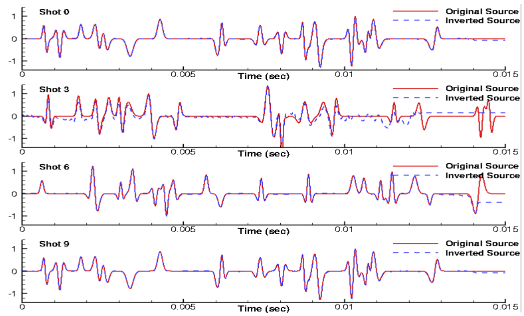
SAND2014-15694PE



Large-Scale Optimization for Non-Invasive Testing with Discontinuous Galerkin Methods

Curtis Ober*, Thomas Smith,

Bart van Bloemen Waanders, and Scott Collis

Sandia National Laboratories

Exceptional service in the national interest

U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2011-XXXXP

Non-Invasive Testing

- … determine the characteristics engineering systems without disassembling or damaging them.
- Source Inversion
  - Interested in mimicing operational conditions within an experimental setup to test conditions throughout the system
  - Direct Field Acoustic Testing – DFAT
    - Acoustic testing of aerospace structures
    - Use an array of acoustic drivers
- Material inversion
  - Characterize material properties
  - Aging, cracks
  - Miss-assembly
  - DFAT and ultrasonic testing

Direct Field Acoustic Testing

http://en.wikipedia.org/wiki/Direct_Field_Acoustic_Testing

Constrained Optimization Problem

$$\min_{\mathbf{\Phi}} J(\mathbf{U}, \mathbf{\Phi}) \equiv \frac{1}{2} \sum_{s=1}^{N_s} \sum_{r=1}^{N_r} \int_{\mathsf{Q}} \xi_{s,r}(\mathbf{x}) \, \left\| \mathsf{R} \left(\mathbf{U}_s - \hat{\mathbf{U}}_s \right) \right\|^2 \, d\mathsf{Q} + \frac{\beta}{2} \int_{\Omega} \left\| \mathsf{R} \left(\mathbf{\Phi} \right) \right\|^2 \, d\Omega$$

- Subject to the acoustic wave equation

$$\frac{\partial p}{\partial t} + \kappa \frac{\partial v_i}{\partial x_i} = -\frac{1}{3} \frac{\partial m_{ii}^{iso}}{\partial t} \quad \text{in } \Omega \times (0, T]$$

$$\rho \frac{\partial v_i}{\partial t} + \frac{\partial p}{\partial x_i} = f_i + \frac{\partial m_{ij}^{dev}}{\partial x_j} \quad \text{in } \Omega \times (0, T]$$

$$p(\mathbf{x}, 0) = 0 \quad \text{for } \mathbf{x} \in \Omega$$

$$v_i(\mathbf{x}, 0) = 0 \quad \text{for } \mathbf{x} \in \Omega$$

$$\mathbf{U} = (p, v_i)^T$$

$$\kappa = \rho c^2 - \text{bulk modulus}$$

$$\rho = \text{mass density}$$

$$c = \text{wave speed}$$

$$f_i = \text{force density}$$

$$m_{ij} = m_{ij}^{iso} + m_{ij}^{dev} - \text{moment density tensor}$$

$$m_{ij}^{iso} \equiv \frac{1}{3} m_{kk} \delta_{ij}$$

$$\Omega = \text{computational domain}$$

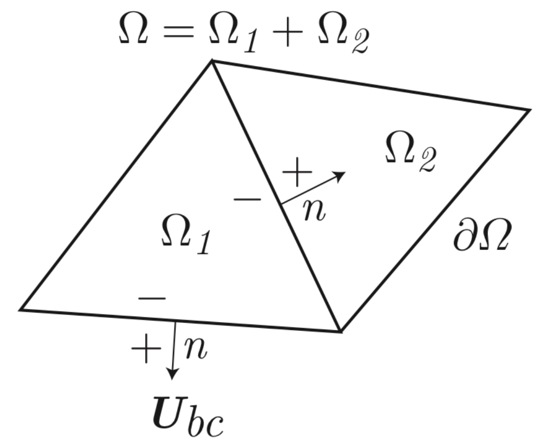
$$T = \text{time horizon}$$

$$\xi_{s,r} = \text{receiver spatial kernel for shot } s$$

$$\mathsf{R} = \text{selection operator}$$

$$\mathbf{\Phi} = \text{optimization parameters}$$

Discontinuous Galerkin



Strong form:

$$\mathbf{U}_{,t} + \mathbf{F}_{i,i} = \mathbf{S}, \quad \text{in } \Omega$$

$$\mathbf{U}(x,0) = \mathbf{U}_0(x) \quad \text{at } t = 0$$

and appropriate boundary conditions on $\partial\Omega$.

Partition Ω into N subdomains Ω_e.

$$\int_{\Omega_e} \left(\mathbf{W}^T\mathbf{U}_{,t} - \mathbf{W}^T_{,i}\mathbf{F}_i\right) d\Omega + \int_{\partial\Omega_e} \mathbf{W}^T\mathbf{F}_n \, d\Gamma = \int_{\Omega_e} \mathbf{W}^T\mathbf{S} \, d\Omega$$

Introduce numerical fluxes $\mathbf{F}_n(\mathbf{U}) \rightarrow \hat{\mathbf{F}}_n(\mathbf{U}^-, \mathbf{U}^+)$ and sum over all elements

$$\sum_{e=1}^{N_e} \int_{\Omega_e} \left(\mathbf{W}^T\mathbf{U}_{,t} - \mathbf{W}^T_{,i}\mathbf{F}_i - \mathbf{W}^T\mathbf{S}\right) d\Omega + \int_{\partial\Omega_e} \mathbf{W}^T\hat{\mathbf{F}}_n \, d\Gamma = 0$$

$$\sum_{e=1}^{N_e} \int_{\Omega_e} \left(\mathbf{W}^T\mathbf{U}_{,t} + \mathbf{W}^T\mathbf{F}_{i,i} - \mathbf{W}^T\mathbf{S}\right) d\Omega + \int_{\partial\Omega_e} \mathbf{W}^T \left(\hat{\mathbf{F}}_n - \mathbf{F}_n\right) \, d\Gamma = 0$$

for all $\boldsymbol{W} \in \mathcal{V}$.

DGM Software

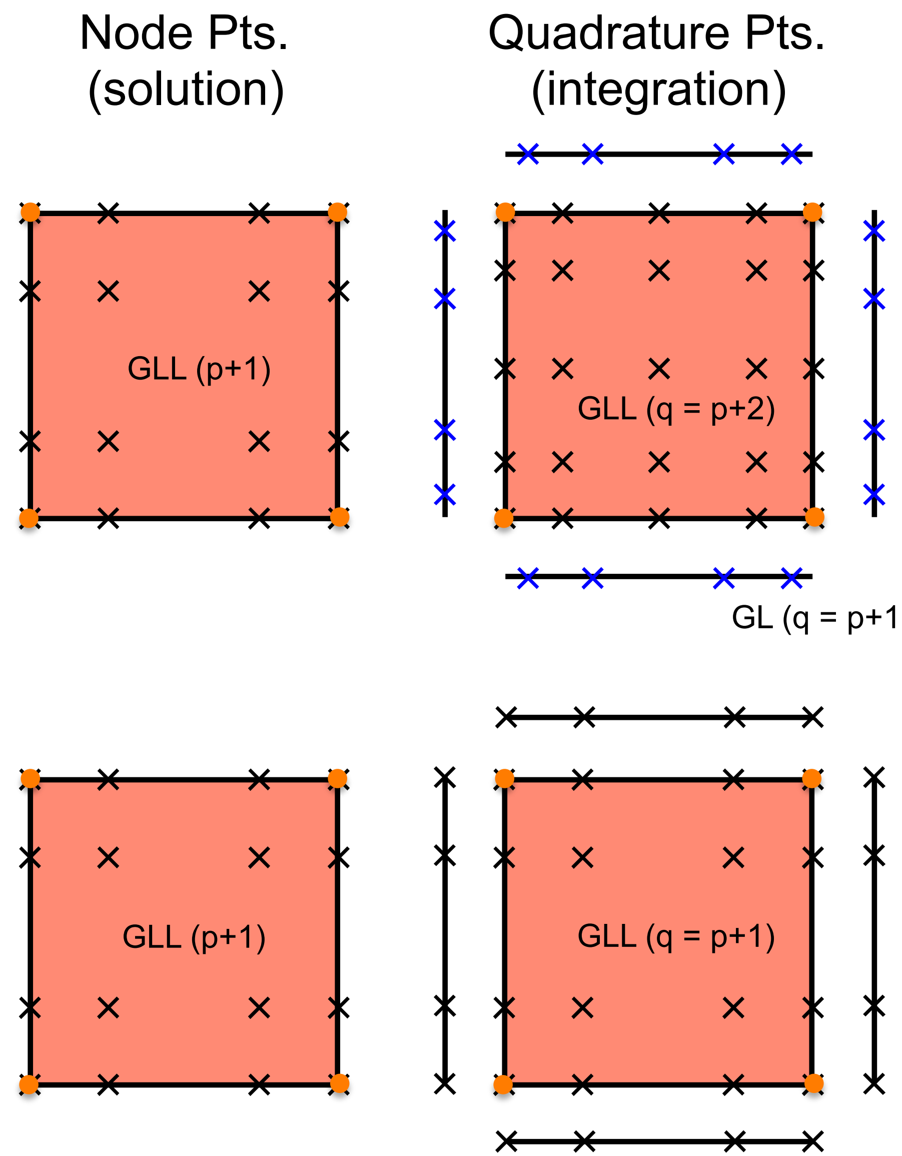
- Variety of physics
  - In- & compressible Euler, compressible Navier-Stokes (with LES)
  - Darcy, Helmholtz, Wave equation
- High-order DG on unstructured meshes (low dispersion)
  - Modal elements: Line, Quad, Tri, Hex
  - Nodal elements: Line, Quad, Tri, Hex, Tet
  - Spectral Elements: Line, Quad, Hex
  - Curved elements and hybrid meshes
  - Variable-order polynomial representation (For both solution and media)
  - Local polynomial and mesh refinement
  - Various flux functions (Lax-Friedrichs, Steger-Warming, Riemann)
- Range of explicit and implicit time integrators
  - Forward & Backward Euler, Trapezoidal, 4^{th} Order Runge-Kutta (low dispersion)
- Designed for adjoint-based optimization
  - ROL, PEOpt, Dakota, MOOCHO, Aristos
  - Steady-state and transient with check pointing

Modal vs. Nodal vs. Spectral

	Modal (Karniadakis & Sherwin, 2005, Collis, 2002)	**Nodal** (Hesthaven & Warburton, 2000; Kaser & Dumbser, 2006, 2008)	**Spectral** (Karniadakis & Sherwin, 2005)
Elements	Line, Tri,Quad, Hex	Line, Tri, Quad, Hex, Tet	Line, Quad, Hex
DoFs	coefficients of polynomial bases	nodes of the Lagrange polynomials	GLL nodes
Interpolation	transform/Lagrange	Lagrange	Lagrange sampled at Legendre zeros
Mass term integration	constant for affine elements, Gauss quadrature for non-affine	orthogonal bases – exact integration for affine elements, Gauss quadrature for non-affine	diagonal for affine and non-affine elements
Volume Integration	Gauss-Lobatto quadrature Q≥P+2; exact integration	orthogonal bases – exact analytic	Gauss-Lobatto quadrature, Q=P+1
Surface integration/ Flux data	interpolate on face, Gauss quadrature Qedg=P+1	data is native to face – no interpolation- exact integration	no interpolation for constant P, face interpolation for non-const. P
Media	variable within element	constant within element	variable within element
Element Jacobian	variable within element	constant over element	variable within element
Non-linear terms	increase number of quadrature points GLL	filter or use quadrature	filter high modes (de-alias)

Modal and Spectral Elements

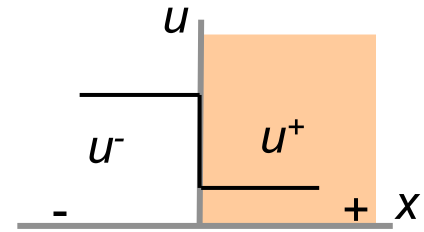
- Modal (more accurate)
  - Uses Legendre basis (modes)
  - Uses quadrature for all integrals
  - Can exactly integrate higher-order polynomial products by increasing quadrature ($q \geq p+2$)
  - Requires interpolation from nodes to quad pts.
  - For affine elements, mass matrix is constant and easily inverted and stored.
  - For non-affine elements, mass matrix is dense and needs inverting every time step and stage.
- In Common
  - Uses tensor-products of orthogonal polynomials
  - Support spatially variable polynomial order
  - Variable media, nonlinearities, & non-affine elements produce higher-order polynomial products
- Spectral (less accurate; less expensive)
  - Uses Lagrange basis on optimal node distributions
  - Under integrates higher-order polynomial products because of fixed quadrature ($q = p+1$)
  - May need to resort to modal or inexact quadrature + filtering for higher-order polynomial products
  - No interpolation required.
  - For affine & non-affine elements, mass matrix is diagonal and easily inverted.



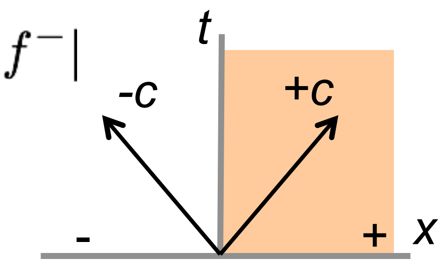
× - Gauss-Labatto-Legendre (GLL); × - Gauss-Legendre (GL)

Numerical Fluxes

Sandia National Laboratories



- Riemann Flux (Exact flux)
  - Conversative form $u_{,t} + f_{,x} = 0$ not $u_{,t} + c\,u_{,x} = 0$
  - Rankine-Hugoniot Jump Conditions $c|u^+ - u^-| = |f^+ - f^-|$
  - Entropy Conditions



- Flux, flux Jacobian, and eigenvalues are

$$\boldsymbol{F}_n = (\boldsymbol{A}_n \boldsymbol{U}) \qquad \boldsymbol{A}_i \equiv \frac{\partial \boldsymbol{F}_i}{\partial \boldsymbol{U}}$$

$$\begin{aligned} \boldsymbol{\Lambda} &= \quad diag\{\lambda_1, ..., \lambda_N\} \\ &= \quad diag\{eigen(\boldsymbol{A}_n)\} \end{aligned}$$

- Lax-Friedrich Flux

$$\hat{\boldsymbol{F}}_n(\boldsymbol{U}^-, \boldsymbol{U}^+) = \frac{1}{2}\left[\boldsymbol{F}_n^- + \boldsymbol{F}_n^+ + \xi\left(\boldsymbol{U}^- - \boldsymbol{U}^+\right)\right] \qquad \xi = \max|\lambda_i|$$

- Steger-Warming Flux (Flux-Vector Splitting)

$$\boldsymbol{A}_n = \boldsymbol{A}^+ + \boldsymbol{A}^- = \boldsymbol{T}\boldsymbol{\Lambda}^+\boldsymbol{T}^{-1} + \boldsymbol{T}\boldsymbol{\Lambda}^-\boldsymbol{T}^{-1} \qquad \boldsymbol{\Lambda}^\pm = \left(\boldsymbol{\Lambda} \pm |\boldsymbol{\Lambda}|\right)/2$$

$$\hat{\boldsymbol{F}}_n = \frac{1}{2}\left[\boldsymbol{A}_n^- \boldsymbol{U}^- + \boldsymbol{A}_n^- \boldsymbol{U}^+ + |\boldsymbol{A}_n|\left(\boldsymbol{U}^- - \boldsymbol{U}^+\right)\right] \qquad |\boldsymbol{A}| = \boldsymbol{T}\,|\boldsymbol{\Lambda}|\,\boldsymbol{T}^{-1}$$

Optimization Software

- Rapid Optimization Library (ROL)
  - Release within Trilinos in October 2014
  - Rewrite and consolidation of existing tools
    - PEOpt, Aristos, Moocho, Optipack
  - Lead developers: Drew Kouri (primary) and Denis Ridzal
  - Unconstrained optimization
    - Gradient descent, quasi-Newton, nonlinear CG, inexact Newton (FD Hessvecs), with line searches and trust regions.
  - Inequality constraints
    - Box constraints using projected gradient and projected Newton methods

ROL: Methods – FD Hessvecs

$$\min_z J(z)$$

- Goal: Obtain near quadratic convergence from Newton-CG with finite difference Hessian-times-a-vector
  - Set $H(z,v,t) = \left[\nabla J(z+tv) - \nabla J(z)\right]/t$
  - Apply inexact CG to Newton system: $H(z_k, s, t) = -\nabla J(z_k)$
  - At j^{th} iteration of inexact CG require:

    $$\|H(z_k, v_j, t_j) - \nabla^2 J(z_k) v_j\| \leq c_m \varepsilon_k / \|r_{j-1}\|$$

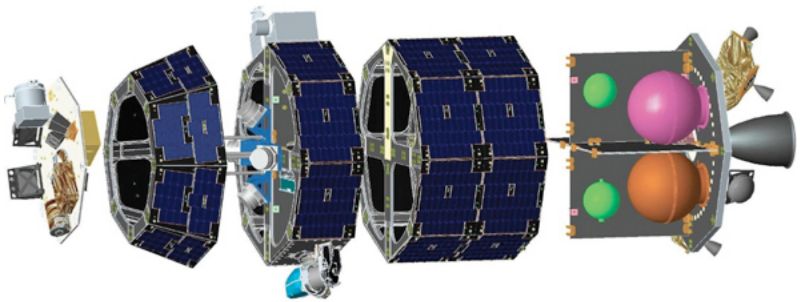
    - Constant depending on max # CG iterations, c_m
    - Previous inexact G residual, ε_k
    - Tolerance governed by inexact Newton's method, r_{j-1}

ROL: Methods – FD Hessvecs

- Convergence
  - Standard assumptions for Newton's
    - If $\nabla^2 J(z)$ is Lipschitz then $\|H(z,v,t) - \nabla^2 J(z)v\| = \mathcal{O}(t)$
  - Error between inexact and true CG residuals is ε_k
    - Simoncini and Szyld 2002
  - Inexact Newton's Methods: Can satisfy Krylov tolerances which ensure superlinear or quadratic convergence.
- Notes
  - FD Hessvecs require an additional gradient computation per CG iteration
  - $t_j = \varepsilon_k / (\|v_j\| \|r_{j-1}\|)$ works well in practice.
  - Use line search to globalize new step.

Mock Satellite

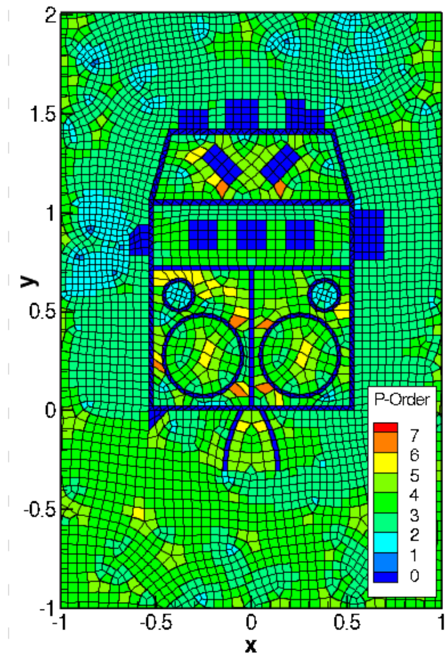
- Modeling structure after
  - NASA Ames Lunar Atmosphere Dust and Environment Explorer (LADEE)
  - Crude 2D representation to demonstrate capabilities



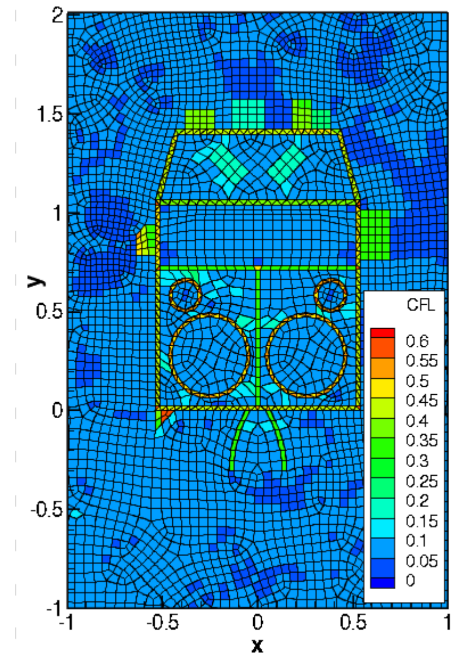
http://machinedesign.com/cad/software-lets-spacecraft-carry-heavier-loads

Mesh

Polynomial Order

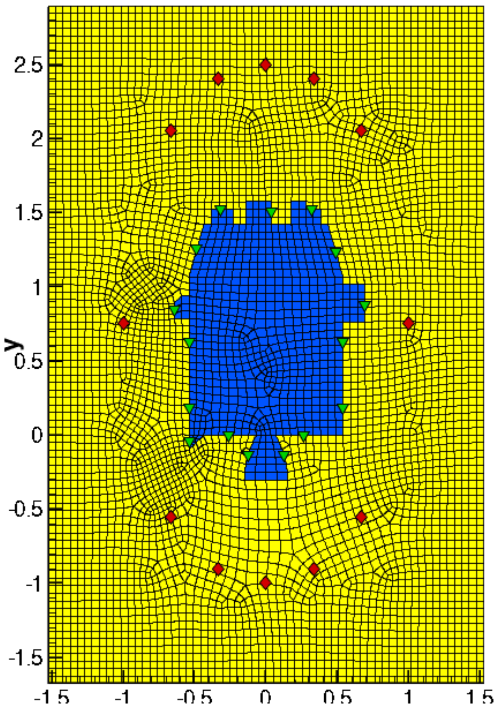


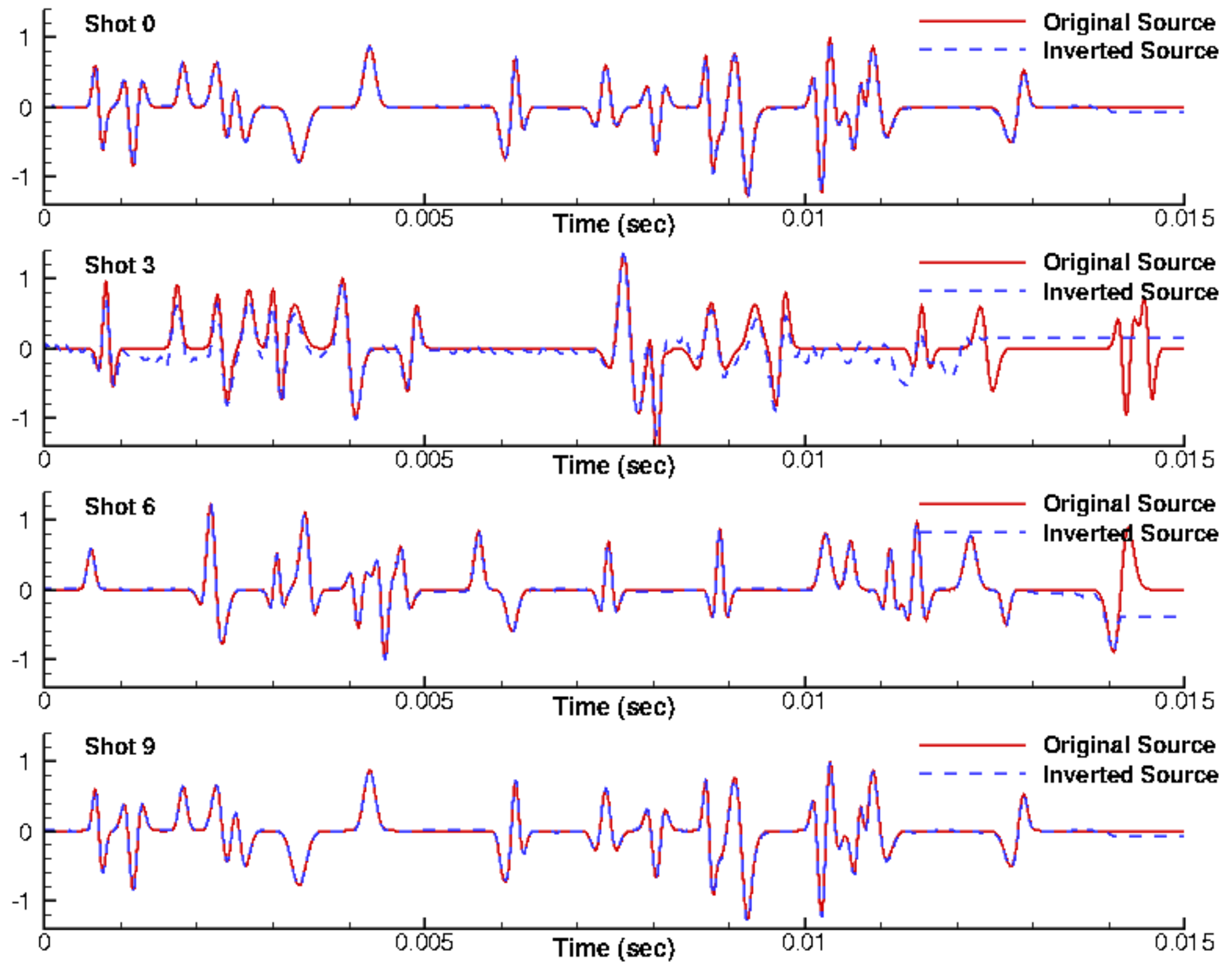
CFL



2D Source Inversion

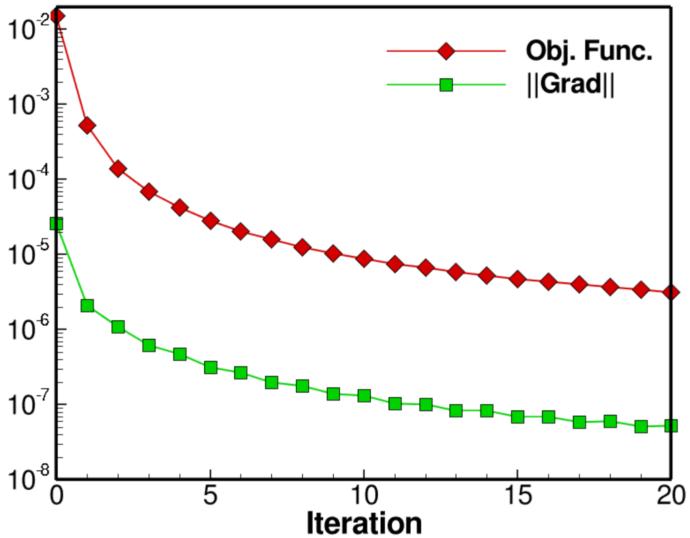
- 12 sources
- Inversion crime
- Supershot

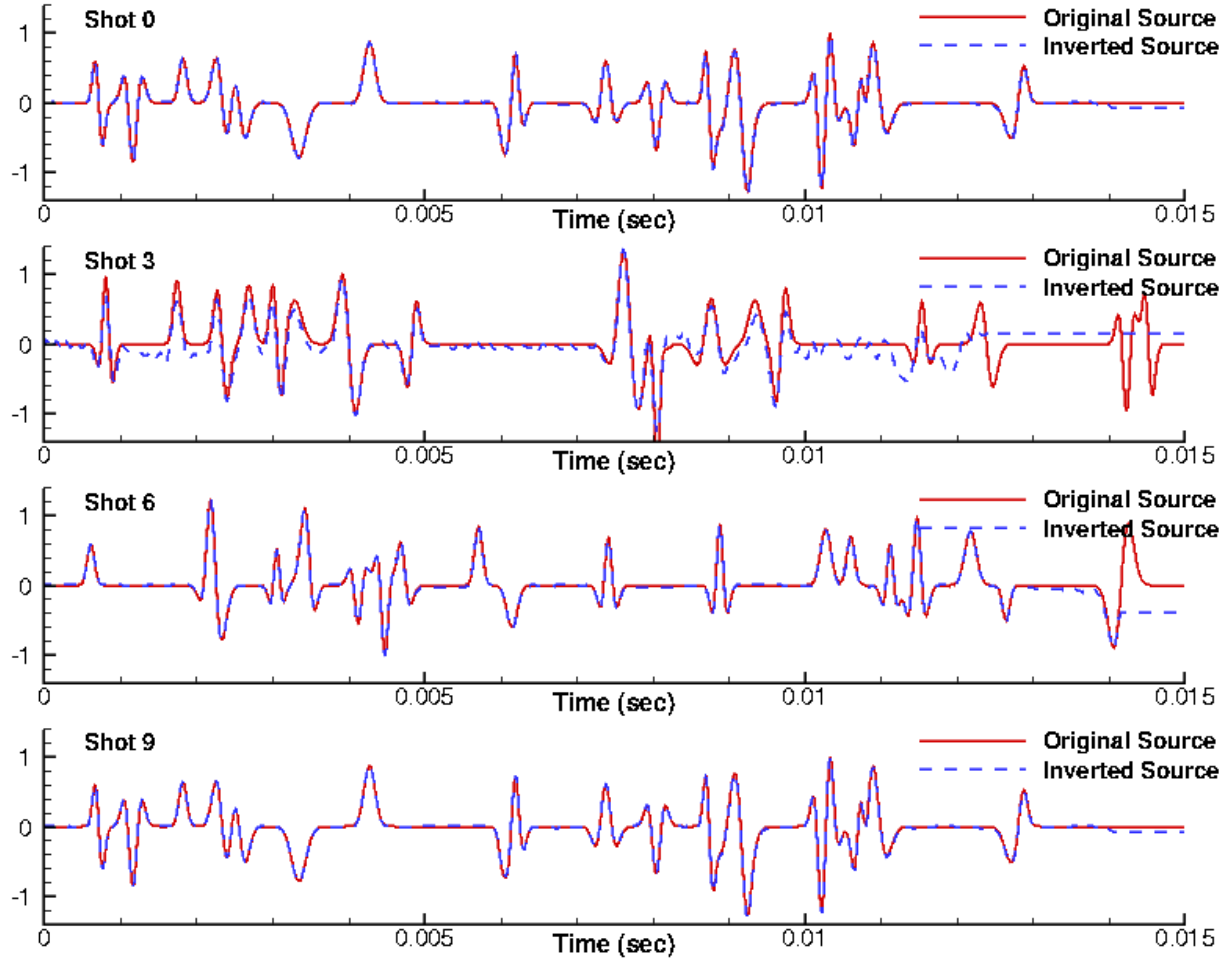




2D Source Inversion

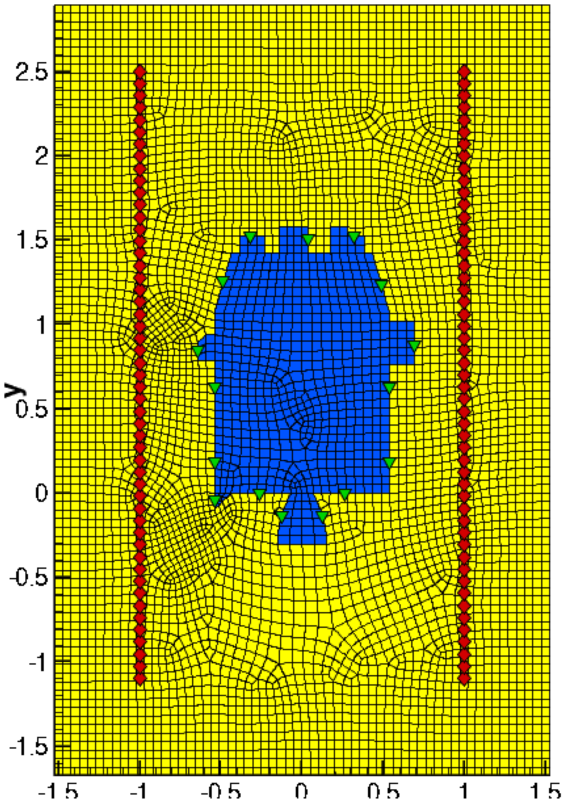
- 12 sources
- Inversion crime
- Supershot

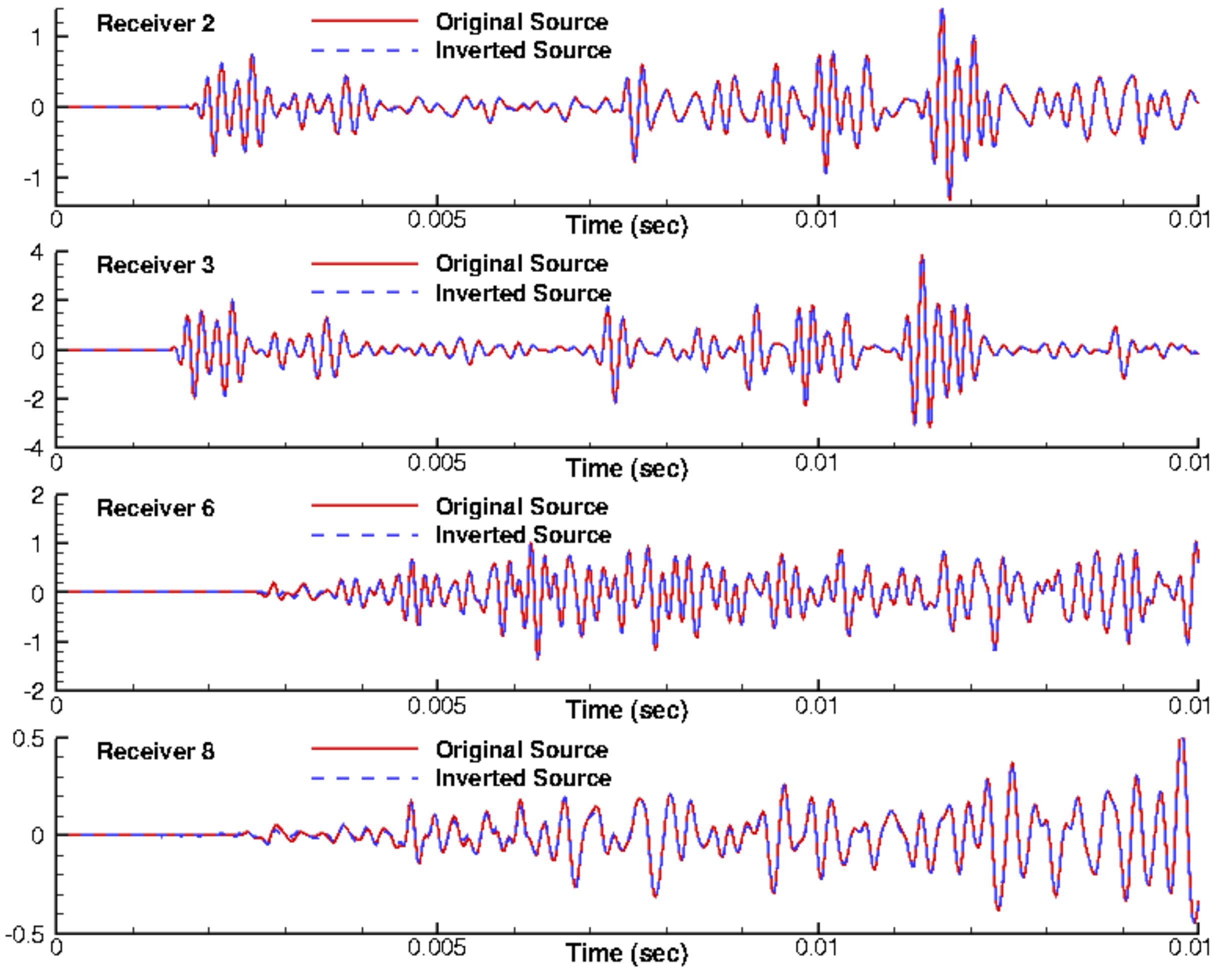




2D Source Inversion

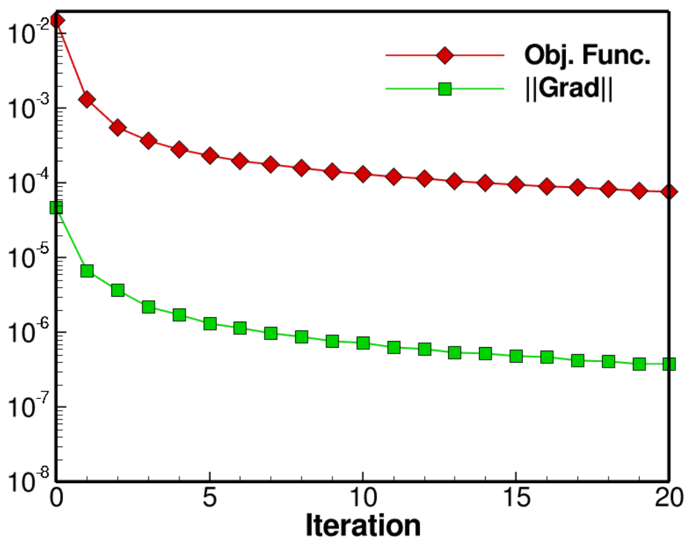
- 102 sources
- Supershot

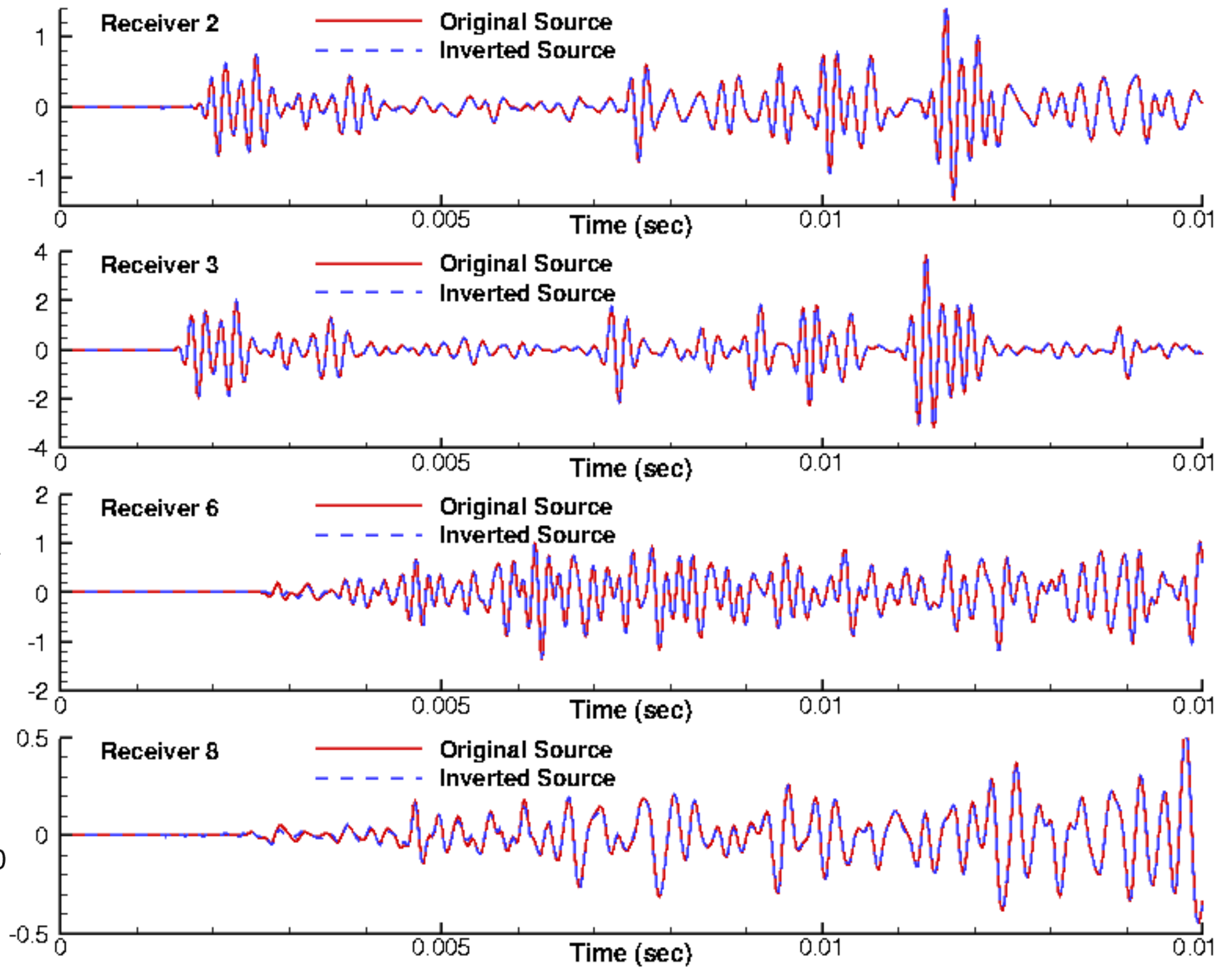




2D Source Inversion

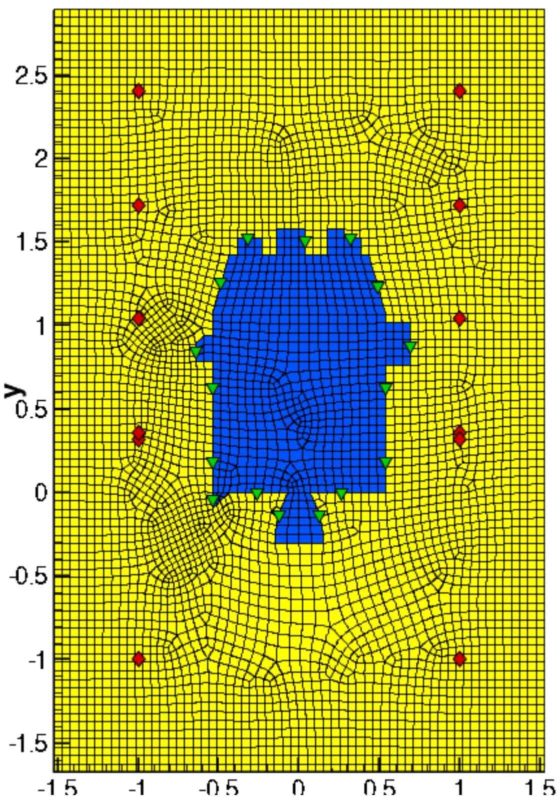
- 102 sources
- Supershot

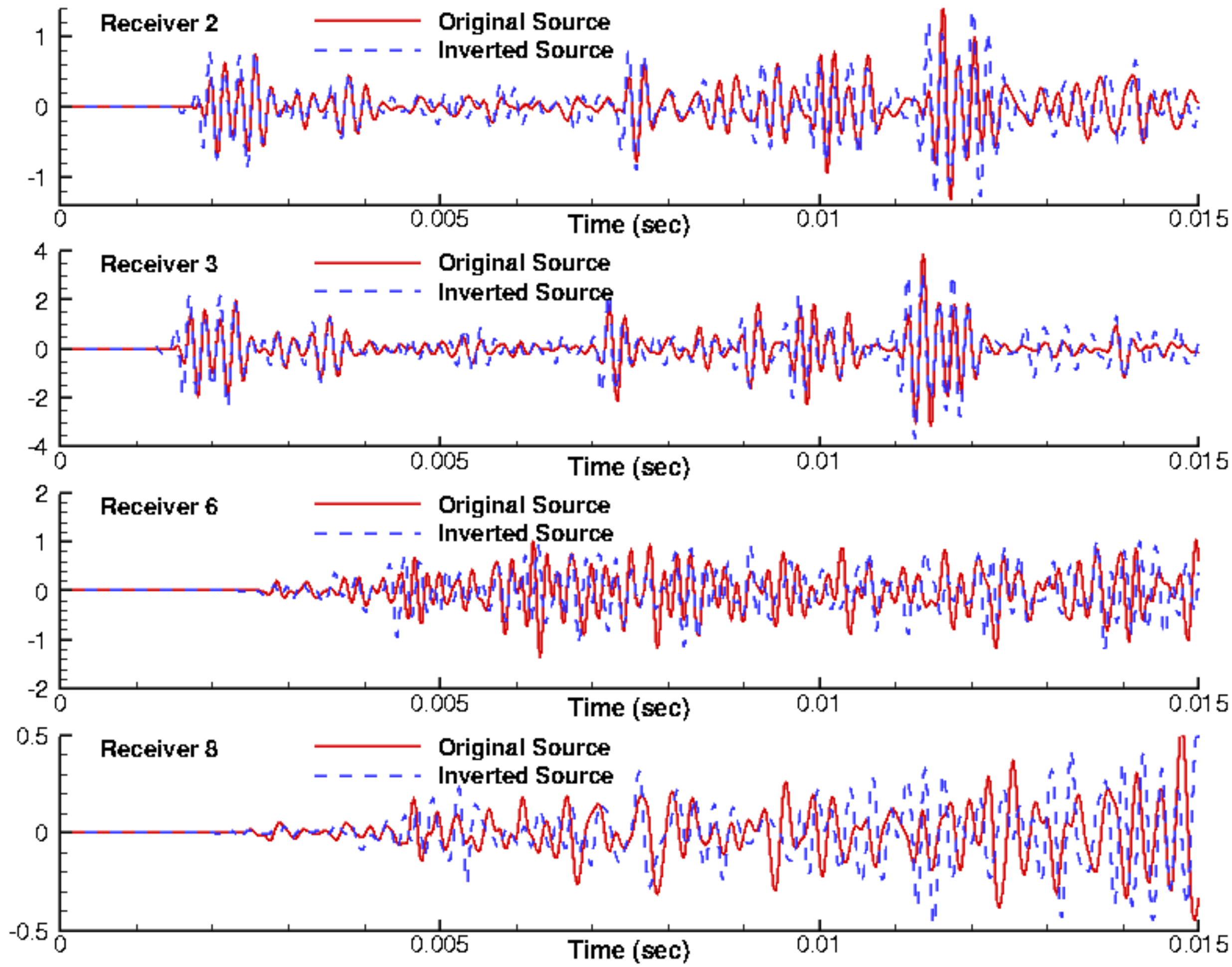




2D Source Inversion

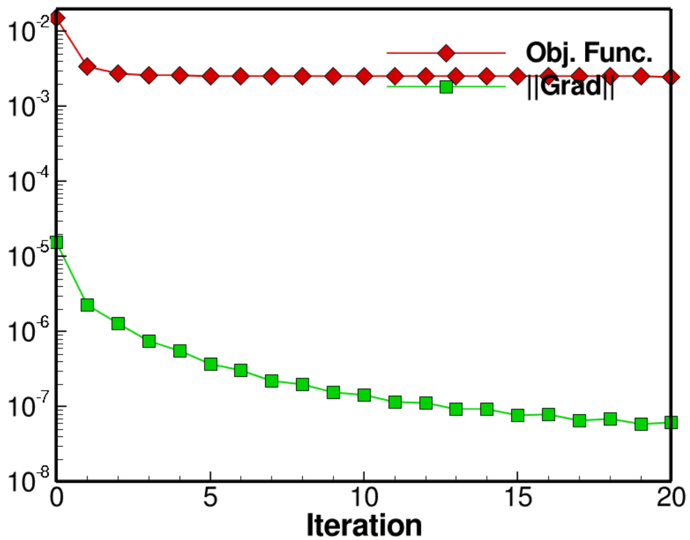
- 12 sources
- Supershot

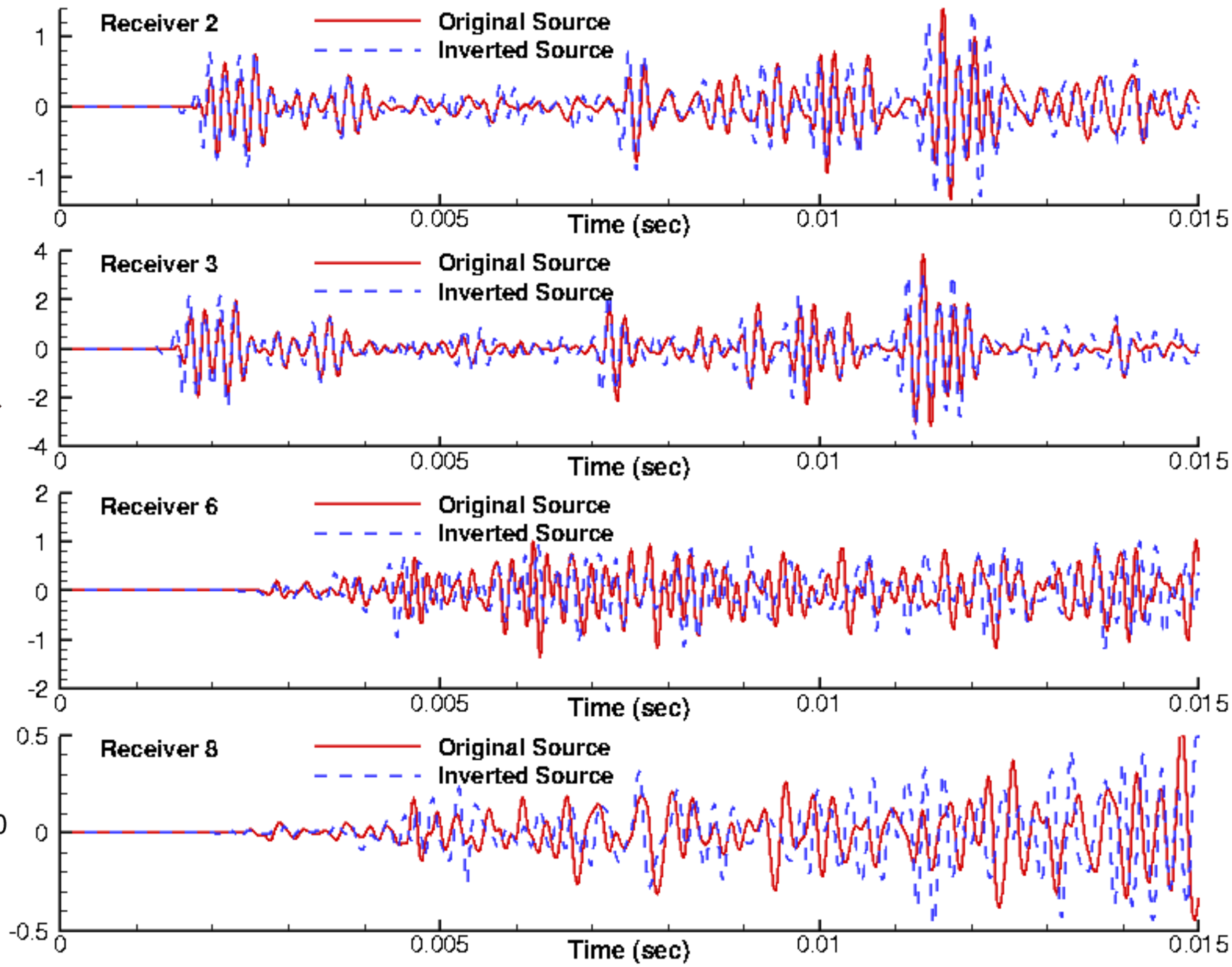




2D Source Inversion

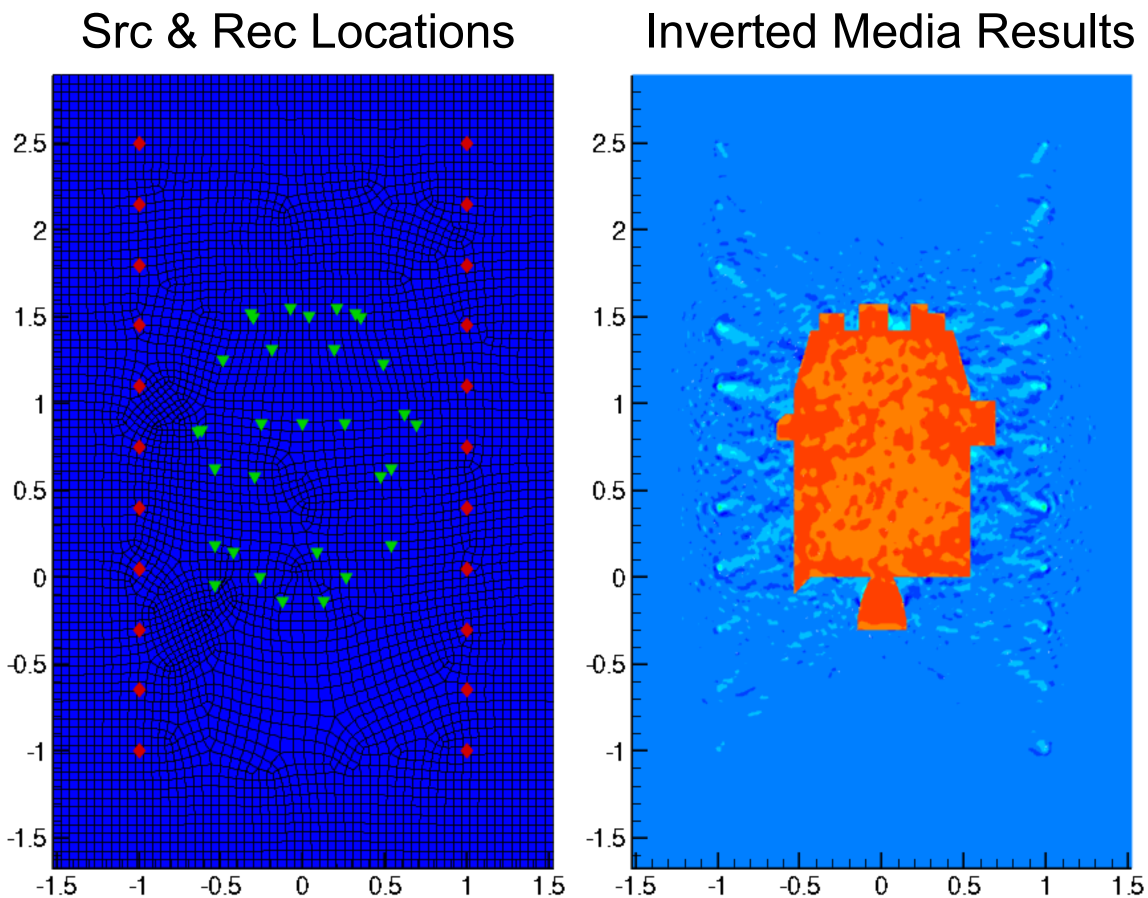
- 12 sources
- Supershot





2D Material Inversion

- 22 individual sources
- 31 receivers
  - 16 on surface
  - 15 on the interior
- Sim time T = 0.015 sec.
- Constant p = 4
- Absorbing BCs
- After 11 iterations



Summary

- Demonstration of DG inversion for non-invasive testing
  - Low dispersion properties allows coarser mesh
  - Variable p-order mesh allows matching order to wavelengths for every element size
  - 2D hybrid mesh using triangles and quadrilaterals
- Source inversion to mimic launch conditions
  - Open question whether there is enough speakers
- Material inversion
  - Initial work promising