INRIA-Pau

A Framework for the Simulation of Multiphysics Problems Based on a hp Fourier-Finite-Element Method

David Pardo, Myung Jin Nam, C. Torres-Verdín

Research Professor at BCAM

Team: D. Pardo, M. J. Nam, V. Calo, L.E. García-Castillo, M. Paszynski, P. Matuszyk, L. Demkowicz, C. Torres-Verdín

February 12th, 2009

overview

1. **Motivation and Objectives: Joint Multiphysics Inversion**
2. **Simulation of Forward Problems**
   - **Parallel Self-Adaptive Goal-Oriented hp-Finite Element Method**
   - **Electromagnetic Applications**
   - **Sonic Applications**
3. **Inversion Library (Work in Progress)**
   - **h-Adaptive Newton's Method**
   - **Implementation**
4. **Conclusions and Future Work**

motivation and objectives

Seismic Measurements

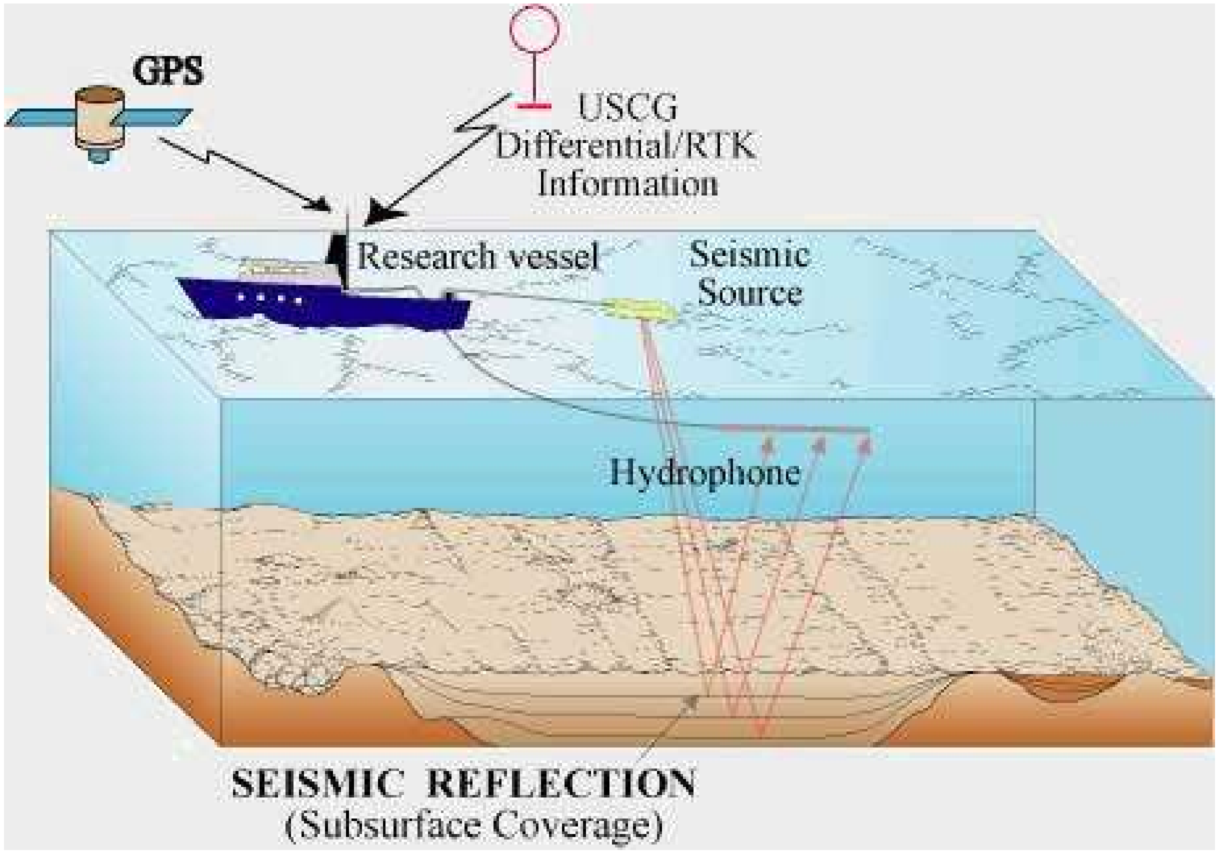


Figure from the USGS Science Center for Coastal and Marine Geology

motivation and objectives

Marine Controlled-Source Electromagnetics (CSEM)

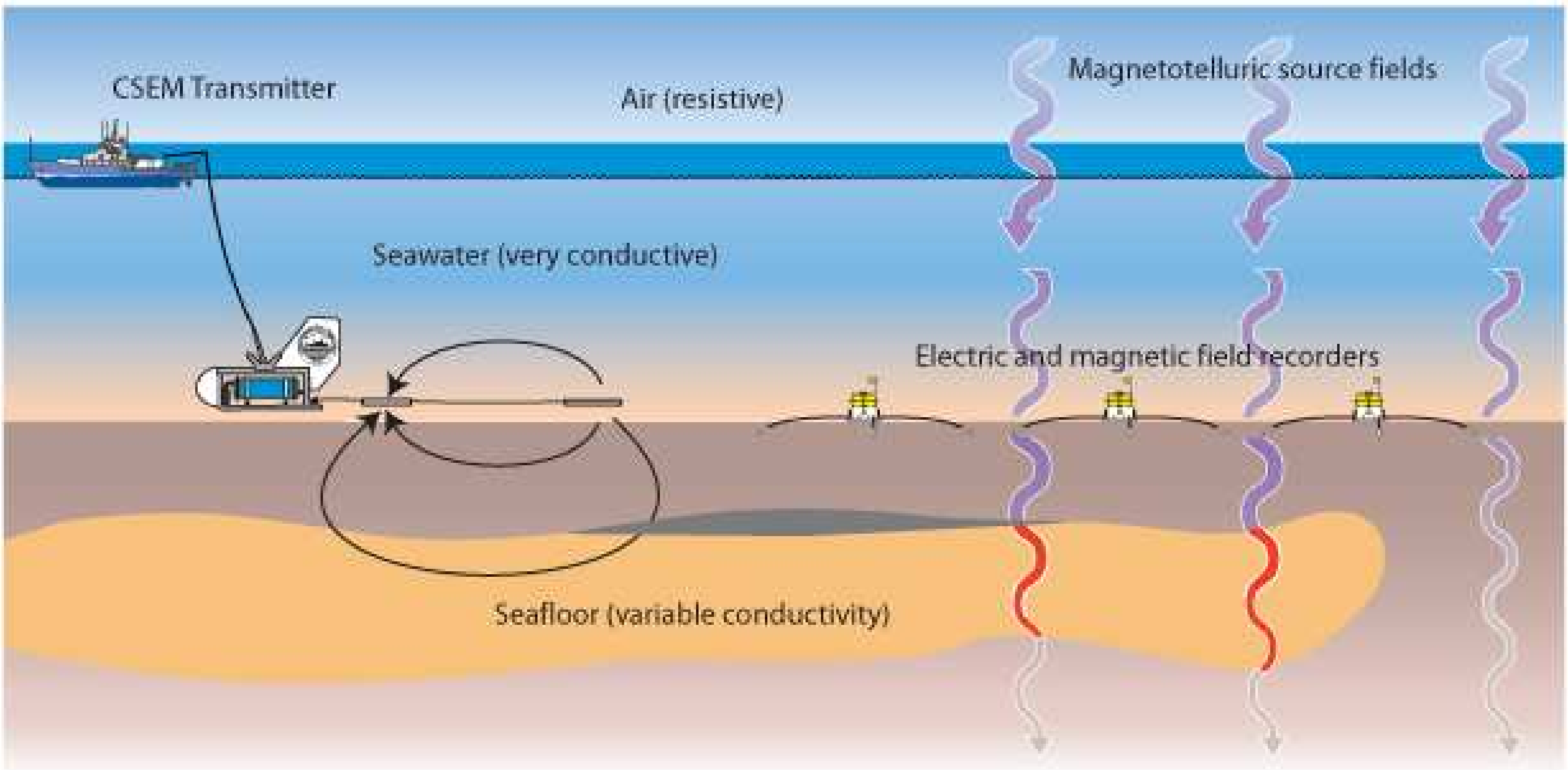
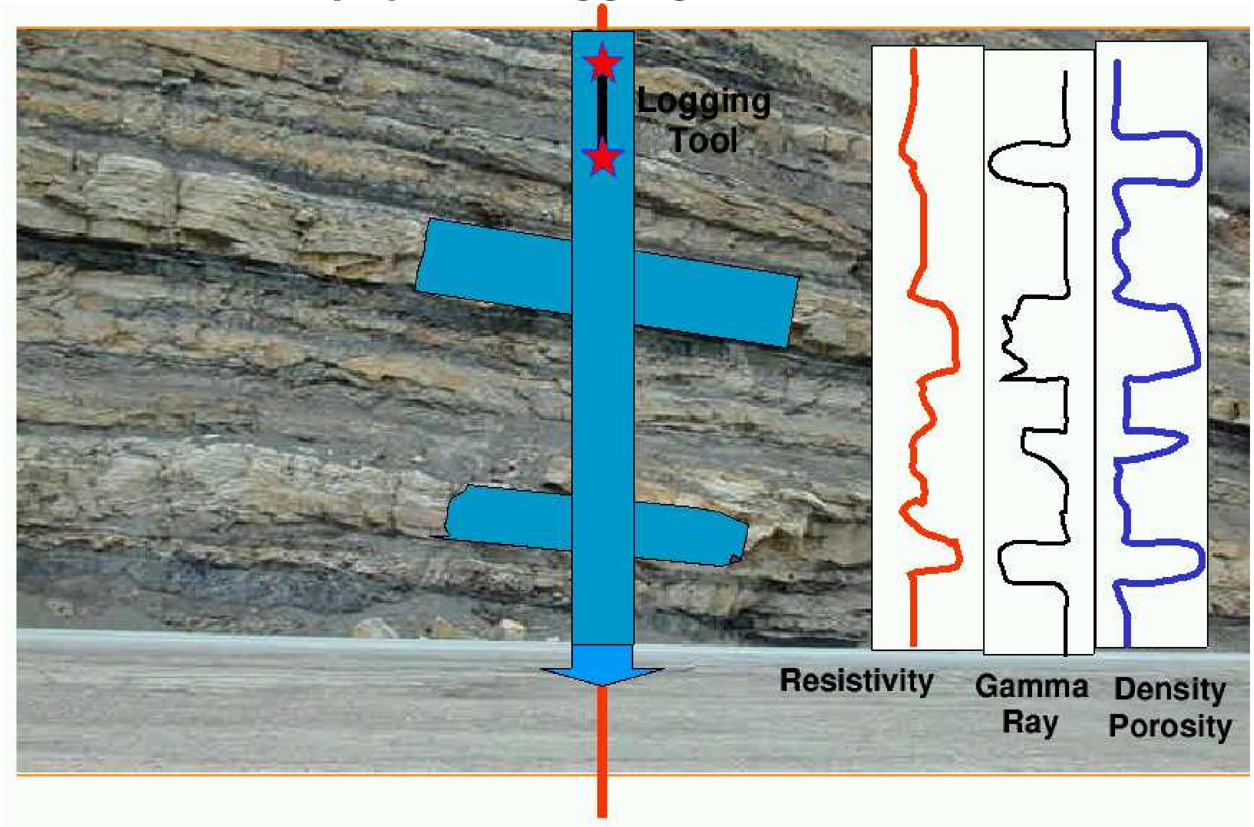


Figure from the UCSD Institute of Oceanography

motivation and objectives

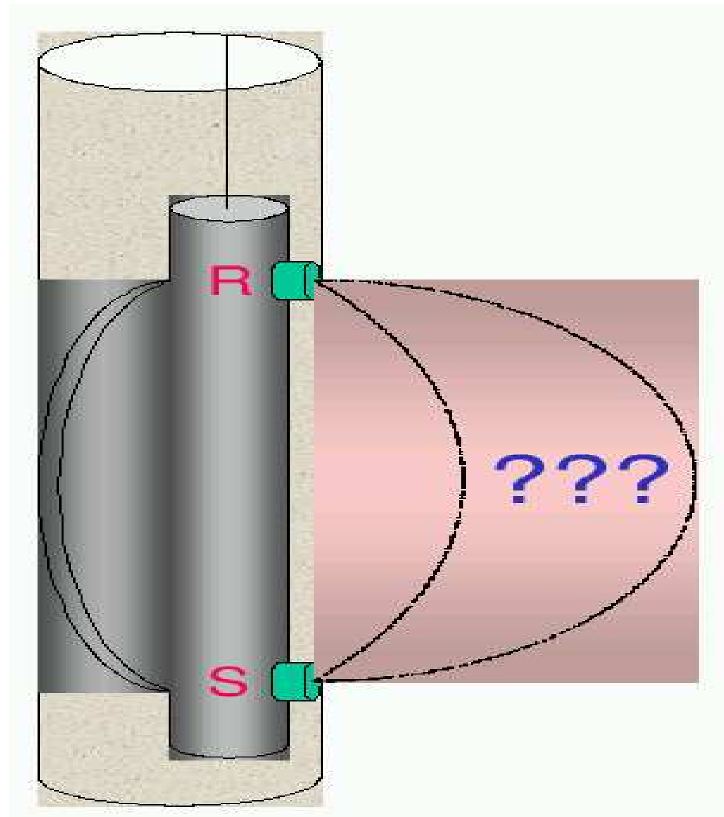
Multiphysics Logging Measurements



OBJECTIVES: To determine payzones (porosity), amount of oil/gas (saturation), and ability to extract oil/gas (permeability).

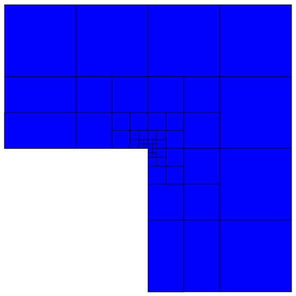
motivation and objectives

Main Objective: To Solve a Multiphysics Inverse Problem



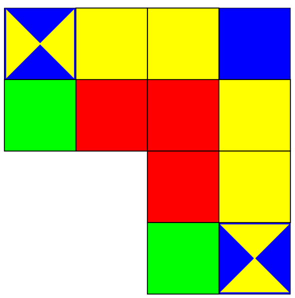
Software to solve the DIRECT problem is essential in order to solve the INVERSE problem.

simulation of forward problems (hp-fem)



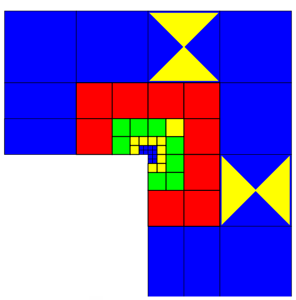
The h-Finite Element Method

1. Convergence limited by the polynomial degree, and large material contrasts.
2. Optimal h-grids do NOT converge exponentially in real applications.
3. They may "lock" (100$\%$ error).



The p-Finite Element Method

1. Exponential convergence feasible for analytical ("nice") solutions.
2. Optimal *p*-grids do NOT converge exponentially in real applications.
3. If initial *h*-grid is not adequate, the *p*-method will fail miserably.



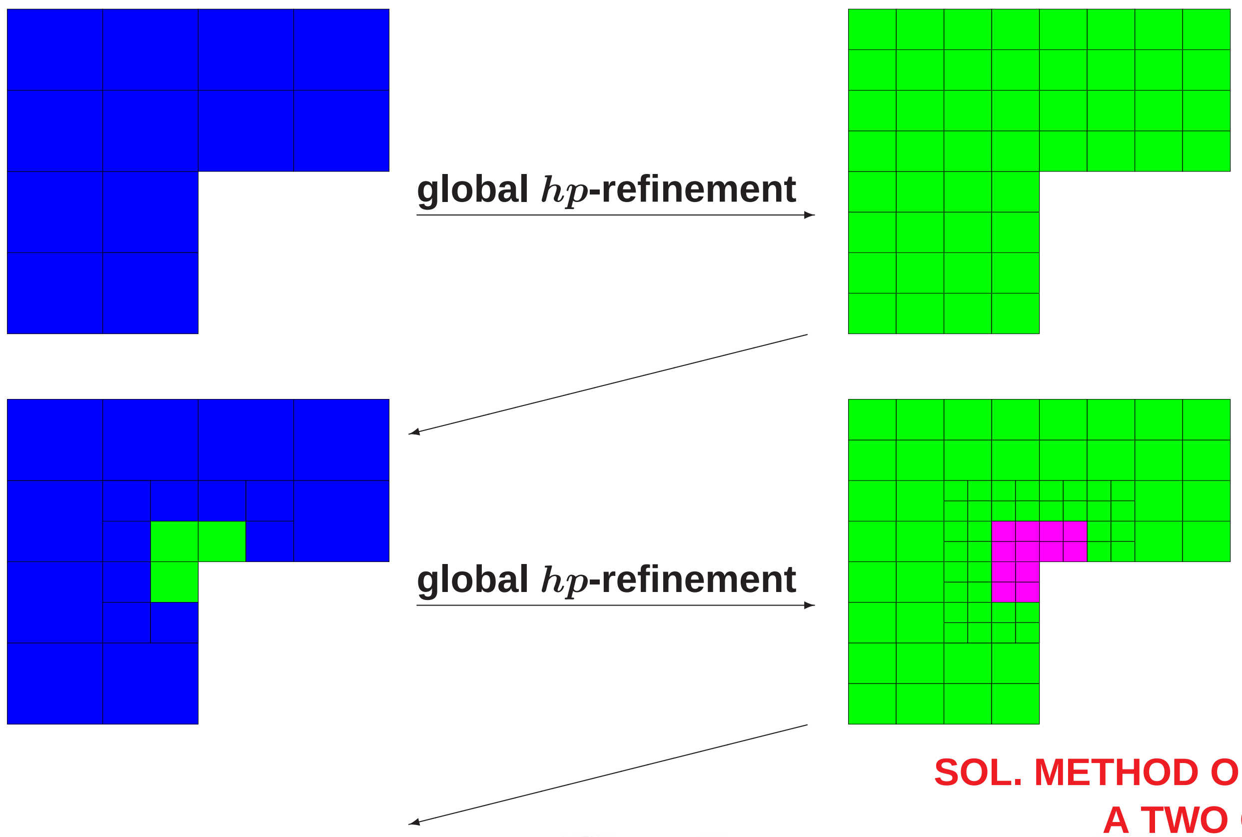
The hp-Finite Element Method

1. Exponential convergence feasible for ALL solutions.
2. Optimal *hp*-grids DO converge exponentially in real applications.
3. If initial *hp*-grid is not adequate, results will still be great.

simulation of forward problems (hp-fem)

Energy norm based fully automatic *hp*-adaptive strategy

Coarse grids (hp) **Fine grids** $(h/2, p+1)$

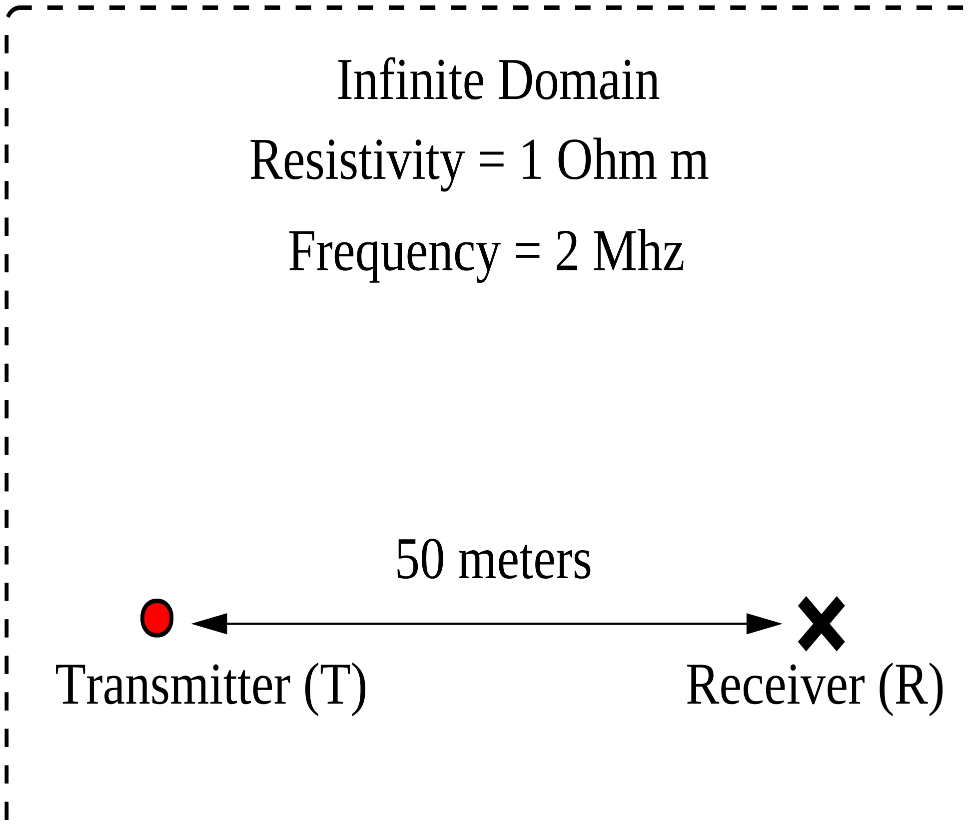


SOL. METHOD ON FINE GRIDS:
A TWO GRID SOLVER

simulation of forward problems (hp-fem)

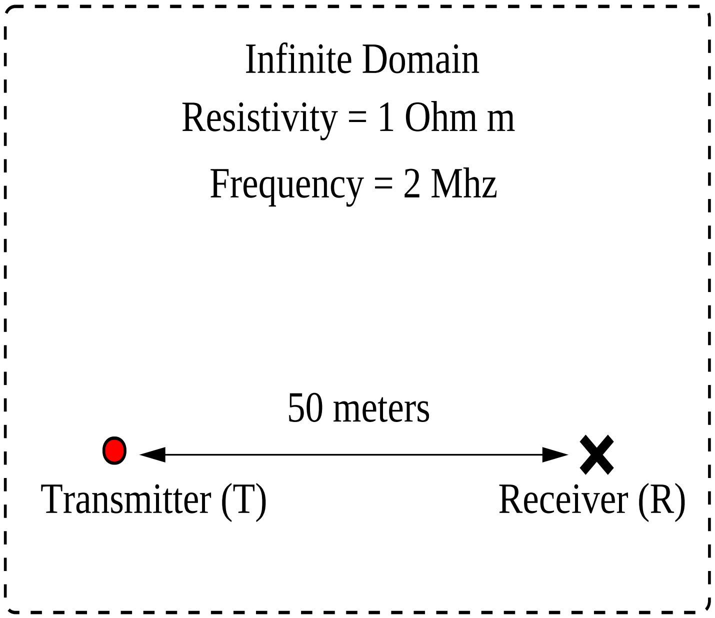
Motivation (Goal-Oriented Adaptivity)

Test Problem



simulation of forward problems (hp-fem)

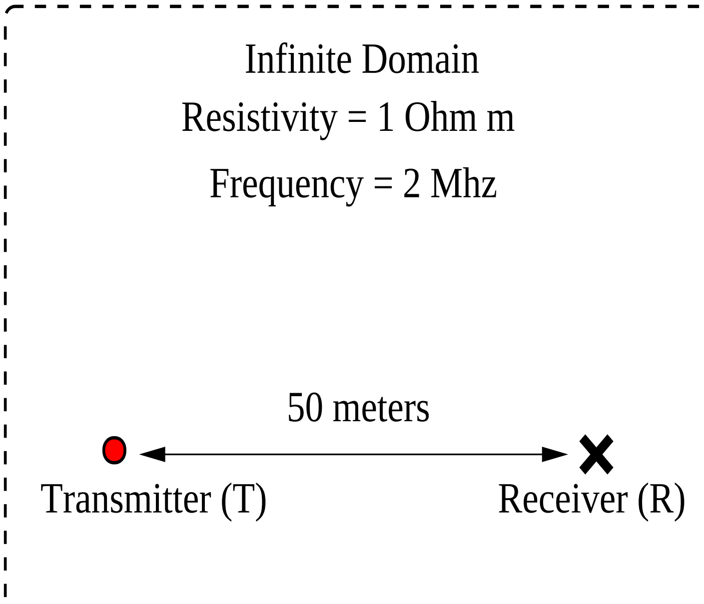
Motivation (Goal-Oriented Adaptivity)



- **Solution decays exponentially.**
- $\dfrac{|E(T)|}{|E(R)|} \approx 10^{60}$
- **Results using energy-norm adaptivity:**
  - **Energy-norm error:** 0.001%
  - **Relative error in the quantity of interest** $> 10^{30}$ %.

simulation of forward problems (hp-fem)

Motivation (Goal-Oriented Adaptivity)

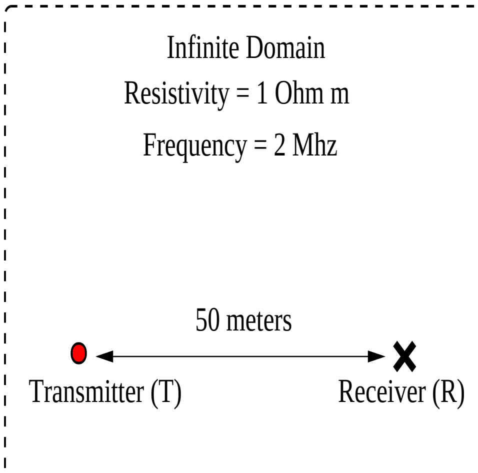


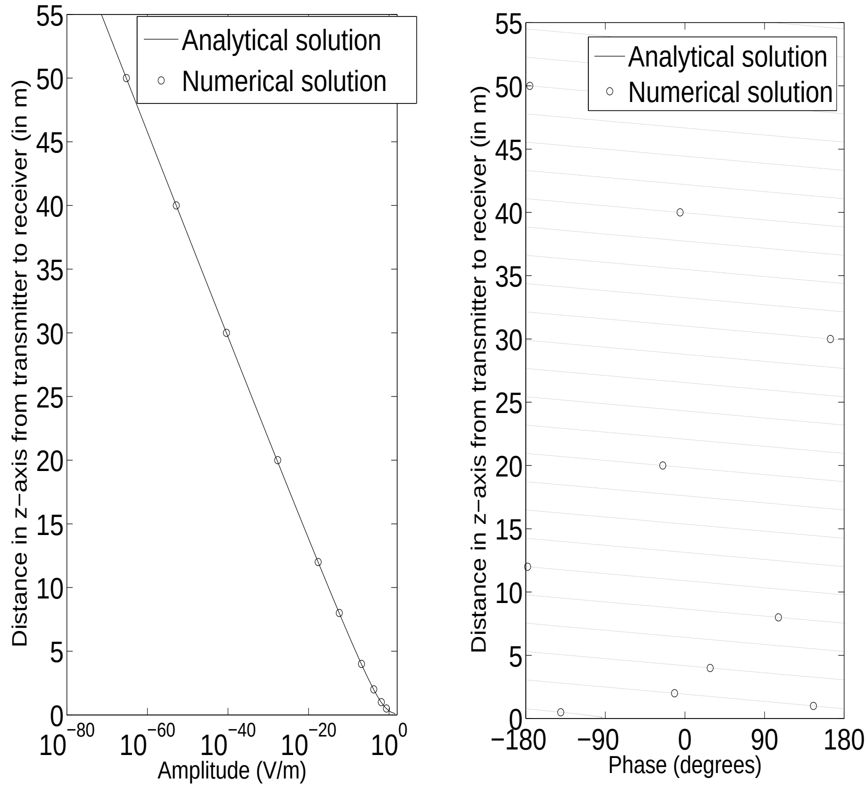
- **Solution decays exponentially.**
- $\frac{|E(T)|}{|E(R)|} \approx 10^{60}$
- **Results using energy-norm adaptivity:**
  - **Energy-norm error:** 0.001%
  - **Relative error in the quantity of interest** $> 10^{30}\ \%$.

Goal-oriented adaptivity is needed. **Becker-Rannacher (1995,1996), Rannacher-Stuttmeier (1997), Cirak-Ramm (1998), Paraschivoiu-Patera (1998), Peraire-Patera (1998), Prudhomme-Oden (1999, 2001), Heuveline-Rannacher (2003), Solin-Demkowicz (2004).**

simulation of forward problems (hp-fem)

Motivation (Goal-Oriented Adaptivity)





Goal-oriented adaptivity is needed

simulation of forward problems (hp-fem)

Mathematical Formulation (Goal-Oriented Adaptivity)

We consider the following problem (in variational form):

$$\begin{cases} \text{Find } L(\Psi) \text{, where } \Psi \in V \text{ such that :} \\ b(\Psi, \xi) = f(\xi) \quad \forall \xi \in V \ . \end{cases}$$

We define residual $r_e(\xi) = b(e, \xi)$. We seek for solution G of:

$$\begin{cases} \text{Find } G \in V'' \sim V \text{ such that :} \\ G(r_e) = L(e) \ . \end{cases}$$

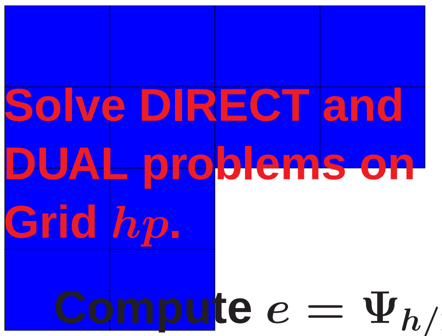
This is necessarily solved if we find the solution of the *dual* problem:

$$\begin{cases} \text{Find } G \in V \text{ such that :} \\ b(\Psi, G) = L(\Psi) \quad \forall \Psi \in V \ . \end{cases}$$

Notice that $L(e) = b(e, G)$.

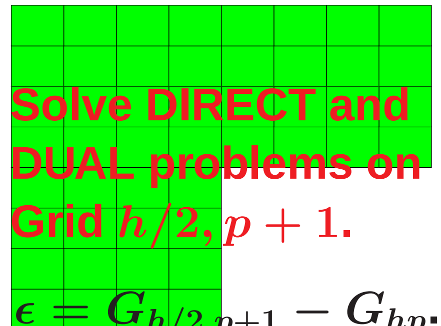
simulation of forward problems (hp-fem)

Algorithm for Goal-Oriented Adaptivity



Solve DIRECT and DUAL problems on Grid hp.

⟶

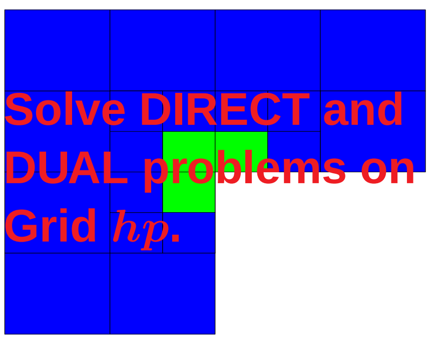


Solve DIRECT and DUAL problems on Grid $h/2, p+1$.

Compute $e = \Psi_{h/2,p+1} - \Psi_{hp}$, and $\epsilon = G_{h/2,p+1} - G_{hp}$.

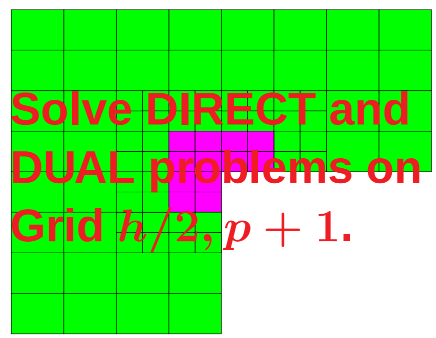
Represent the error as: $|L(e)| = |b(e,\epsilon)| \leq \sum_K |b_K(e,\epsilon)|$.

Apply the fully automatic hp-adaptive algorithm.



Solve DIRECT and DUAL problems on Grid hp.

⟶

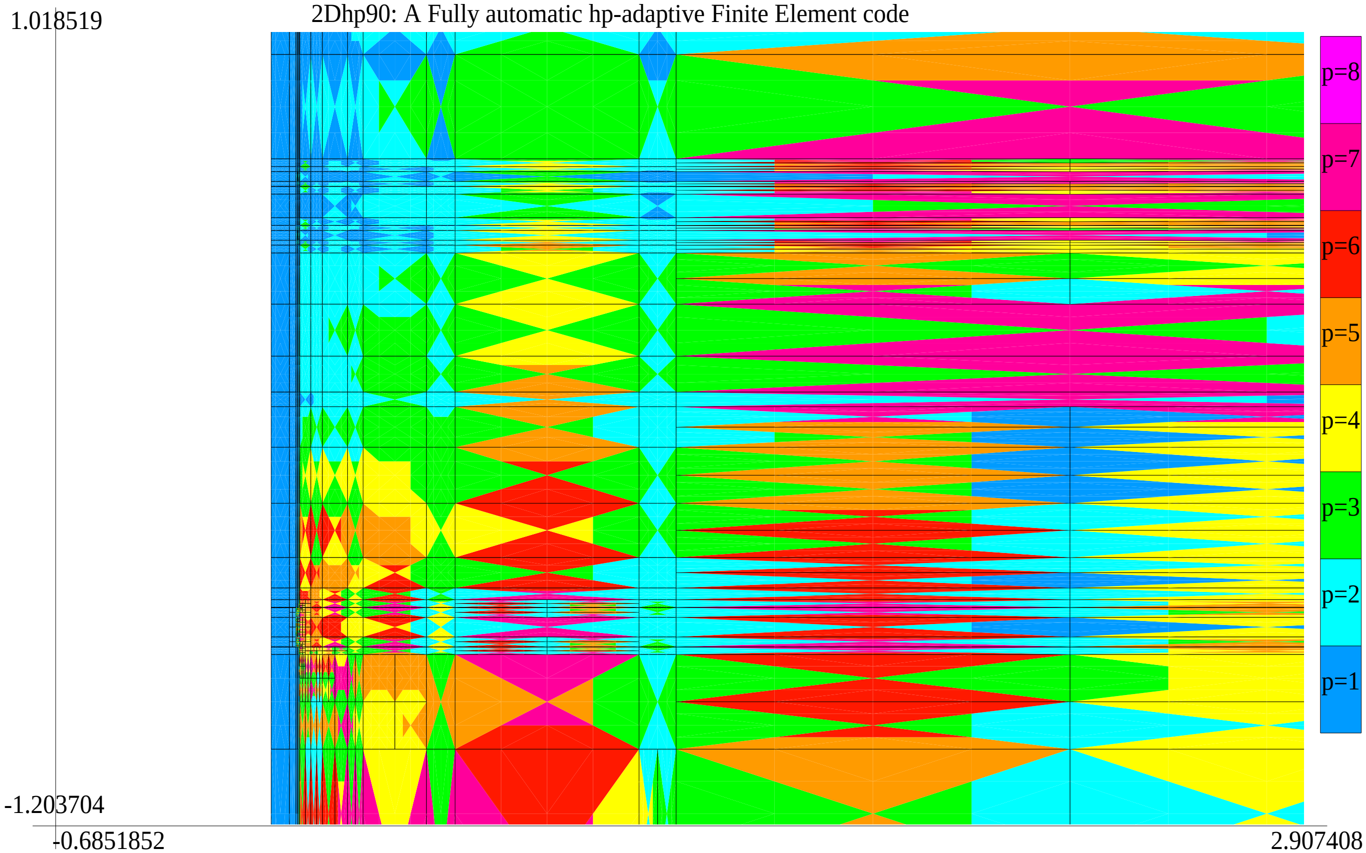


Solve DIRECT and DUAL problems on Grid $h/2, p+1$.

simulation of forward problems (hp-fem)

Axisymmetric Logging-While-Drilling (LWD) Simulation

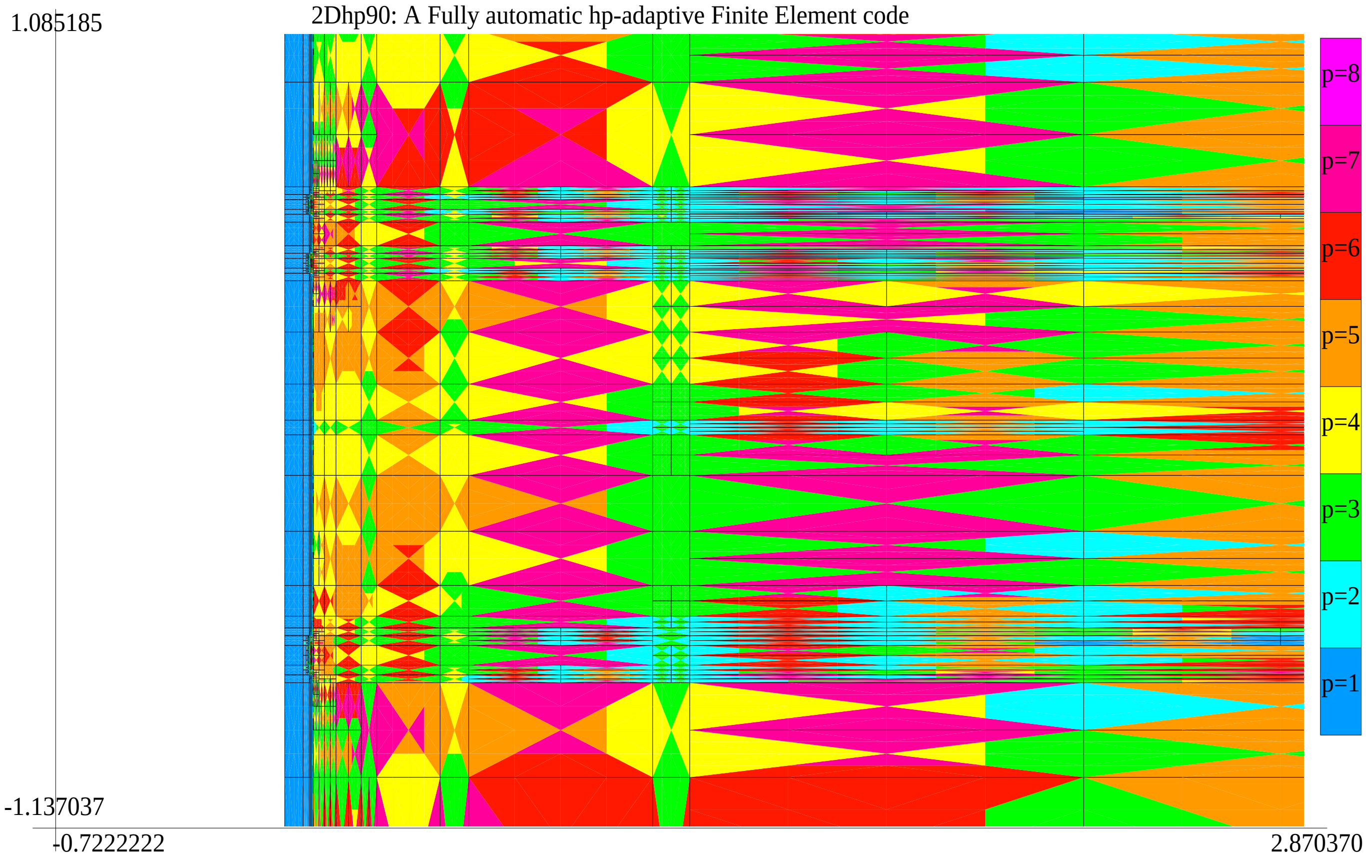
ENERGY-NORM HP-ADAPTIVITY



simulation of forward problems (hp-fem)

Axisymmetric Logging-While-Drilling (LWD) Simulation

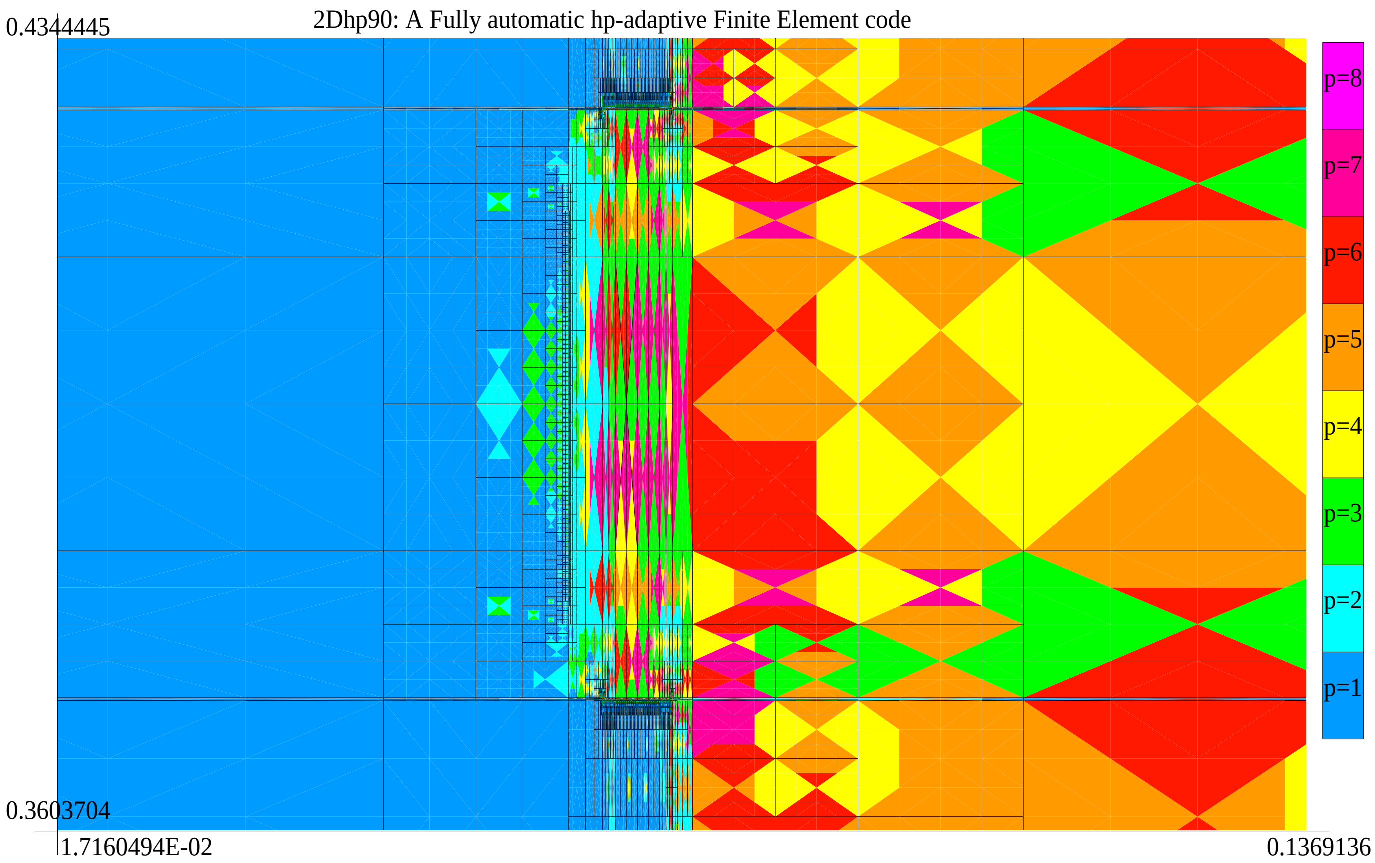
GOAL-ORIENTED HP-ADAPTIVITY (Quadrilateral Elements)



simulation of forward problems (hp-fem)

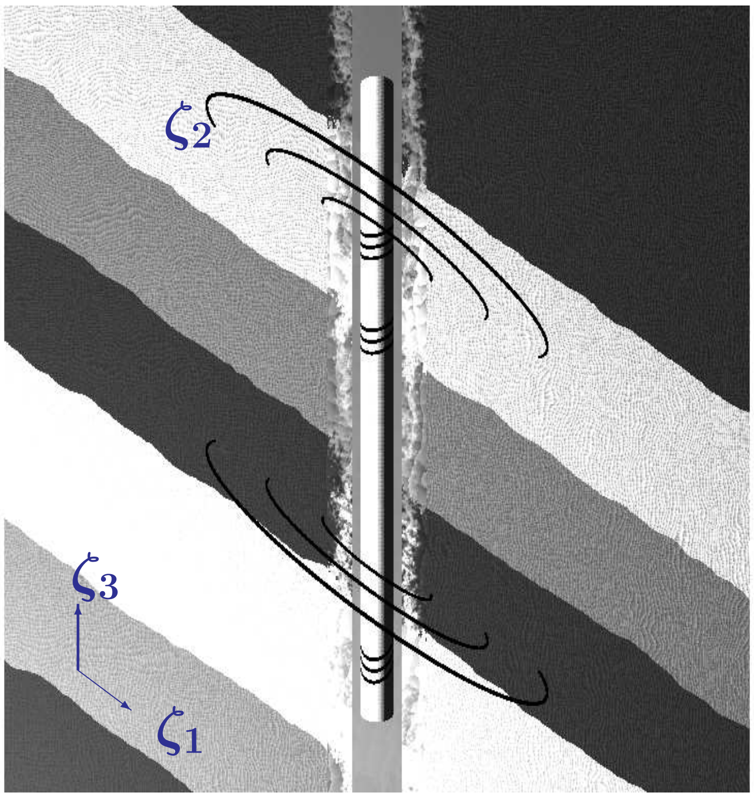
Axisymmetric Logging-While-Drilling (LWD) Simulation

GOAL-ORIENTED HP-ADAPTIVITY (Zoom towards first receiver)



simulation of forward problems (hp-fem)

Non-Orthogonal System of Coordinates



Fourier Series Expansion in ζ_2

DC Problems: $-\nabla \sigma \nabla u = f$

$$u(\zeta_1, \zeta_2, \zeta_3) = \sum_{l=-\infty}^{l=\infty} u_l(\zeta_1, \zeta_3) e^{jl\zeta_2}$$

$$\sigma(\zeta_1, \zeta_2, \zeta_3) = \sum_{m=-\infty}^{m=\infty} \sigma_m(\zeta_1, \zeta_3) e^{jm\zeta_2}$$

$$f(\zeta_1, \zeta_2, \zeta_3) = \sum_{n=-\infty}^{n=\infty} f_n(\zeta_1, \zeta_3) e^{jn\zeta_2}$$

Fourier modes $e^{jl\zeta_2}$ are orthogonal high-order basis functions that are (almost) invariant with respect to the gradient operator.

simulation of forward problems (hp-fem)

De Rham diagram

De Rham diagram is critical to the theory of FE discretizations of multi-physics problems.

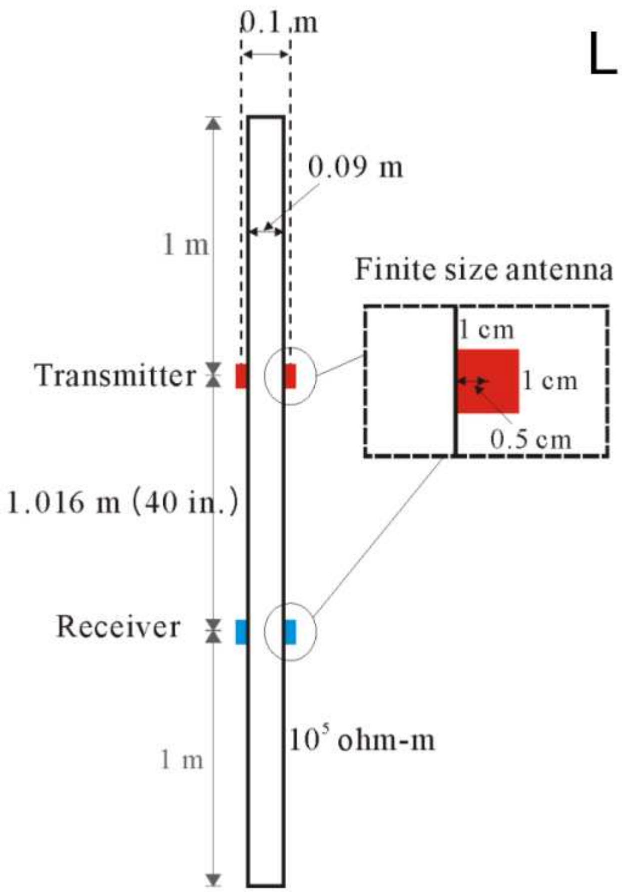
$$
\begin{array}{ccccccccccc}
\mathbb{R} & \longrightarrow & W & \xrightarrow{\nabla} & Q & \xrightarrow{\nabla\times} & V & \xrightarrow{\nabla\circ} & L^2 & \longrightarrow & 0 \\
\downarrow id & & \downarrow \Pi & & \downarrow \Pi^{\mathrm{curl}} & & \downarrow \Pi^{\mathrm{div}} & & \downarrow P & & \\
\mathbb{R} & \longrightarrow & W^{\mathrm{p}} & \xrightarrow{\nabla} & Q^{\mathrm{p}} & \xrightarrow{\nabla\times} & V^{\mathrm{p}} & \xrightarrow{\nabla\circ} & W^{\mathrm{p}-1} & \longrightarrow & 0\,.
\end{array}
$$

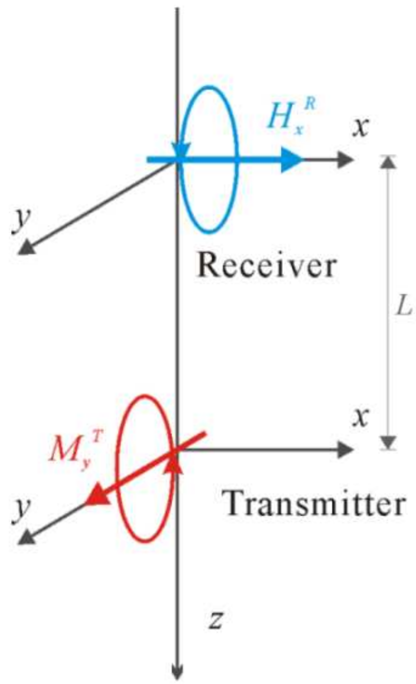
This diagram relates two exact sequences of spaces, on both continuous and discrete levels, and corresponding interpolation operators.

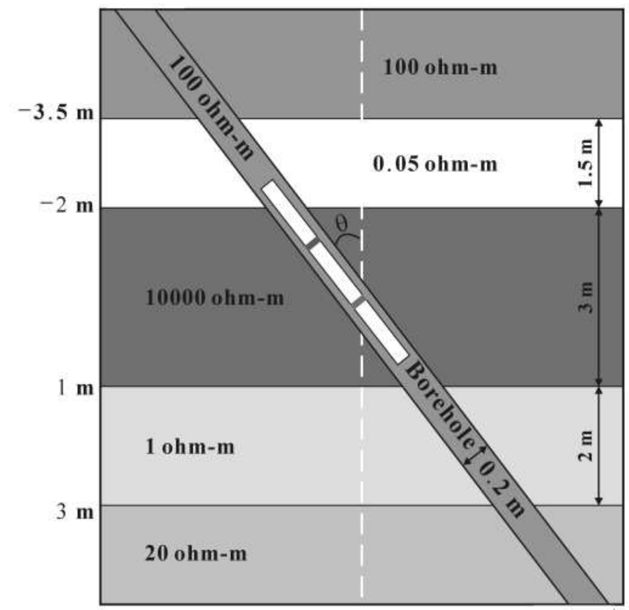
simulation of forward problems (hp-fem)

Tri-Axial Induction Tool

L=40 in, Frequency=20 kHz

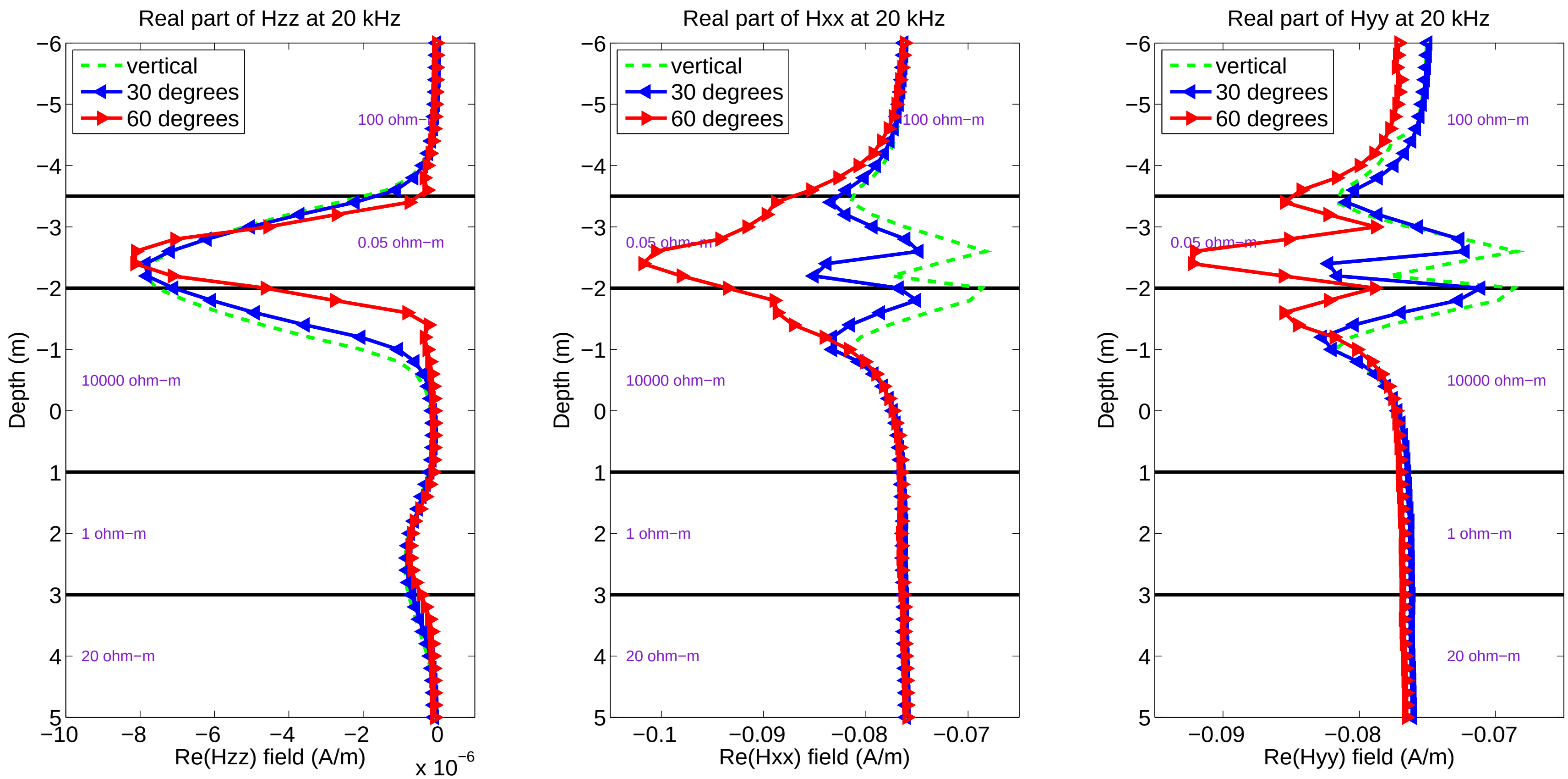






simulation of forward problems (hp-fem)

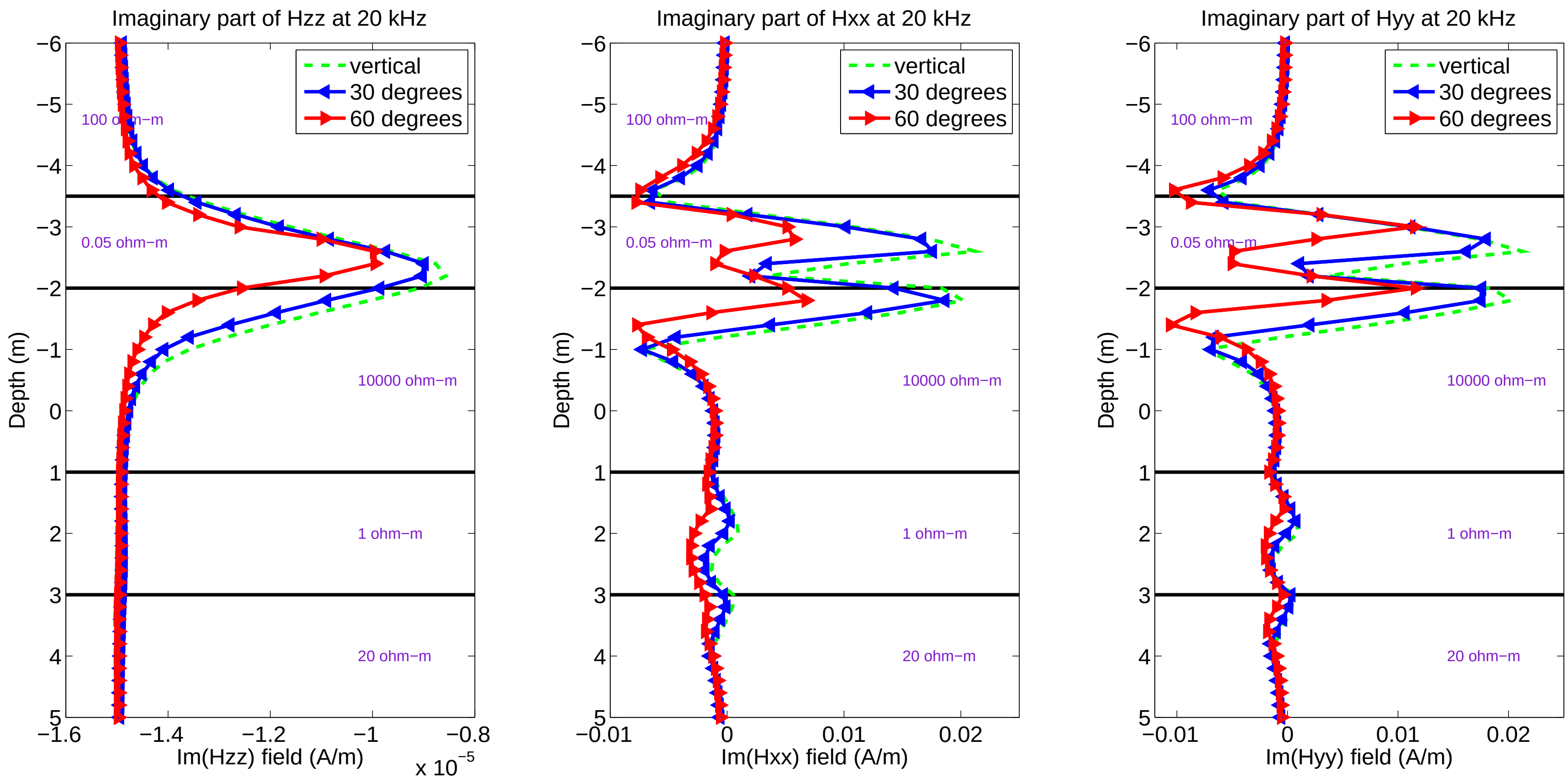
Tri-Axial Induction Tools in Deviated Wells (0, 30, and 60 degrees)



Triaxial tools are more sensitive to dip angle effects

simulation of forward problems (hp-fem)

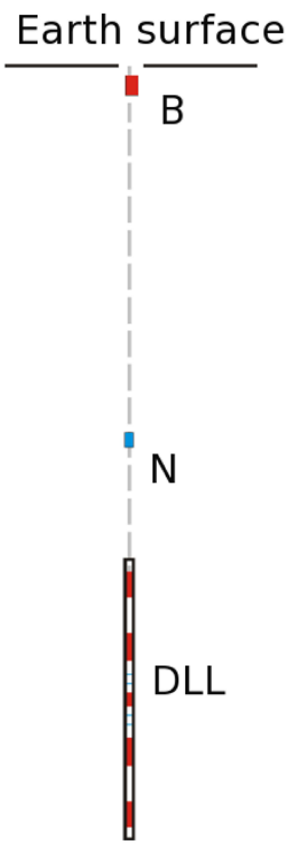
Tri-Axial Induction Tools in Deviated Wells (0, 30, and 60 degrees)

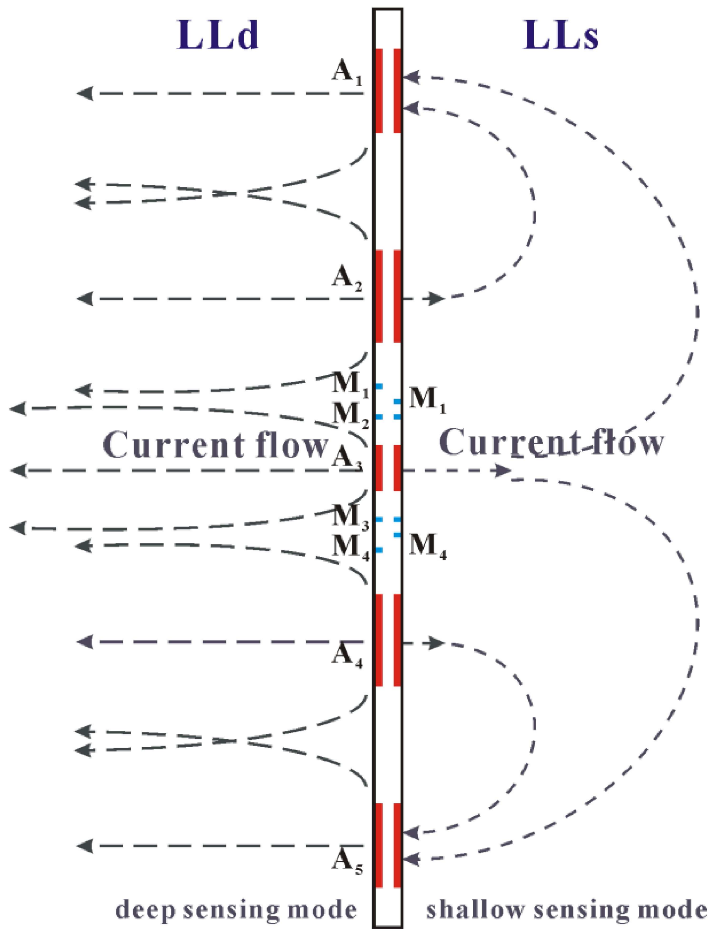


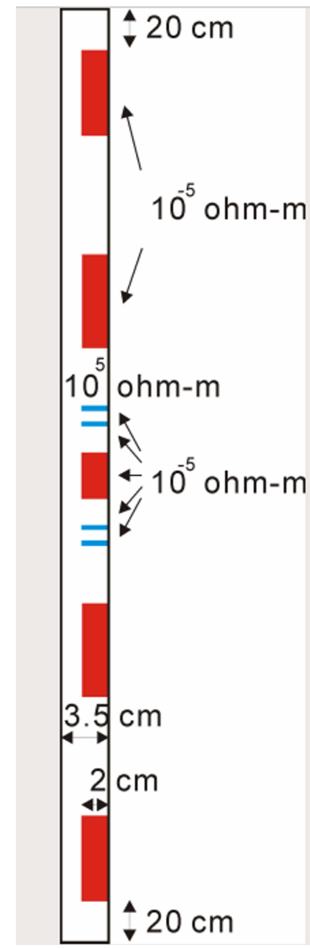
Triaxial tools are more sensitive to dip angle effects

simulation of forward problems (hp-fem)

DLL Tool

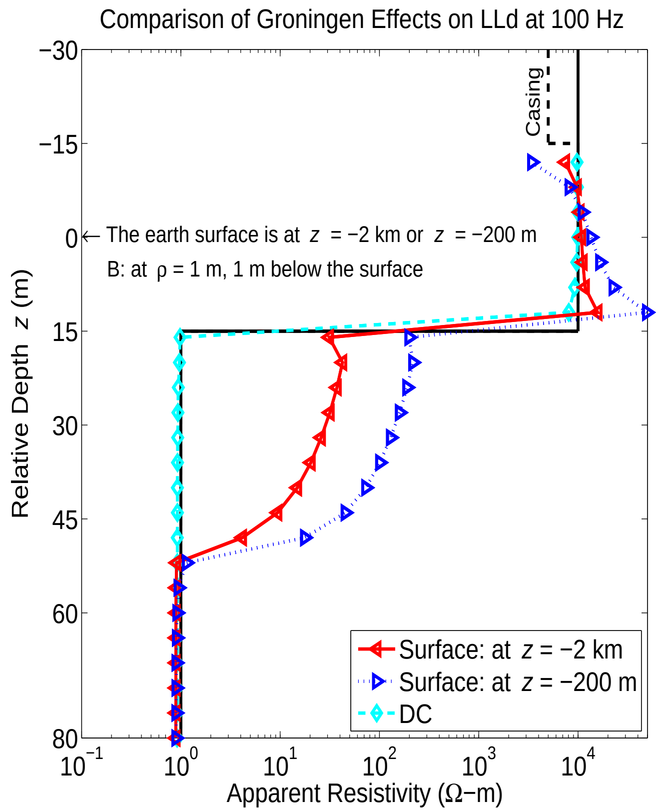


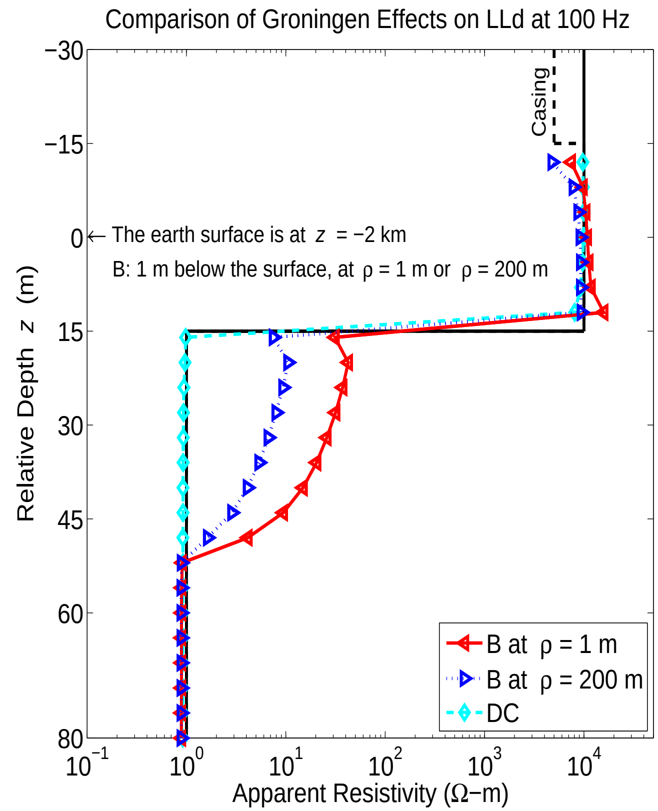




simulation of forward problems (hp-fem)

Groningen Effect

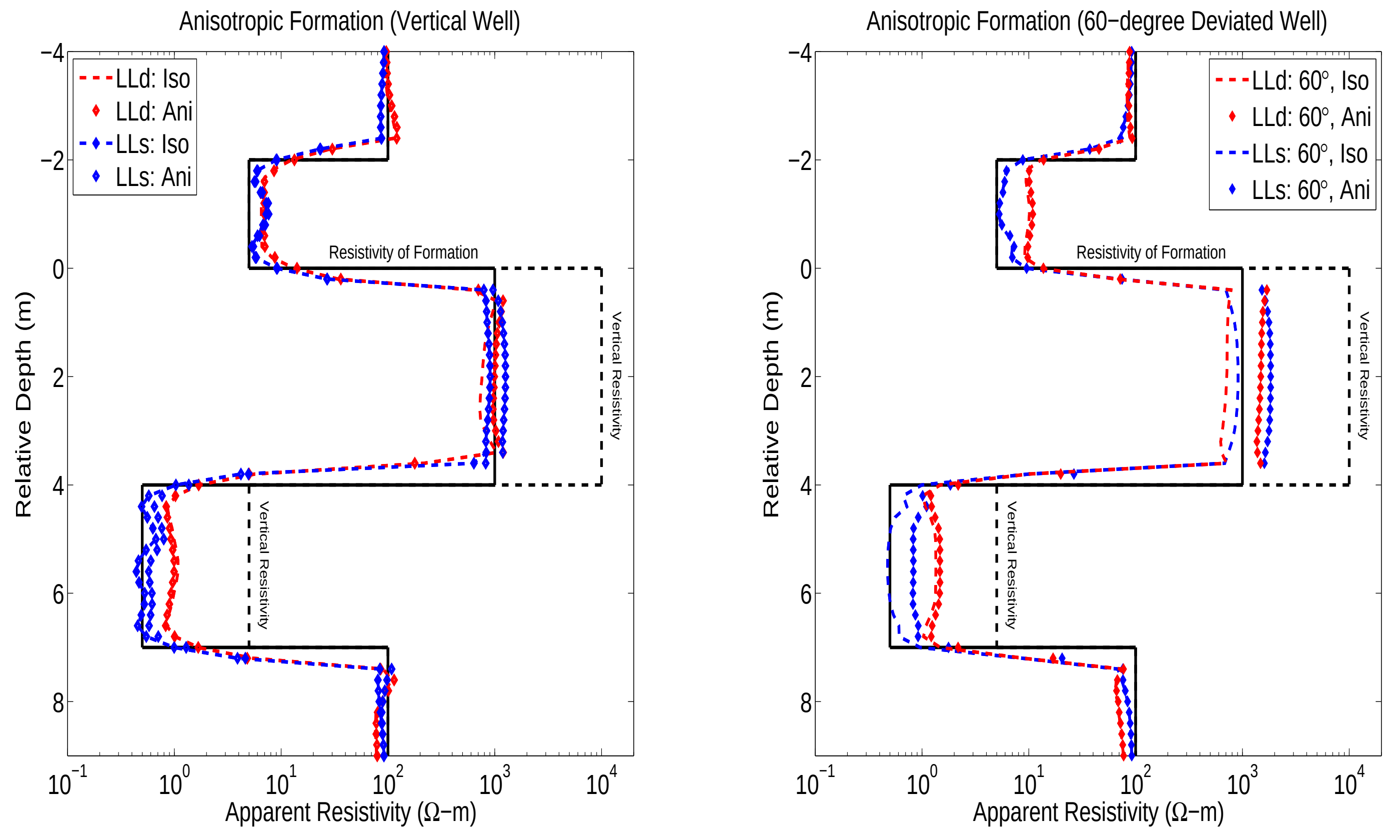




As we place the current return electrode B farther from the logging instrument, the Groningen effect diminishes

simulation of forward problems (hp-fem)

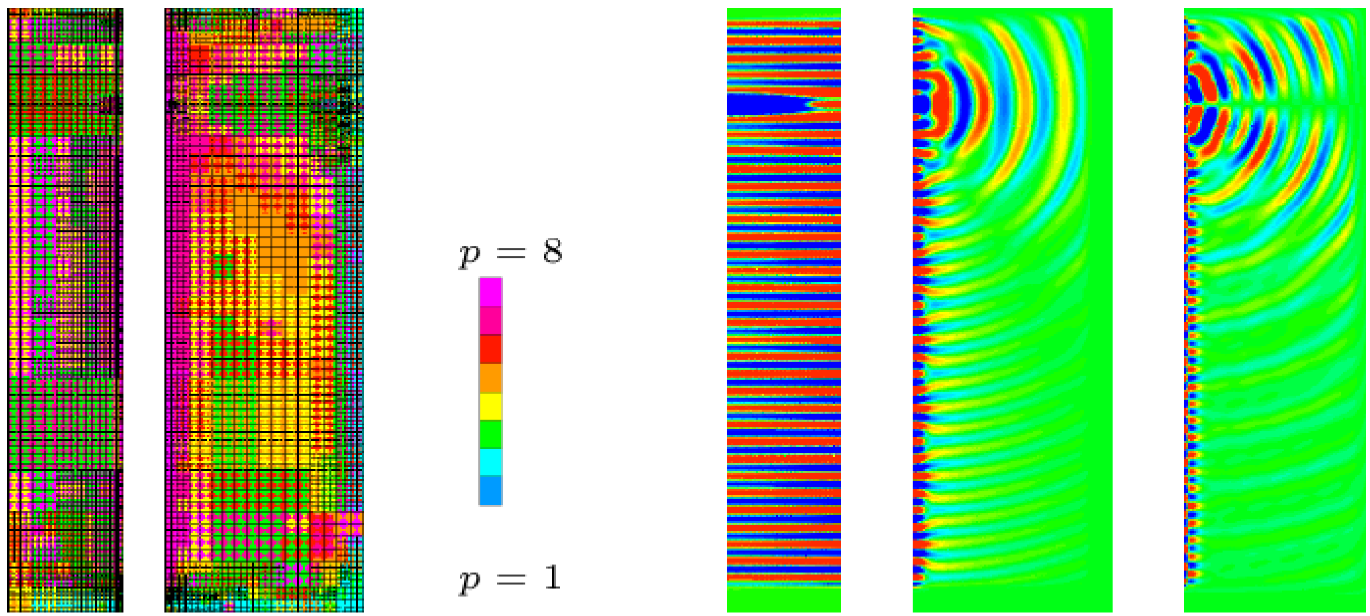
DC DLL in Deviated Wells



Anisotropy is better identified when using deviated wells

simulation of forward problems (hp-fem)

Final hp-grid and solution



acoustic hp-mesh | elastic hp-mesh | acoustic p_{acoust} | elastic u_r | elastic u_z

8 KHz, acoustics, open borehole setting (no logging instrument).

new library for inverse problems

Variational Formulation (DC)

Notation:

$$B(u, v; \sigma) =< \nabla v, \sigma \nabla u >_{L^2(\Omega)}$$ **(bilinear u, v)**

$$F_i(v) =< v, f_i >_{L^2(\Omega)} + < v, g_i >_{L^2(\partial\Omega)}$$ **(linear v)**

$$L_i(u) =< l_i, u >_{L^2(\Omega)} + < h_i, u >_{L^2(\partial\Omega)}$$ **(linear u)**

Direct Problem (homogeneous Dirichlet BC's):

$$\begin{cases} \text{Find } \hat{u}_i \in \text{V such that :} \\ B(\hat{u}_i, v; \sigma) = F_i(v) \quad \forall v \in V \end{cases}$$

Dual (Adjoint) Problem:

$$\begin{cases} \text{Find } \hat{v}_i \in \text{V such that :} \\ B(u, \hat{v}_i; \sigma) = L_i(u) \quad \forall u \in V \end{cases}$$

new library for inverse problems

Variational Formulation (AC)

Notation:

$$B(\mathrm{E},\mathrm{F};\sigma) = < \nabla\times\mathrm{F}, \mu^{-1}\nabla\times\mathrm{E} >_{L^2(\Omega)} - < \mathrm{F}, (\omega^2\epsilon - j\omega\sigma)\mathrm{E} >_{L^2(\Omega)}$$

$$F_i(\mathrm{F}) = -j\omega < \mathrm{F}, \mathrm{J}_i^{imp} >_{L^2(\Omega)} + j\omega < \mathrm{F}, \mathrm{J}_{S,i}^{imp} >_{L^2(\partial\Omega)}$$

$$L_i(\mathrm{E}) = < \mathrm{J}_i^{adj}, \mathrm{E} >_{L^2(\Omega)} + < \mathrm{J}_{S,i}^{adj}, \mathrm{E} >_{L^2(\partial\Omega)}$$

Direct Problem (homogeneous Dirichlet BC's):

$$\begin{cases} \text{Find } \hat{\mathrm{E}}_i \in \text{W such that :} \\ B(\hat{\mathrm{E}}_i, \mathrm{F};\sigma) = F_i(\mathrm{F}) \quad \forall \mathrm{F} \in W \end{cases}$$

Dual (Adjoint) Problem:

$$\begin{cases} \text{Find } \hat{\mathrm{F}}_i \in \text{W such that :} \\ B(\mathrm{E}, \hat{\mathrm{F}}_i;\sigma) = L_i(\mathrm{E}) \quad \forall \mathrm{E} \in W \end{cases}$$

new library for inverse problems

Constrained Nonlinear Optimization Problem

Cost Functional:

$$\begin{cases} \text{Find } \sigma > 0 \text{ such that it minimizes } C_\beta(\sigma)\text{, where :} \\ C_\beta(\sigma) = ||W_m(L(\hat{u}_\sigma) - M)||^2_{l_2} + \beta||R(\sigma - \sigma_0)||^2_{L_2} \, , \end{cases}$$

where

M_i denotes the i-th measurement, $M = (M_1, ..., M_n)$

L_i is the i-th quantity of interest, $L = (L_1, ..., L_n)$

$$||M||^2_{l_2} = \sum_{i=1}^{n} M_i^2 \quad ; \quad ||R(\sigma - \sigma_0)||^2_{L_2} = \int (R(\sigma - \sigma_0))^2$$

β is the relaxation parameter, σ_0 is given, W_m are weights

Main objective (inversion problem): Find $\hat{\sigma} = \min_{\sigma > 0} C_\beta(\sigma)$

new library for inverse problems

Solving a Constrained Nonlinear Optimization Problem

We select the following deterministic iterative method:

$$\boldsymbol{\sigma}^{(n+1)} = \boldsymbol{\sigma}^{(n)} + \boldsymbol{\alpha}^{(n)}\boldsymbol{\delta\sigma}^{(n)}$$

- **How to find a search direction $\delta\sigma^{(n)}$?**
  - **We will employ a change of coordinates and a truncated Taylor's series expansion.**
- **How to determine the step size $\alpha^{(n)}$?**
  - **Either with a fixed size or using an approximation for computing $L(\boldsymbol{\sigma}^{(n)} + \boldsymbol{\alpha}^{(n)}\boldsymbol{\delta\sigma}^{(n)})$.**
- **How to guarantee that the nonlinear constraints will be satisfied?**
  - **Imposing the Karush-Kuhn-Tucker (KKT) conditions or with a penalization method, or via a change of variables.**

new library for inverse problems

Search Direction Method

Change of coordinates:

$$h(s) = \sigma \quad => \text{ Find } \hat{s} = \min_{h(s)>0} C_\beta(s)$$

Taylor's series expansion:

A) $C_\beta(s+\delta s) \approx C_\beta(s) + \delta s \nabla C_\beta(s) + 0.5 \delta s^2 H_{C_\beta}(s)$

B) $L(s+\delta s) \approx L(s) + \delta s \nabla L(s)\ ,\ R(s+\delta s) = R(s) + \delta s \nabla R(s)$

Expansion A) leads to the Newton-Raphson method.

Expansion B) leads to the Gauss-Newton method.

Expansion A) with $H_{C_\beta} = I$ leads to the steepest descent method.

Higher-order expansions require from higher-order derivatives.

new library for inverse problems

Computation of Jacobian Matrix

Using the Fréchet Derivative:

$$\frac{\partial L_i(\hat{u}_i)}{\partial s_j} = B\left(\frac{\partial \hat{u}_i}{\partial s_j}, \hat{v}_i, h(s)\right) + B\left(\hat{u}_i, \frac{\partial \hat{v}_i}{\partial s_j}, h(s)\right) + B\left(\hat{u}_i, \hat{v}_i, \frac{\partial h(s)}{\partial s_j}\right)$$

$$\|$$

$$L_i\left(\frac{\partial \hat{u}_i}{\partial s_j}\right) = B\left(\frac{\partial \hat{u}_i}{\partial s_j}, \hat{v}_i, h(s)\right)$$

$$\|$$

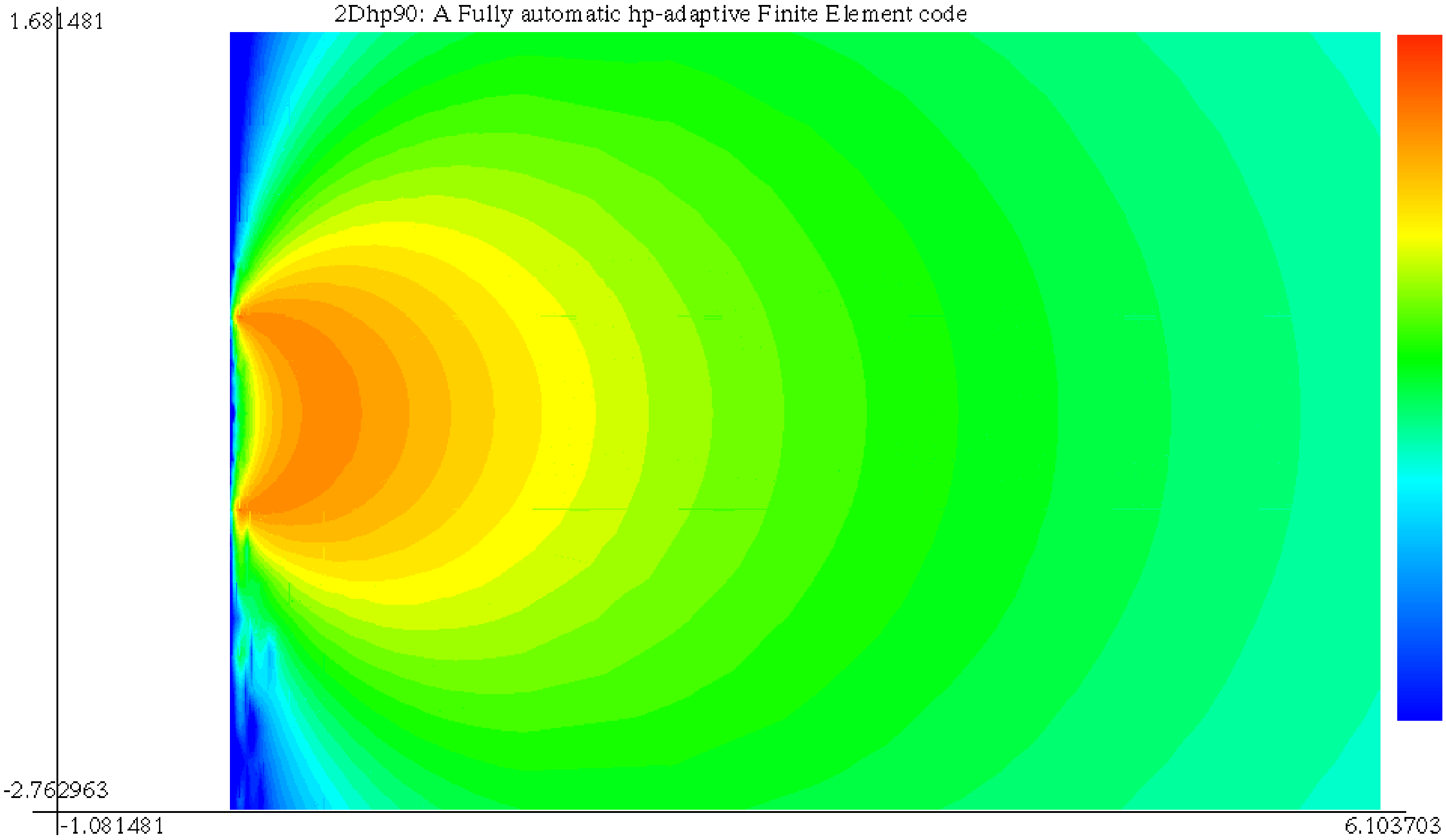
$$F_i\left(\frac{\partial \hat{v}_i}{\partial s_j}\right) = B\left(\hat{u}_i, \frac{\partial \hat{v}_i}{\partial s_j}, h(s)\right)$$

Therefore, we conclude:

Jacobian Matrix $= \dfrac{\partial L_i(\hat{u}_i)}{\partial s_j} = -B\left(\hat{u}_i, \hat{v}_i, \dfrac{\partial h(s)}{\partial s_j}\right)$

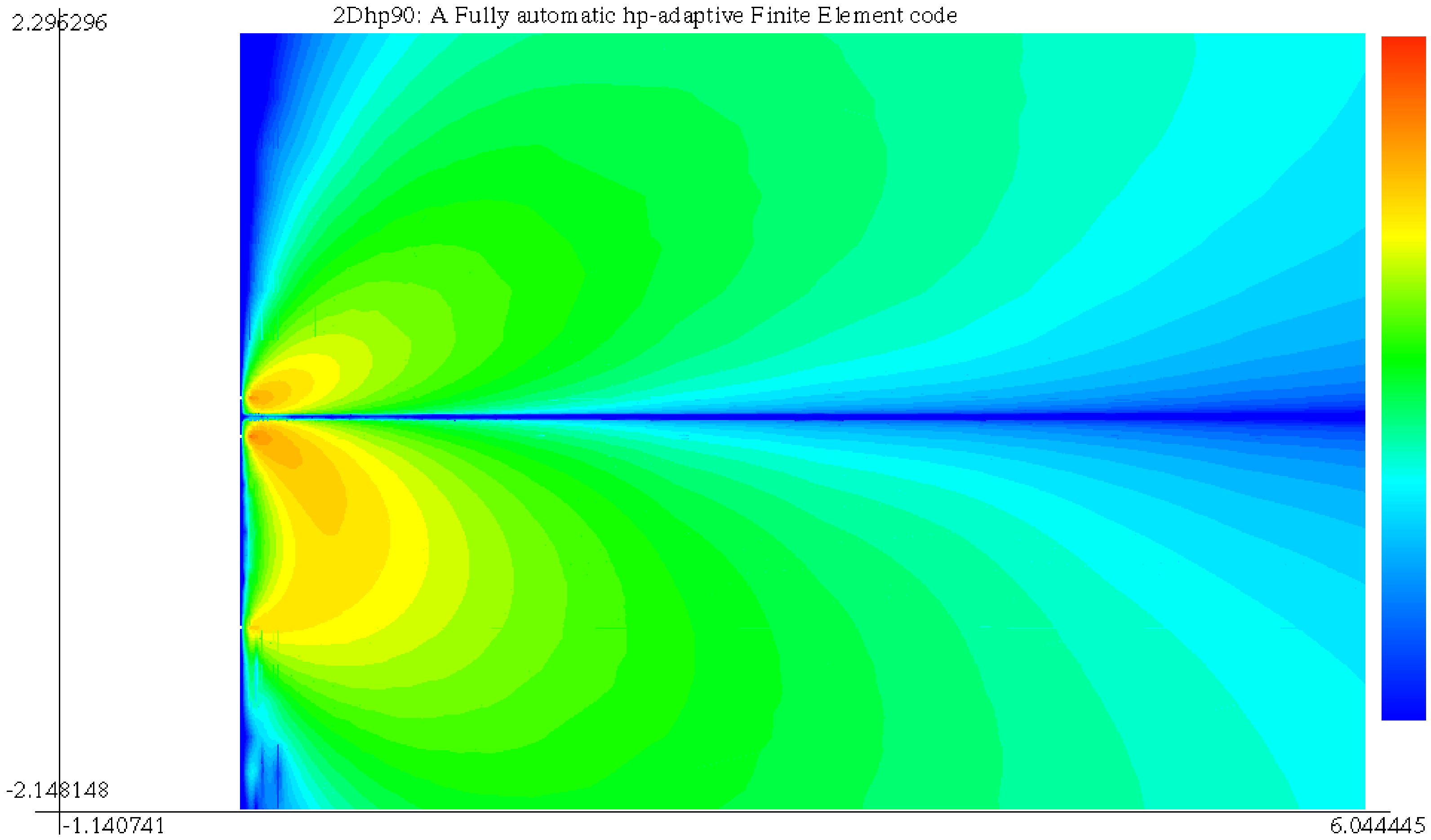
new library for inverse problems

Jacobian Function: One TX, one RX



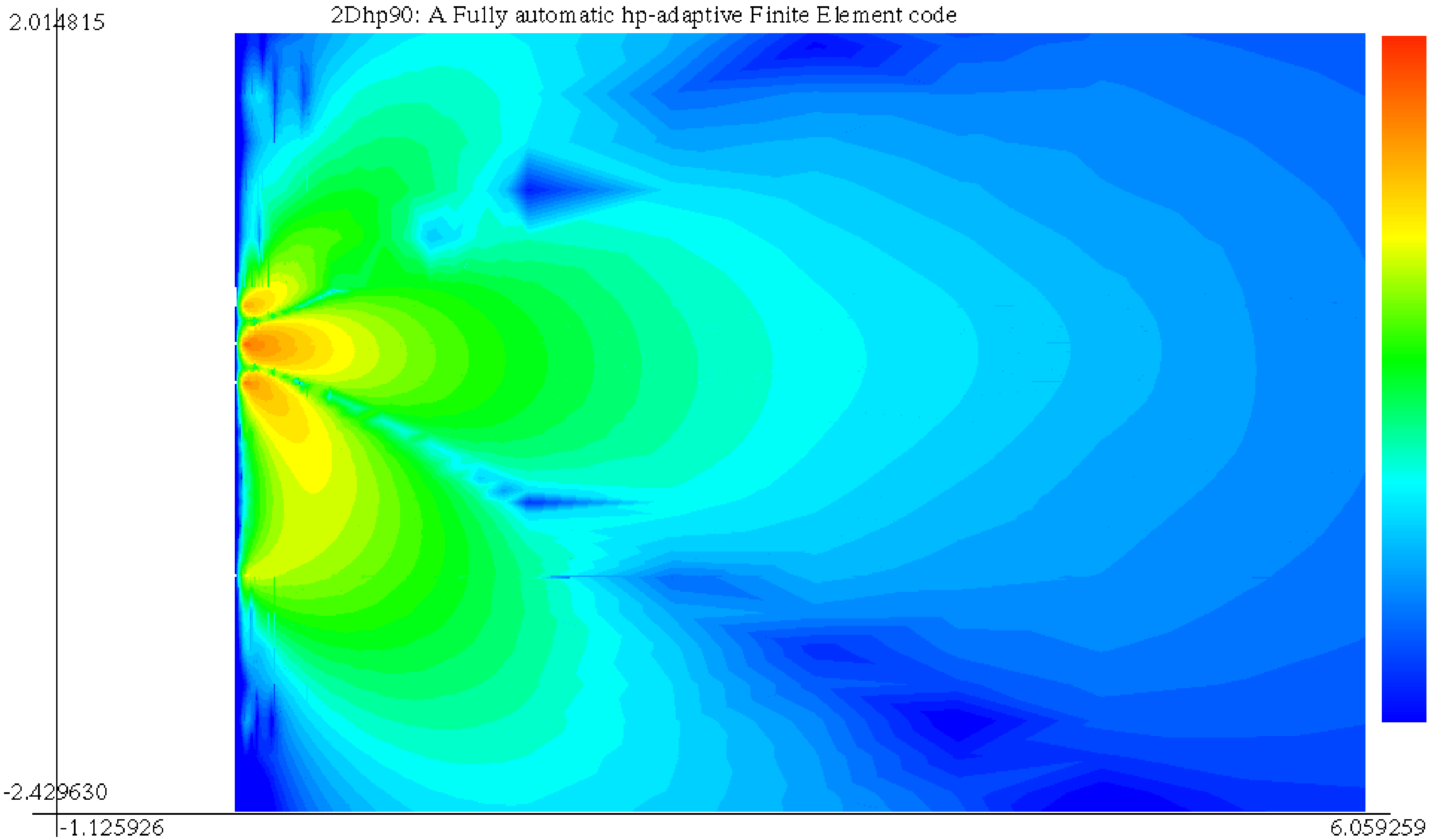
new library for inverse problems

Jacobian Function: One TX, one RX



new library for inverse problems

Jacobian Function: One TX, one RX



new library for inverse problems

Computation of Hessian Matrix

Following a similar argument as for the Jacobian matrix, we obtain:

$$\frac{\partial^2 L_i(\hat{u}_i)}{\partial s_j \partial s_k} = -B\left(\frac{\partial \hat{u}_i}{\partial s_j}, \hat{v}_i, \frac{\partial h(s)}{\partial s_k}\right) - B\left(\hat{u}_i, \frac{\partial \hat{v}_i}{\partial s_j}, \frac{\partial h(s)}{\partial s_k}\right) - B\left(\hat{u}_i, \hat{v}_i, \frac{\partial^2 h(s)}{\partial s_j \partial s_k}\right)$$

How do we compute $\frac{\partial \hat{u}_i}{\partial s_j}$ and $\frac{\partial \hat{v}_i}{\partial s_j}$?

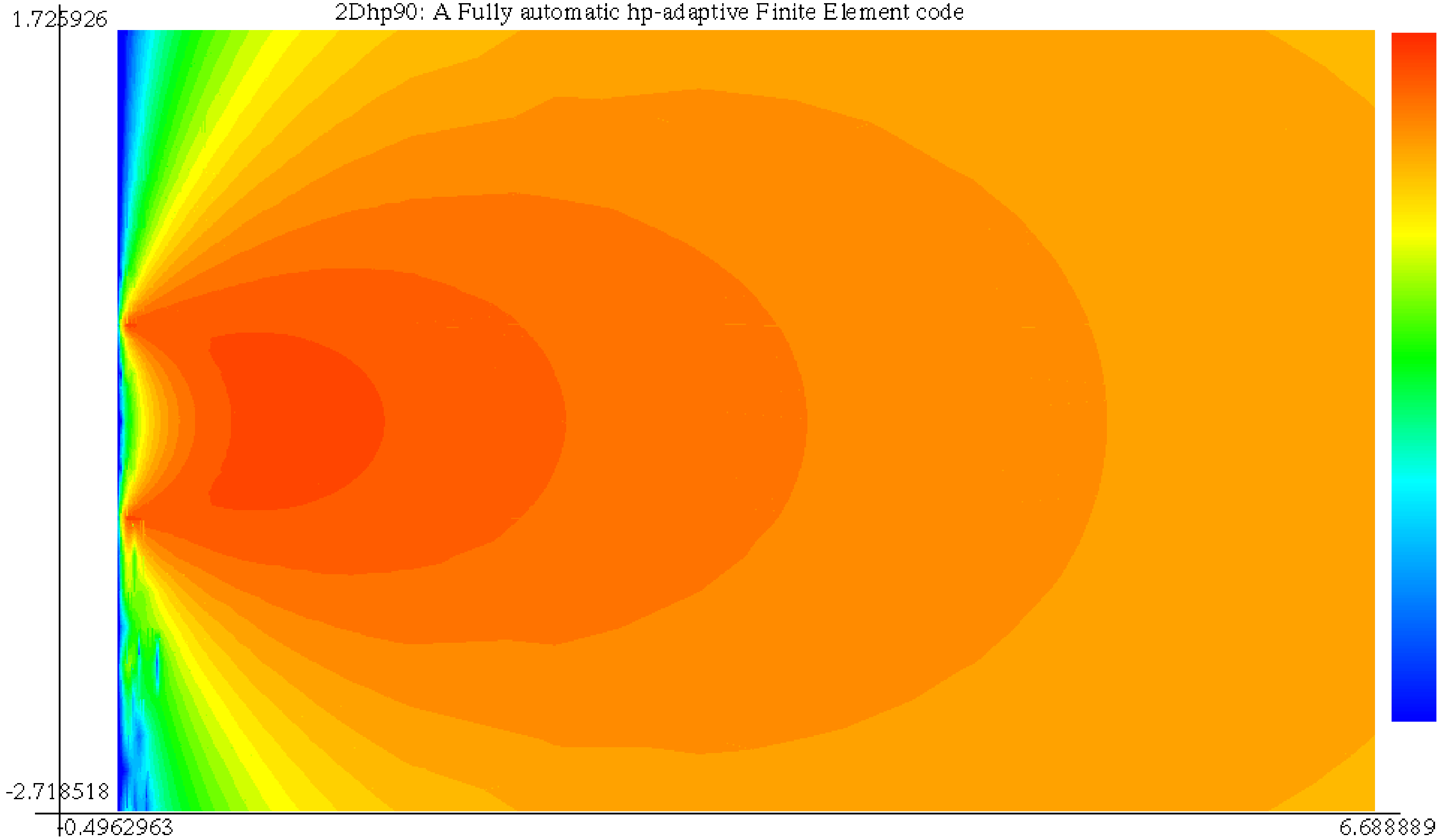
Find $\frac{\partial \hat{u}_i}{\partial s_j}$ such that : $B\left(\frac{\partial \hat{u}_i}{\partial s_j}, v_i, h(s)\right) = -B\left(\hat{u}_i, v_i, \frac{\partial h(s)}{\partial s_j}\right) \quad \forall v_i$

Find $\frac{\partial \hat{v}_i}{\partial s_j}$ such that : $B\left(\frac{\partial \hat{v}_i}{\partial s_j}, u_i, h(s)\right) = -B\left(\hat{v}_i, u_i, \frac{\partial h(s)}{\partial s_j}\right) \quad \forall u_i$

We can compute the Hessian matrix EXACTLY by just solving our original problem for different right-hand-sides, and performing additional integrations.

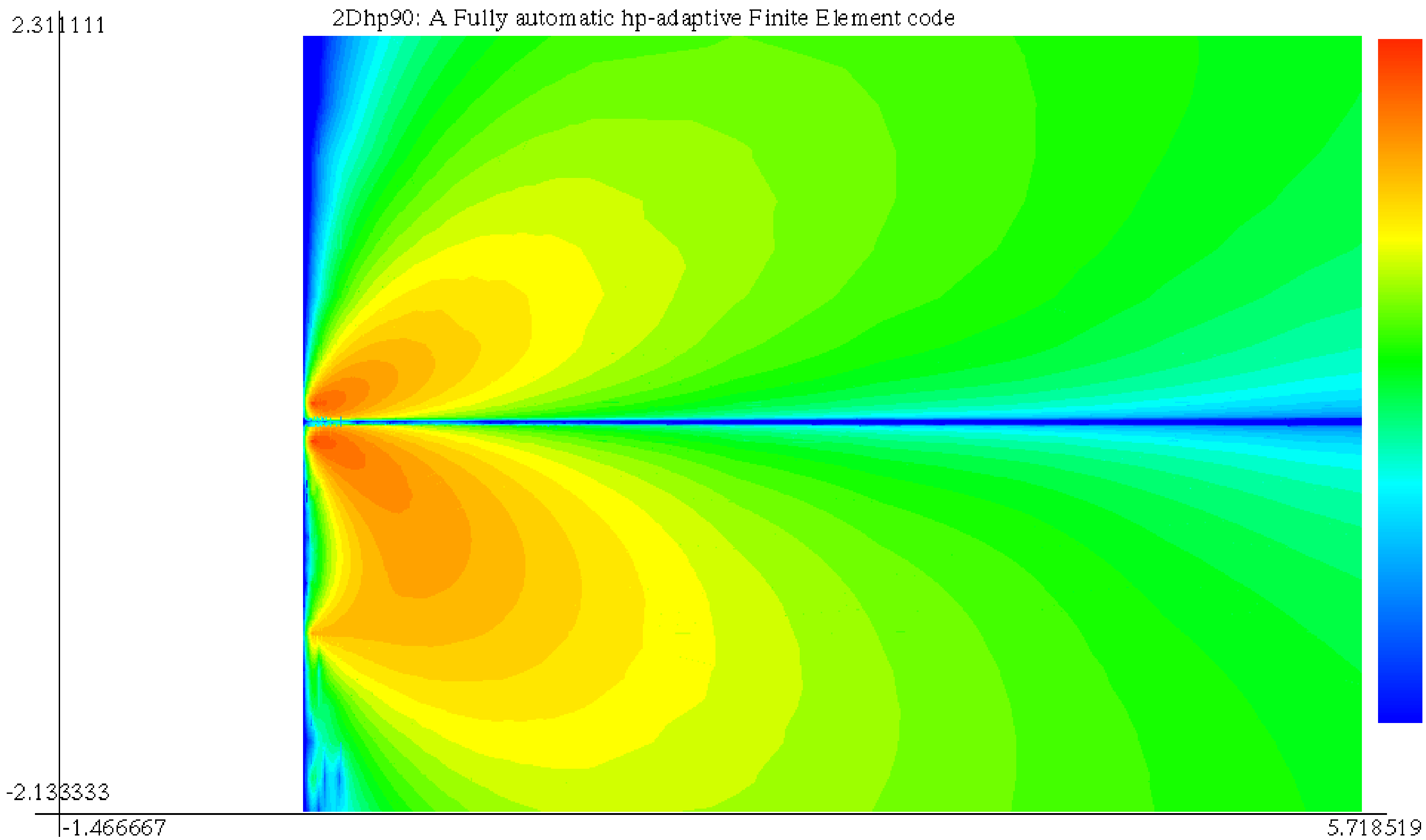
new library for inverse problems

Hessian Function: One TX, one RX



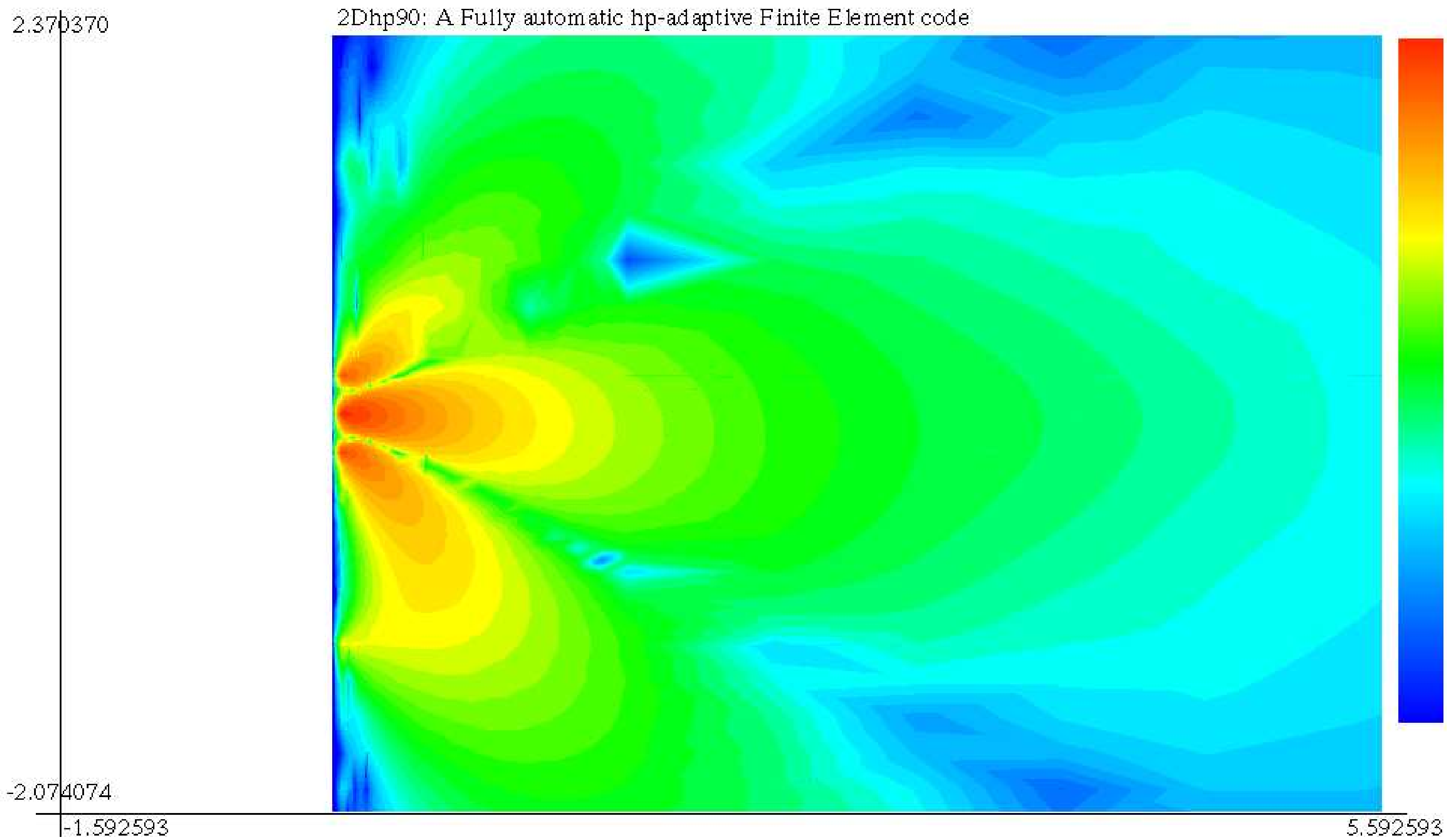
new library for inverse problems

Hessian Function: One TX, two RXs



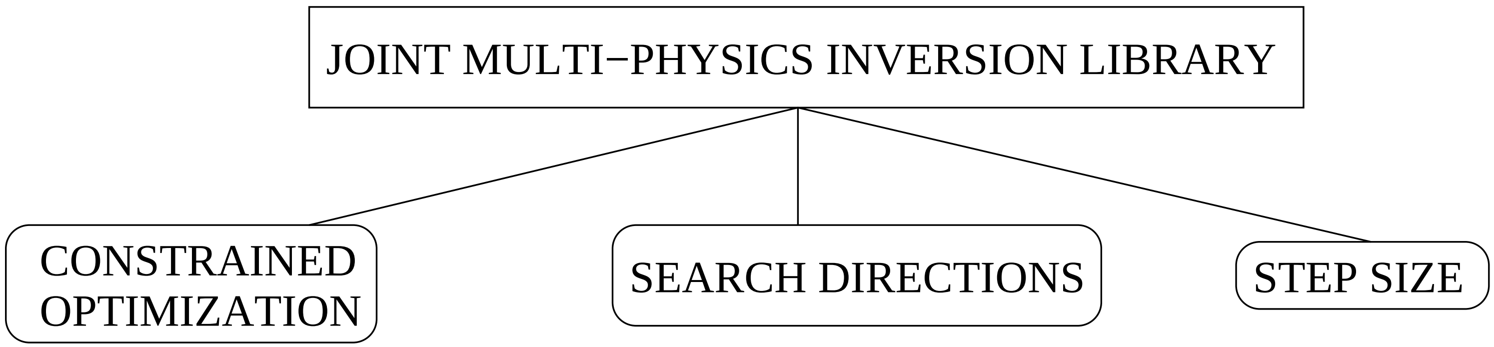
new library for inverse problems

Hessian Function: One TX, three RXs



new library for inverse problems

Algorithms implemented within the inverse library



1. KKT− Conditions
2. Penalization Method
3. Change of coordinates

1. Steepest Descent
2. Gauss−Newton
3. Newton−Raphson

1. Uniform step−size
2. Variable step−size (multiple algorithms)

The inverse library is composed of multiple algorithms for imposing constraints, and finding search directions and corresponding step sizes.

Jacobian and Hessian matrices are computed exactly by simply solving the dual (adjoint) formulation and performing additional integrations.

The inverse library is compatible with multi-physics problems.

conclusions

- We have described an efficient numerical method for solving PDE's based on a self-adaptive goal-oriented hp refinement strategy.
- We are developing a multiphysics version of the code.
- We are building a new module for the resolution of inverse problems.
- Our main objective is to create a software infrastructure enabling solution of challenging multiphysics inverse problems with applications to geophysics (hydrocarbon detection and monitoring, etc.), aeronautics and medicine.
- To achieve this objective, we need Ph.D. students, post-doctoral fellows, experienced researchers, and collaborators in different areas (inversion, solvers, etc).