(k)} + \mu \operatorname{diag}((J^{(k)})' J^{(k)})\right]^{-1} \nabla f^{(k)}$, where μ is the damping parameter and in the Levenberg version of the LM method, $J^{(k)} = I$.
- We choose **Levenberg-Marquardt (LM)** Method because:
  - LM method can be superior to steepest descent or Newton-type methods in that it converges much faster than steepest descent and is more robust to the initial guess than Newton-type methods.

Numerical Optimization Methods - Fundamental

- Search direction of Steepest Descent Method, Newton-Type Methods and Levenberg-Marquardt (LM) Method:
  - Steepest Descent Method: $\mathbf{p}^{(k)} = -\nabla f^{(k)}$.
  - Newton-Type Methods: $\mathbf{p}^{(k)} = -((J^{(k)})' J^{(k)} + S^{(k)})^{-1} \nabla f^{(k)}$, where $J^{(k)} = J(\mathbf{x}^{(k)})$ is the Jacobian matrix for the model parameter $\mathbf{x}^{(k)}$ and $S^{(k)}$ is the higher-order term in Hessian.
  - Levenberg-Marquardt (LM) Method: $\mathbf{p}^{(k)} = -\left[(J^{(k)})' J^{(k)} + \mu \,\mathrm{diag}((J^{(k)})' J^{(k)})\right]^{-1} \nabla f^{(k)}$, where μ is the damping parameter and in the Levenberg version of the LM method, $J^{(k)} = I$.
- We choose **Levenberg-Marquardt (LM)** Method because:
  - LM method can be superior to steepest descent or Newton-type methods in that it converges much faster than steepest descent and is more robust to the initial guess than Newton-type methods.
  - LM method can be more stable than either steepest descent method or Newton-type method in the cases when the inverse problem becomes ill-posed.

Conventional Levenberg-Marquardt Method

- The LM method can be seen as a combination of the steepest descent method and Newton-type methods.
- The damping parameter of μ plays an important role in ensuring the search direction in the parameter space provides an optimal balance between first-order and second-order optimization steps.
- The heuristic to update the damping parameter, $\mu^{(k)}$,

$$\mu^{(k+1)} = \begin{cases} \beta \cdot \mu^{(k)} & \text{if } \rho < \rho_1 \\ \frac{\mu^{(k)}}{\gamma} & \text{if } \rho > \rho_2 \\ \mu^{(k)} & \text{otherwise} \end{cases},$$

and the gain factor, ρ, can be defined as,

$$\rho = \frac{f(\mathbf{x}) - f(\mathbf{x} + \mathbf{h})}{L(\mathbf{0}) - L(\mathbf{h})}.$$

Conventional Levenberg-Marquardt Method

Algorithm 1 Conventional Levenberg-Marquardt Method - Major Steps

if {Jacobian needs updated} **then**
    Calculate the new Jacobian matrix;
end if
Solve for the search direction $\mathbf{p}^{(k)}$;
if {Stopping criterion are satisfied} **then**
    Return with solution $\mathbf{x}^{(k)}$;
else
    Obtain the current solution, $\mathbf{x}_{\text{new}} = \mathbf{x}^{(k)} + \mathbf{p}^{(k)}$;
    if {Damping parameter is appropriate} **then**
        Update the iteration, $\mathbf{x}^{(k+1)} = \mathbf{x}_{\text{new}}$;
    else
        Update the damping parameter μ;
    end if
end if

Conventional Levenberg-Marquardt Method

- In most existing hydrogeologic inverse modeling, direct solvers such as QR decomposition or singular value decomposition (SVD) based methods are used to solve for $\mathbf{p}^{(k)}$.
- These existing hydrogeologic inverse modeling methods can be rather computationally expensive for two reasons:
    - The Jacobian matrix can be large and sparse, therefore the direct methods will not appropriate.
    - The re-calculation of $\mathbf{p}^{(k)}$ can be expensive when searching for the optimal damping parameter.
- **How can we improve the computational efficiency?**

LM Method Revisit - Exploring the Matrix Structure

- The **Levenberg version** of the LM Method:
  $\mathbf{p}^{(k)} = -\left[(J^{(k)})' J^{(k)} + \mu I\right]^{-1} \nabla f^{(k)}$ can be posed equivalently as a matrix form,

Levenberg-Marquardt Method in Matrix Form

$$\mathbf{p}^{(k)} = \arg\min_{\mathbf{p}^k} \left\{ \left\| \begin{bmatrix} J^{(k)} \\ \sqrt{\mu}\, I \end{bmatrix} \mathbf{p}^{(k)} - \begin{bmatrix} -r^{(k)} \\ 0 \end{bmatrix} \right\|_2 \right\},$$

- We observe that,
  - The system matrices consist of two parts: the Jacobian matrix $J^{(k)}$ and the diagonal matrix and both of them can be large and sparse when the measurements and model increases.
  - At any iteration, the Jacobian matrix remains the same while the damping parameter μ can vary.

Efficient LM Method - Krylov Subspace Solvers

- Definition of Krylov Subspace,

$$\mathcal{K}_n(A, \mathbf{r}_0) = \text{span}\left\{\mathbf{r}_0, A\,\mathbf{r}_0, A^{(2)}\,\mathbf{r}_0, \ldots, A^{(n-1)}\,\mathbf{r}_0\right\}$$

- The basic idea of a Krylov solver is to construct a sequence of approximations getting closer to the exact solution **x**, such that

$$\mathbf{x_n} \in \mathbf{x_0} + \mathcal{K}_n(A, \mathbf{r}_0)$$

- We select the LSQR method, a type of Krylov Subspace Method, considering its superior performance of accuracy and efficiency in solving large-scale ill-posed problems.

Efficient LM Method - LSQR Iterative Method

- The Krylov subspace generated at the k^{th} step using LSQR method,

$$\mathcal{K}_k = \text{span}\left\{(J^{(k)})' J^{(k)} + \mu I, -(J^{(k)})' r^{(k)}\right\},$$
$$= \text{span}\left\{(J^{(k)})' J^{(k)}, -(J^{(k)})' r^{(k)}\right\}.$$

  - We observe that the Krylov subspace generated for the damped least-squares problem is independent of the damping parameter μ.
  - This gives us the hint to generate a common subspace using a initial damping parameter and project the remaining damping down-to the generated subspace.

Efficient LM Method - Recycled LSQR Method

- Krylov Subspace Generate Step
  - The Golub-Kahan-Lanczos (GKL) bidiagonalization technique

$$
\begin{aligned}
&\beta^{(1)} U^{(k+1)} \mathbf{e}^{(1)} = \mathbf{b}, \\
&A V^{(k)} = U^{(k+1)} B^{(k)}, \\
&A' U^{(k+1)} = V^{(k)} (B')^{(k)} + \alpha^{(k+1)} V^{(k+1)} \mathbf{e}'^{(k+1)},
\end{aligned}
$$

  where the unit vector $\mathbf{e}^{(i)}$ has value 1 at the i^{th} location and zeros elsewhere, i.e., $\mathbf{e}^{(i)} = \left[0, \ldots, 1, \ldots 0\right]$.

  - The GKL bidiagonalization procedure also generates a subspace, which is spanned by the column vectors in V_k, i.e.,

$$
\mathcal{K}_k = \mathrm{span}(V^{(k)}) = \mathcal{K}_k(A' A, A' \mathbf{b}).
$$

Efficient LM Method - Recycled LSQR Method

- Subspace Projection and Recycling Step
  - A three-term-recursion to update the solution $\mathbf{x}^{(k)}$ at each iteration step can be obtained,

$$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + \phi^{(k)}\mathbf{z}^{(k)},$$
$$\mathbf{z}^{(k)} = \frac{1}{\rho^{(k)}}(\mathbf{v}^{(k)} - \theta^{(k-1)}\mathbf{z}^{(k-1)}).$$

- The major computational cost is the GKL recursion procedure in generating the Krylov subspace. The three-term-recursion procedure to update the solution by projection is computationally efficient in comparison.

Efficient LM Method - Summary

Algorithm 2 Efficient Levenberg-Marquardt Method - Solution of Search Direction

1: **if** {Initial damping parameter} **then**
2: Generate the Krylov subspace;
3: **else**
4: Recycle the subspace generated previously;
5: **end if**
6: Solve the search direction $\mathbf{p}^{(k)}$ by projection;

- Extension to **Marquardt's version** of the LM method - Variable Substitution

$$\bar{\mathbf{p}}^{(k)} = \arg\min_{\bar{\mathbf{p}}^k} \left\{ \left\| \bar{J}^{(k)} \bar{\mathbf{p}}^{(k)} - (-r^{(k)}) \right\|_2^2 + \mu \|\bar{\mathbf{p}}^{(k)}\|_2^2 \right\},$$

where $\bar{J}^{(k)} = J^{(k)} D^{-1}$ and $D = \mathrm{diag}\left((J^{(k)})' J^{(k)} \right)$.

Outline

Computational Cost Analysis - Setup

- Assume that the number of model parameter is $\tilde{m}$, the number of observations is $\tilde{n}$, hence the size of the Jacobian matrix $\tilde{n} \times \tilde{m}$.
- As a reference method, we choose the linear solver via the most often used QR decomposition to solve the LM search directions, denoted as **"LM-QR"**.
- We denote our new LM method as **"LM-RLSQR"** and **"R"** stands for "recycled".
- We report both the computational costs using the initial damping parameter as well as using the rest of the damping parameters.

Computational Cost Analysis - LM-QR Method

- Given an initial guess of the damping parameter, the associated computational costs are

$$\mathrm{COST}_{LM-QR-Initial} \approx \mathcal{O}(\tilde{n} \times \tilde{m}^2) + \mathcal{O}(\tilde{m}^3) + \mathcal{O}(\tilde{n} \times \tilde{m}) + \mathcal{O}(\tilde{m}^2),$$

  where the first term is associated with forming the normal equation, the second term associating with the QR factorization, the third term associating with forming the right hand-hand side, and the fourth term associating with the back-substitution for the solution.

- Once the damping parameter is updated, some of the calculation can be saved and reused. However, the expensive QR factorization and the back-substitution cannot be avoided, therefore the costs for the updated damping parameter will be,

$$\mathrm{COST}_{LM-QR-Rest} \approx \mathcal{O}(\tilde{m}^3) + \mathcal{O}(\tilde{n} \times \tilde{m}) + \mathcal{O}(\tilde{m}^2).$$

Computational Cost Analysis - Efficient LM Method

- Assuming the dimension of the Krylov subspace to be k_1, the cost associated using the initial damping parameter is,

$$\mathrm{COST}_{LM-RLSQR-Initial} \approx k_1 \cdot \mathcal{O}(\tilde{m} \times \tilde{n}),$$

- The computational cost for solving for the search directions of the rest of the damping parameters is,

$$\mathrm{COST}_{LM-RLSQR-Rest} \approx k_1(n-1) \cdot \mathcal{O}(\tilde{m}).$$

  where n is the number of μ values that are being used.

- To compare the total computational costs associated with the **LM-QR** and **LM-RLSQR**, we conclude that the cost associated with **LM-QR** is much more expensive than the one with **LM-RLSQR**.

Outline

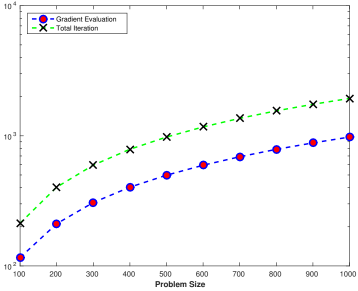
Performance on Benchmark Testing Functions

Problem	Function Name	Reference
1	Dixon-Price	Dixon and Price, 1989
2	Griewank	Locatelli, 2003
3	Powell	Powell, 1964
4	Rosenbrock	Dixon and Szego, 1978
5	Rotated Hyper-Ellipsoid	Molga and Smutnicki, 2005
6	Sphere	Picheny et al., 2013
7	Sum Squares	Hedar, 2013

Table: Set of benchmark testing functions

- The results of the following are reported,
  - **Linear Solver Time** V.S. **Total Time**
  - **Number of Gradient Evaluation** V.S. **Number of Total Iteration**

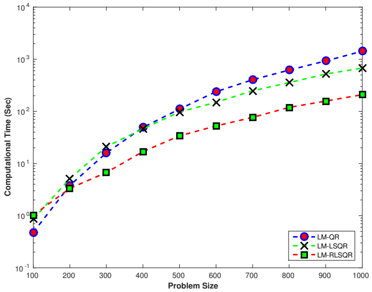
Performance on Benchmark Testing Functions



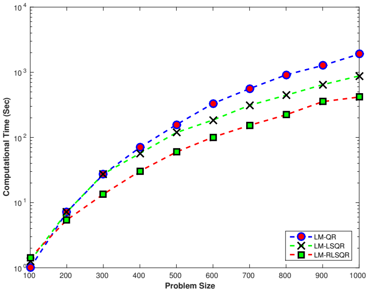
Gradient Evaluation V.S. Total Iteration on Rosenbrock function

- Significant amount of trials are needed for the optimal damping parameters at every LM iteration.

Performance on Benchmark Testing Functions



Linear System

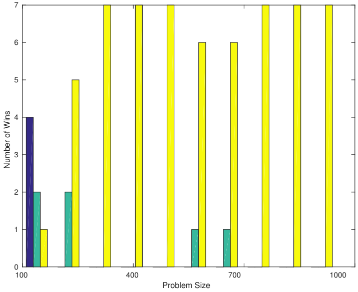


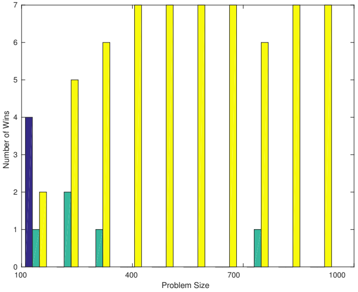
Overall

Time Profiles on the Rosenbrock function

- The “LM-QR” (red circle) wins the most when problem size is small.
- The “LM-RLSQR” (green box) dominates for most of the testing cases.

Performance on Benchmark Testing Functions





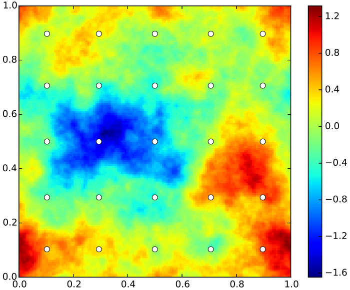
Linear System

Overall

Counts of Wins on the Computational Time Costs

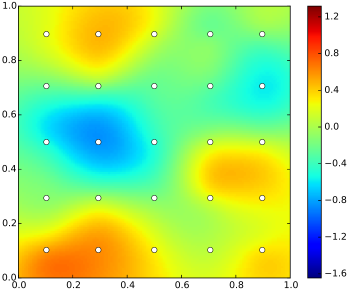
- The "LM-QR" (in blue) wins the most when problem size is small.
- As the size of the problem increases, "LM-LSQR" (in cyan) wins occasionally and "LM-RLSQR" (in yellow) wins most of times.
- When the size of the problem becomes large, "LM-RLSQR" dominates the other two methods.

Model Calibration in Hydrology - True Model

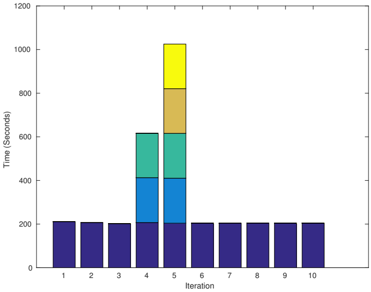


True Model

- Synthetic transmissivity field.
- Hydraulic conductivity and hydraulic head observation locations are indicated with circles.
- Model dimension, 69×69, (a total of 9660 model parameters)



Inversion

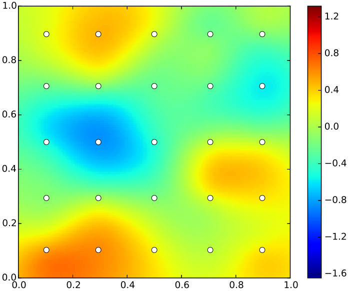


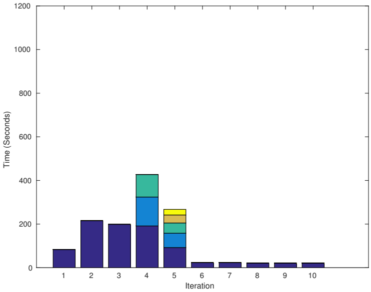
Time Profile

Results of LM-QR Method

- A total number of 10 iteration steps are needed before a full convergence.
- At iteration step of 4 and 5, there are multiple trials needed to search for the damping parameter.

Model Calibration in Hydrology - Inversion Results





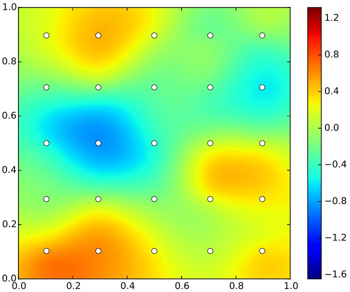
Inversion

Time Profile

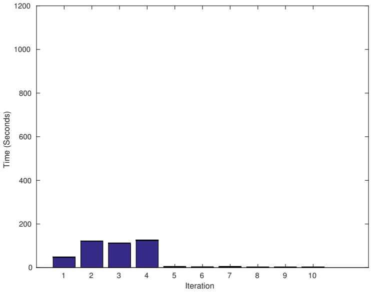
Results of LM-LSQR Method

- At every iteration step, the length of the blocks at the first trial are mostly shorter than those obtained using LS-QR method.
- Each extra trial for an acceptable LM descent direction yields less computational time by comparing to LS-QR method.

Model Calibration in Hydrology - Inversion Results



Inversion

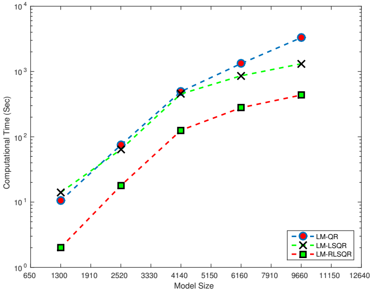


Time Profile

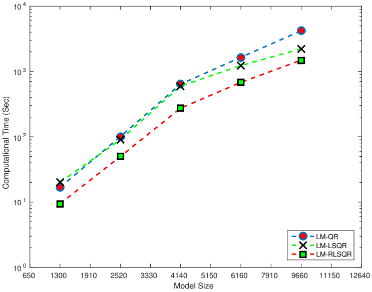
Results of LM-RLSQR Method

- At iteration steps 4 and 5, the same number of trials for an acceptable LM damping parameter are needed.
- The time costs are significantly saved, even though they are hard to visualize because of the small time costs associated.

Model Calibration in Hydrology - Time Costs



Linear System



Overall

- Five different model sizes including 1300, 2520, 4140, 6160, and 9660 are tested.
- Both time costs on linear solver and the total time using our LM-RLSQR method are always less than the time costs of the other two methods for these problems.

Outline

Conclusions

- We have developed an approach to hydrogeologic inverse modeling employing our new computationally efficient Levenberg-Marquardt algorithm.
- we recycle the Krylov subspace in-between linear systems sharing the same Jacobian matrix, but different damping parameters.
- Through our numerical results, we show that our new LM method yields an improved computational efficiency over both “LM-QR” and “LM-LSQR” methods.
- We implement our new LM algorithm using Julia in the MADS computational framework (http://madsjulia.lanl.gov/), which can be downloaded at https://github.com/madsjulia/Mads.jl.

Acknowledgement

- Youzuo Lin and Velimir V. Vesselinov were support by LANL Environmental Programs Projects.
- Daniel O'Malley was supported by a LANL Director's Postdoctoral Fellowship,
- In addition, Velimir V. Vesselinov was supported by the DiaMonD project (An Integrated Multifaceted Approach to Mathematics at the Interfaces of Data, Models, and Decisions, U.S. Department of Energy Office of Science, Grant #11145687).