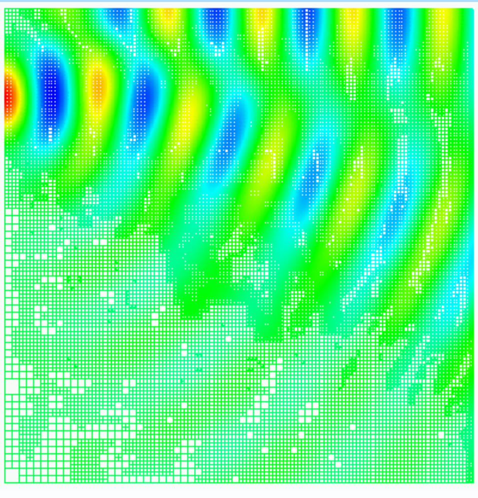
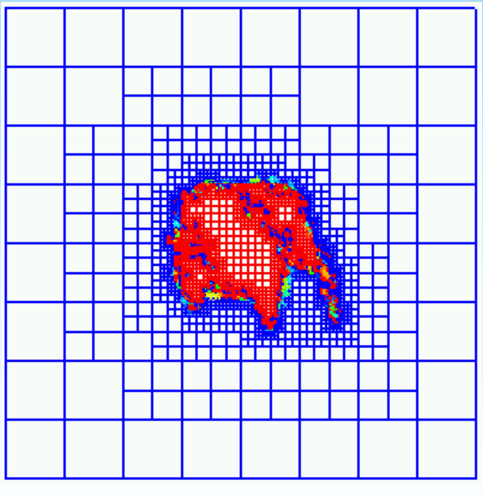
Duality-Based Error Estimates and their Application to Inverse Problems

Wolfgang Bangerth

Institute for Computational Engineering
Institute for Geophysics

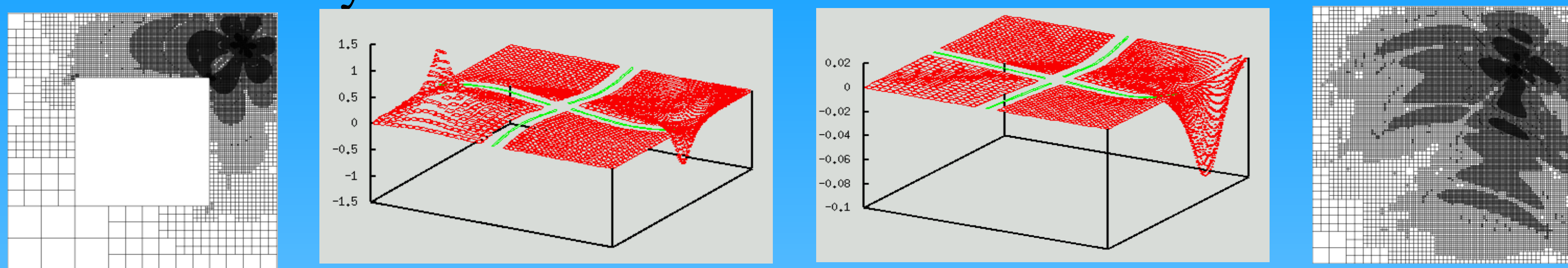
The University of Texas at Austin



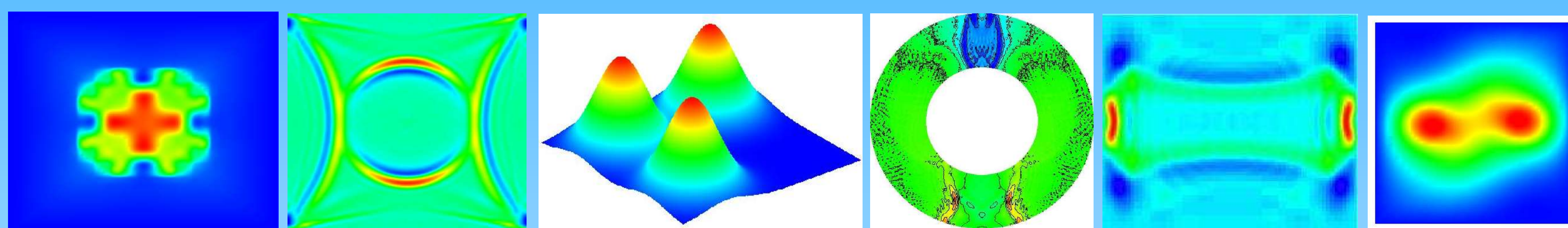


My different hats

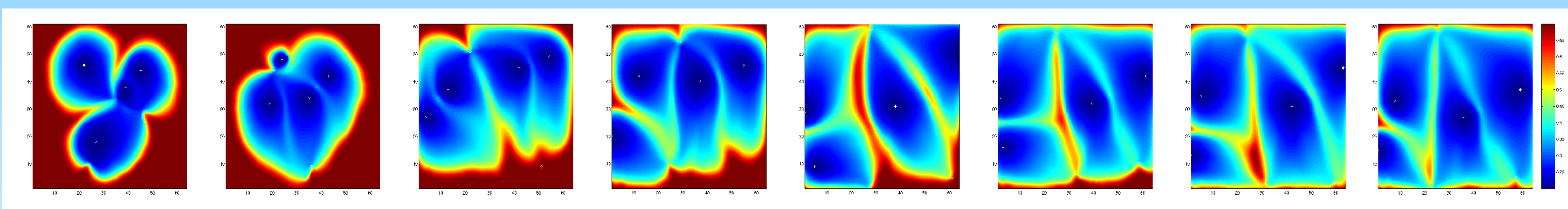
- Numerical Analyst: Efficient discretizations and error estimates



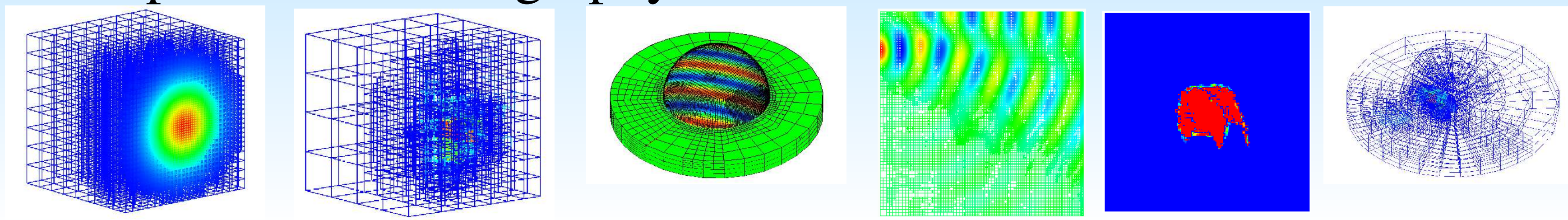
- Software designer and engineer: The *deal.II* finite element library



- Geophysicist: Optimization of oil and water reservoir operations



- Inverse problems/Tomography



What and why?

In finite element computations, we can only approximate the real, but unknown solution u of a differential or integral equation by an approximation u_h.

To be sure that our computed approximation satisfies practical requirements (e.g. safety margins), we want to know by how much u and u_h differ.

Part of numerical analysis is concerned with *error estimates* for this difference, i.e. with quantifying $u-u_h$ with knowing only u_h but not the exact solution u (*a posteriori* error estimates). Ideally, such estimates are also used for mesh refinement.

The standard way

Assume we wanted to solve the variational equality

$$a(u,v)=(f,v) \qquad \forall v\in V$$

For example, when solving the Laplace equation:

$$(\nabla u,\nabla v) = (f,v) \qquad \forall v\in H^1$$

Then, the Galerkin solution is given by defining $V_h\subset V$ and finding $u_h\in V_h$:

$$a(u_h,v_h)=(f,v_h) \qquad \forall v_h\in V_h$$

For the error $e=u\text{-}u_h$, we then know that Galerkin orthogonality holds:

$$a(u-u_h,v_h)=0 \qquad \forall v_h\in V_h$$

The standard way

A simple way to error estimates for linear problems:

Assume we have *inf-sup* stability and interpolation estimates

$$\sup_{v\in V} \frac{a(u,v)}{\|v\|_V} \geq \gamma \|u\|_V$$

$$\inf_{v_h\in V_h} \|v - v_h\| \leq Ch\|v\|_V$$

We then immediately have an a posteriori estimate:

$$\|e\|_V \leq \frac{1}{\gamma} \sup_v \frac{a(e,v)}{\|v\|_V} = \frac{1}{\gamma} \sup_v \inf_{v_h\in V_h} \frac{a(e, v - v_h)}{\|v\|_V}$$

$$\leq \frac{Ch}{\gamma} \|R(u_h)\|$$

The standard way

For example, for the Laplace equation:

$$\sup_{v\in V}\frac{a(u,v)}{\|v\|_V}=\sup_{v\in V}\frac{(\nabla u,\nabla v)}{\|v\|_{H^1}}=\frac{(\nabla u,\nabla u)}{\|u\|_{H^1}}\geq\gamma\|u\|_V$$

$$\inf_{v_h\in V_h}\|u-v_h\|\leq Ch\|u\|_V$$

The error estimate then reads:

$$\|e\|_V\leq\frac{Ch}{\gamma}\|R(u_h)\|=C\left(\sum_K h\|f+\Delta u_u\|_K^2+h^2\|[\frac{\partial u_h}{\partial n}]\|_{\partial K}^2\right)^{1/2}$$

Why isn't this sufficient?

This approach lacks in several respects:

- The *energy norm* $\|u-u_h\|_V$ may not always be the norm with respect we would like to measure the error (e.g. lift and drag coefficients)

- Stability estimates often yield no numerical value for the stability constant γ or underestimate it by so much that actual estimates are unreliable (e.g. almost all equations)

- Some equations or formulations may be stable, but stability hasn't been proved yet (e.g. in particular for nonlinear equations)

- Some equations may admit stable solutions sometimes, but not for all data (e.g. Navier-Stokes for laminar/turbulent flow)

Overview

- What we want instead
- Duality-based error estimates
- Application to inverse problems
- Numerical examples

What we want instead

- Estimates in terms of output functionals: $J(u)-J(u_h)$, where $J(.)$ could be
  - the lift or drag of an airfoil
  - the flow into a well
  - the location of a reconstructed tumor

- Estimates should not depend on analytical expressions of stability constants, but rather compute them as part of the algorithm

- Idea: Use a dual problem that needs to be solved numerically to assess the actual stability problems of the primal solution

Formulation of estimates

Basic idea (Johnson et al. 1990s, Becker & Rannacher 1995, 2001, Bangerth & Rannacher 2003): Assume for simplicity that both *a(.,.)* and *J(.)* are linear. Then define the solution *z* of dual problem

$$J(v) = a(v, z) \qquad \forall v \in V$$

Then the error is immediately expressed as

$$\begin{aligned} J(u) - J(u_h) &= J(e) = a(e, z) = a(e, z - v_h) \\ &= a(u, z - z_h) - a(u_h, z - z_h) = (f, z - z_h) - a(u_h, z - z_h) \end{aligned}$$

For example, for the Laplace equation:

$$J(u) - J(u_h) = \sum_K (f + \Delta u_h, z - z_h)_K - \frac{1}{2} \left([\frac{\partial u_h}{\partial n}], z - z_h \right)_{\partial K}$$

Formulation of estimates

The estimate

$$J(u) - J(u_h) = (f, z - z_h) - a(u_h, z - z_h)$$

does not contain the unknown exact solution u or any unknown or poorly known stability constants any more. However, it now contains the unknown dual solution z . To be a practically viable way to estimate the error in $J(.)$, we need to approximate it.

Unfortunately, we can't use the same approximation scheme as for the primal solution. There are various other methods, though (Bangerth & Rannacher, 2003), and the development of very efficient schemes is possible for different equations (Rannacher, Becker, Kanschat, Bangerth).

On the other hand, the derivation of similar estimates are straightforward even for nonlinear problems or nonlinear nonlinear target functionals.

Comparison of estimates

Traditional estimates of the form

$$\|e\|_V \leq \frac{Ch}{\gamma} \|R(u_h)\|$$

are typically global, i.e. they consider the presence of a residual equally important everywhere in the domain. On the contrary, duality-based estimates *weigh* the residual with the dual solution:

$$J(u) - J(u_h) = (f, z - z_h) - a(u_h, z - z_h) = (R(u_h), z - z_h)$$

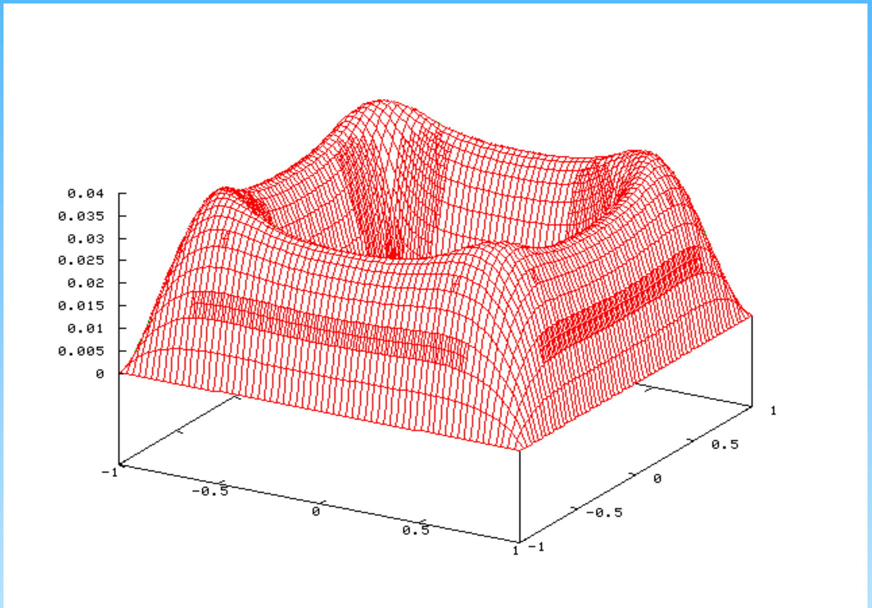
indicating that a residual in a remote part of the domain may not be as important to the accuracy with which we approximate *J(u)*.

This ensures that

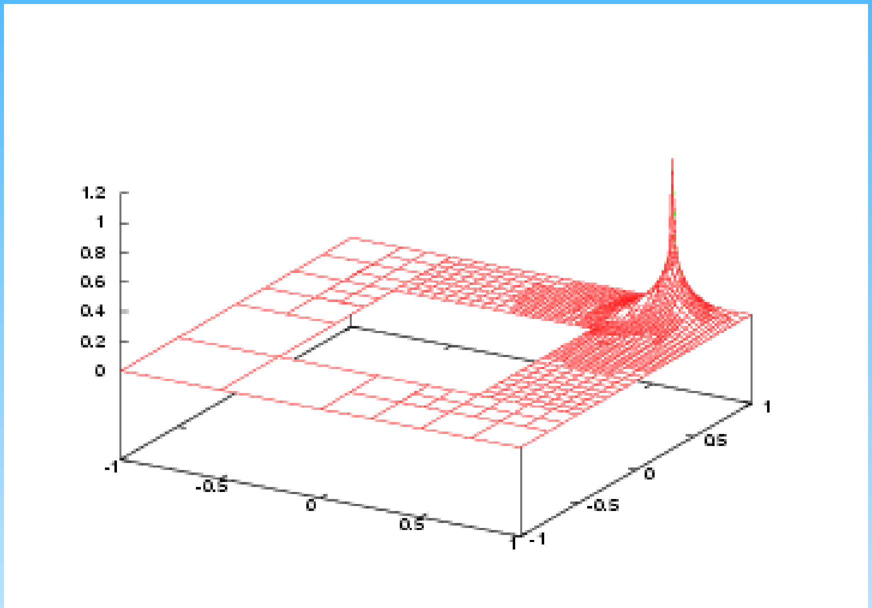
- estimates are accurate
- we can generate meaningful refinement indicators from the estimates

Example

Example: Estimate the error in the vicinity of a single point for the solution of a Laplace equation (from Bangerth, Kanschat, & Hartmann, 2004):



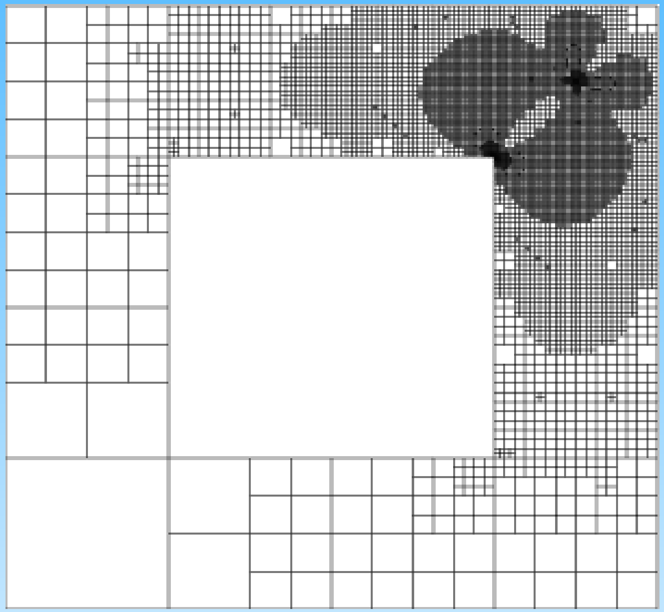
Primal solution

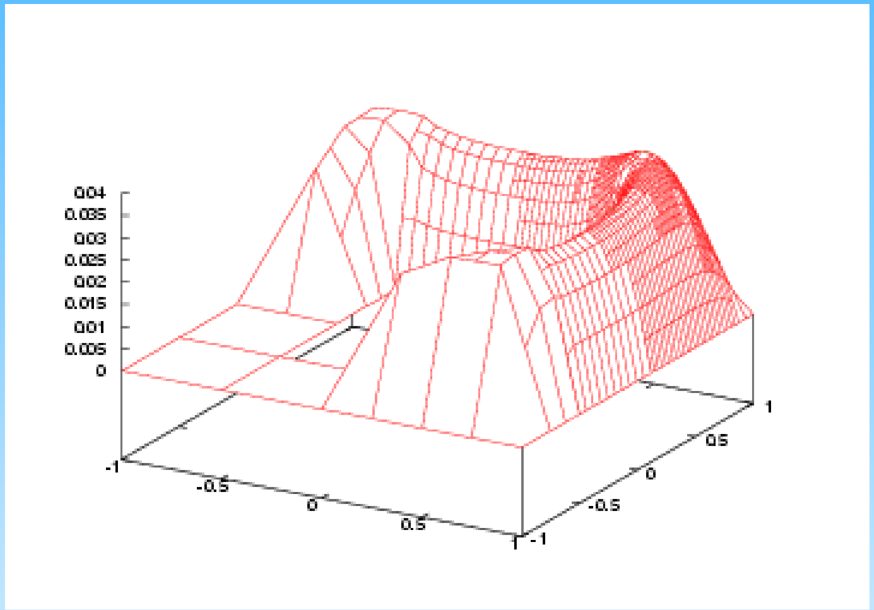


Dual solution

Example

Example: Estimate the error in the vicinity of a single point for the solution of a Laplace equation. Using duality-based error estimates to refine the mesh, we realize that fine resolution is only required close to the target area, not elsewhere:



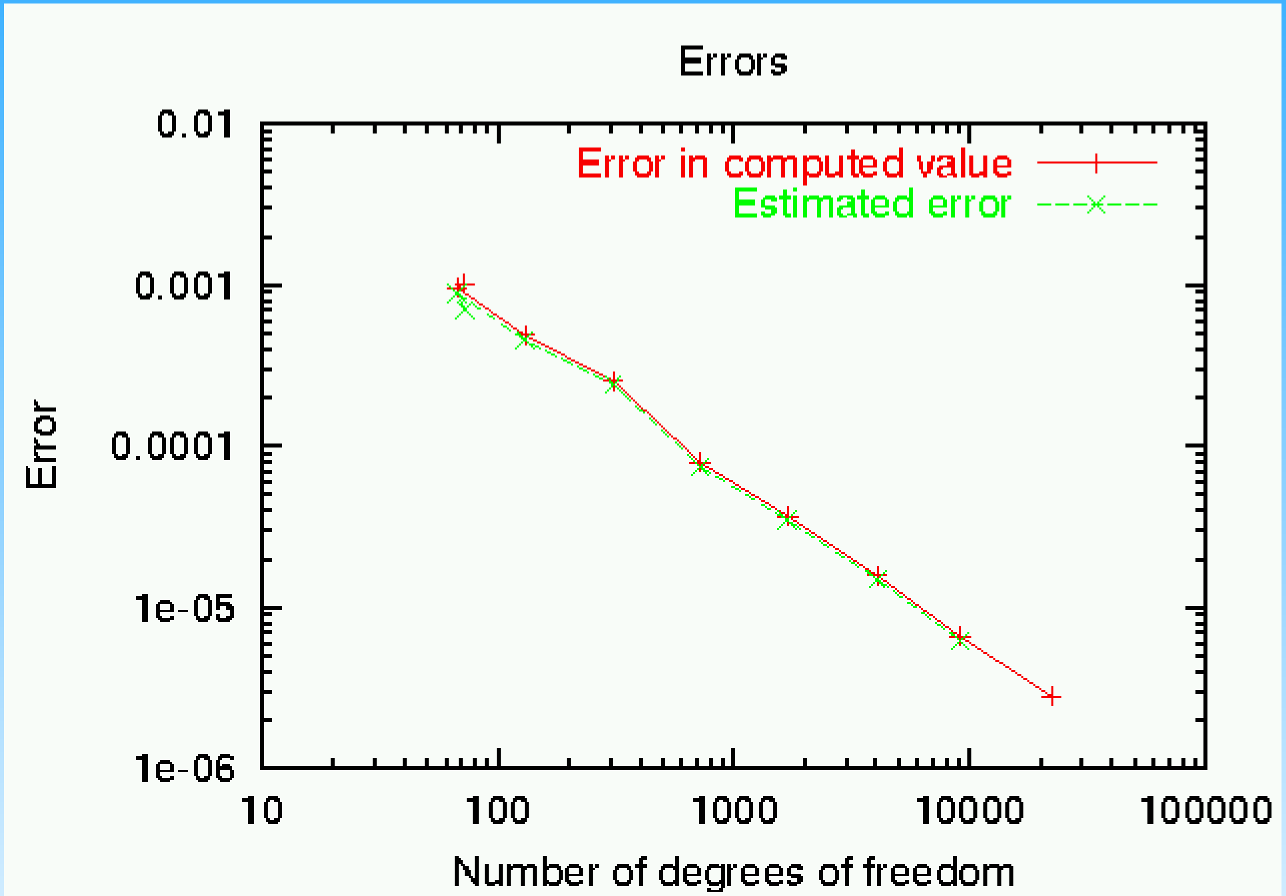


Mesh

Solution on this mesh

Example

Example: Comparison of real and estimated error:



Application to inverse problems

In inverse problems, one wants to identify a coefficient in a PDE from measurements of its solution (output) for a known source (input).

For example, identify the stiffness coefficient $q=q(x)$ in

$$-\nabla \cdot q \nabla u = f$$

by comparing the solution $u(x)$ with measured values $z(x)$. Such problems appear in almost all biomedical and geophysical imaging applications.

- Solution is usually very expensive, since PDE is only a subproblem
- Accurate error estimates as termination criteria therefore important
- Mesh refinement based on estimates even more important to generate efficient discretizations
- Problem often ill-posed, i.e. stability estimates unknown or inexistant

Formulation of inverse problem

Reconstruct *q(x)* by minimizing

$$M(u,q) = \frac{1}{2}\|u-z\|^2 + \frac{\beta}{2}\|q\|^2$$

subject to the state equation

$$(q\nabla u, \nabla v) = (f, v) \qquad \forall v \in V$$

Set $x = \{u, \lambda, q\}$, and introduce the Lagrangian:

$$L(x) = \frac{1}{2}\|u-z\|^2 + \frac{\beta}{2}\|q\|^2 + (q\nabla u, \nabla\lambda) - (f, \lambda)$$

Optimum then stationary point of *L(x)*:

$$\nabla_x L(x)(y) = 0 \quad \forall y \longleftrightarrow \quad \begin{aligned} (q\nabla\lambda, \nabla\psi) &= -(u-z, \psi) \\ (\beta q + \nabla u \cdot \nabla\lambda, \chi) &= 0 \\ (q\nabla u, \nabla\varphi) &= (f, \varphi) \end{aligned}$$

A posteriori error estimate

Assume that we want to estimate the error with respect to arbitrary functionals *J(x)* ? E.g. $J(x)=\|q\|^2_{\Omega'}$

Define dual solution z by

$$\nabla^2_x L(x)(z,y) = -\nabla_x J(x)(y) \qquad \forall y$$

$\xi=\{x,z\}$ is stationary point of combined Lagrangian

$$\Lambda(\xi) = J(x)+\nabla_x L(x)(z)$$

Since continuous and discrete solutions satisfy stationarity of *L*:

$$J(x)-J(x_h) = \Lambda(\xi)-\Lambda(\xi_h) = \int_0^1 \nabla_\xi \Lambda(\xi_h+se_\xi)(e_\xi)\,ds$$

$$= \frac{1}{2}\underbrace{\nabla_\xi \Lambda(\xi)(e_\xi)}_{=0}+\frac{1}{2}\underbrace{\nabla_\xi \Lambda(\xi_h)(e_\xi)}_{\nabla_\xi \Lambda(\xi_h)(\xi-I_h\xi)}+R = \frac{1}{2}\nabla_\xi \Lambda(\xi_h)(\xi-I_h\xi)+R$$

A posteriori error estimate

For other partial differential equations, one does not need a dual solution to prove error estimates in the natural energy norm. Or, rather, the dual solution equals the primal one. For nonlinear target functionals, the dual solution is defined by

$$a(v,z)=J'(u)(v) \qquad \forall v\in V$$

If $J(.)$ is the squared energy norm, the $a(.,.)=J'(.)(.)$. For the Laplace equation,

$$J(v)=\frac{1}{2}\|\nabla v\|^2; \qquad\qquad a(u,v)=(\nabla u,\nabla v)$$

$$a(v,z)=(\nabla v,\nabla z) \;=\; (\nabla u,\nabla v) \qquad \forall v\in V$$

Since the solution of inverse problems is so expensive, we would like to not compute a dual solution of it and use a trick as above. This is indeed possible (Bangerth 2002).

Estimates with respect to $M(.)$

In the inverse problem, we are minimizing the misfit

$$M(u,q) = \frac{1}{2}\|u-z\|^2 + \frac{\beta}{2}\|q\|^2$$

subject to the state equation constraint

$$(q\nabla u, \nabla v) = (f, v) \qquad \forall v \in V$$

If one wants to estimate the error with respect to the functional *J(x)=M(x)*, it can be shown that the dual solution z is indeed equal to the primal solution x.

In that case, error estimates significantly simplify and are also numerically cheap to evaluate!

Estimates with respect to $M(.)$

In the general case, the error estimate had the form

$$J(u)-J(u_h)=(f,z-z_h)-a(u_h,z-z_h)$$

For the present problem, this reads

$$M(x)-M(x_h)=-\nabla_x L(x_h,z-z_h)$$

However, for this special case, primal and dual solution were the same, so we obtain:

$$\begin{aligned} M(x)-M(x_h) &= \frac{1}{2}\sum_K(\rho_u,\lambda-\lambda_h)_K+\text{jump term} \\ &\quad +(\rho_\lambda,u-u_h)_K+\text{jump term} \\ &\quad +(\rho_q,q-q_h)_K \\ &\quad +R \end{aligned}$$

$$\rho_u=f+\nabla\cdot q_h\nabla u_h$$

$$\rho_\lambda=(u-z)+\nabla\cdot q_h\nabla\lambda_h$$

$$\rho_q=\beta q+\nabla u_h\cdot\nabla\lambda_h$$

$$R=-\frac{1}{12}(e_q\nabla e_u,\nabla e_\lambda)$$

Estimates with respect to $M(.)$

$$
\begin{aligned}
M(x)-M(x_h) = \frac{1}{2}\sum_{i=1}^{N} \sum_K (\rho_u^i, \lambda^i-\lambda_h^i)_K + \text{jump term} \\
+ (\rho_\lambda^i, u^i-u_h^i)_K + \text{jump term} \\
+\frac{1}{2}\sum_{K_q} (\rho_q, q-q_h)_{K_q} \\
+R
\end{aligned}
$$

Error estimate provides us with

- An error estimate
- Refinement indicators for each cell of
  - the meshes for each state equation
  - the mesh used in the discretization of the parameters

Remainder is cubic in the errors and can presumably be neglected

Numerical experiments

Use numerical experiments to verify:

- accuracy of error estimates
- efficiency of error estimates as refinement criteria

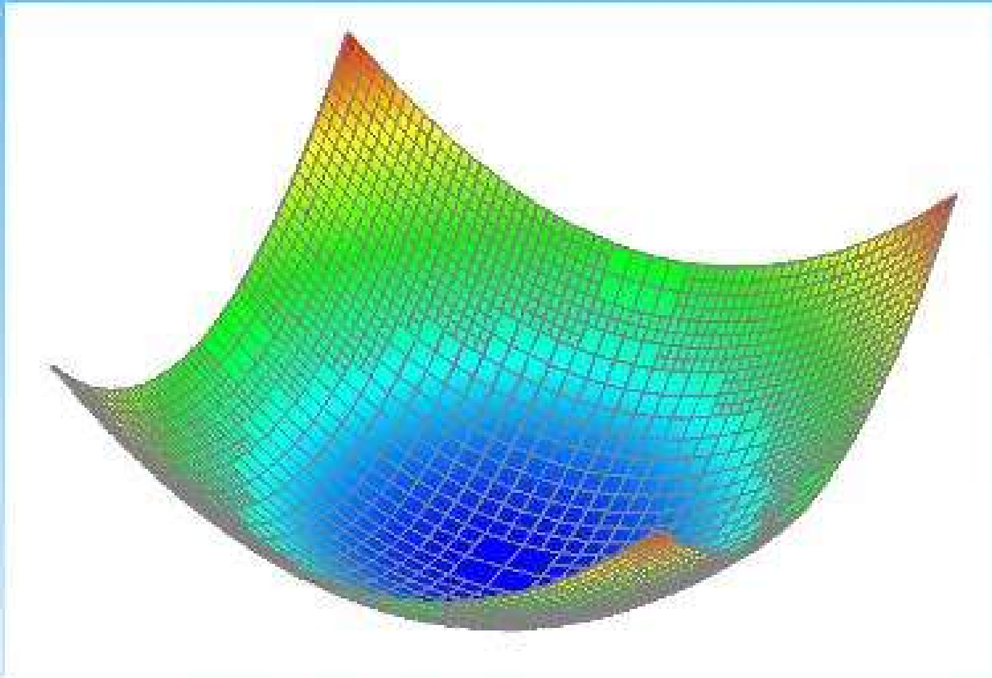
Compare our estimator with ad-hoc refinement criteria:

$$\begin{aligned} \eta_K^u &= h^s \|\nabla_h^2 u\|_K \\ \eta_K^\lambda &= h^s \|\nabla_h^2 \lambda\|_K \end{aligned}$$

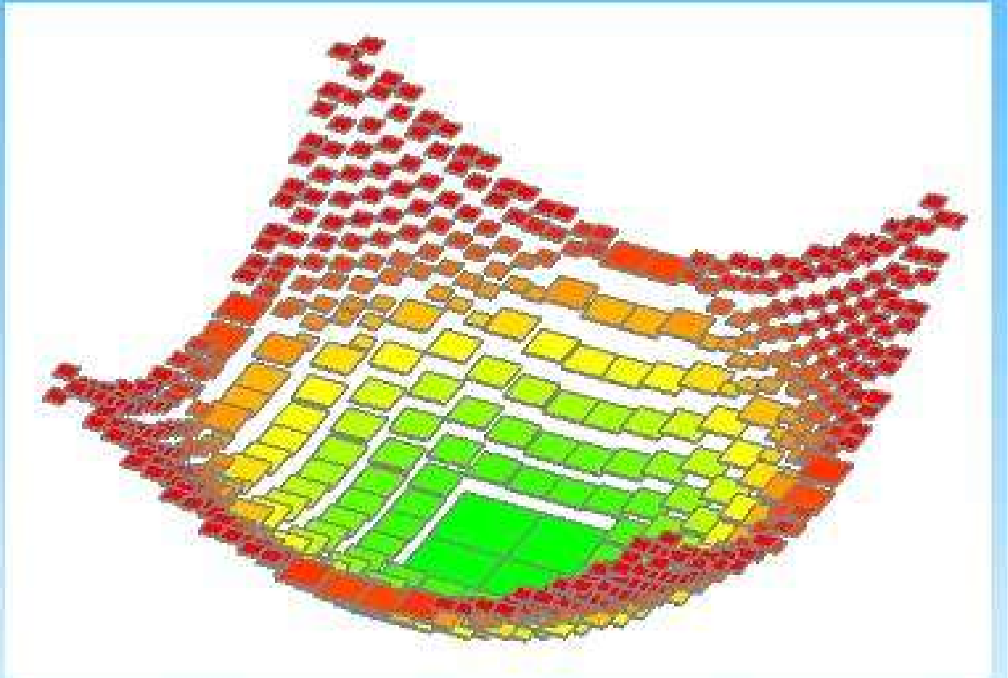
Example 1: Smooth Coefficient

Exact solution:

$$u = |x|^2, \qquad a = 1 + |x|^2, \qquad z = u$$



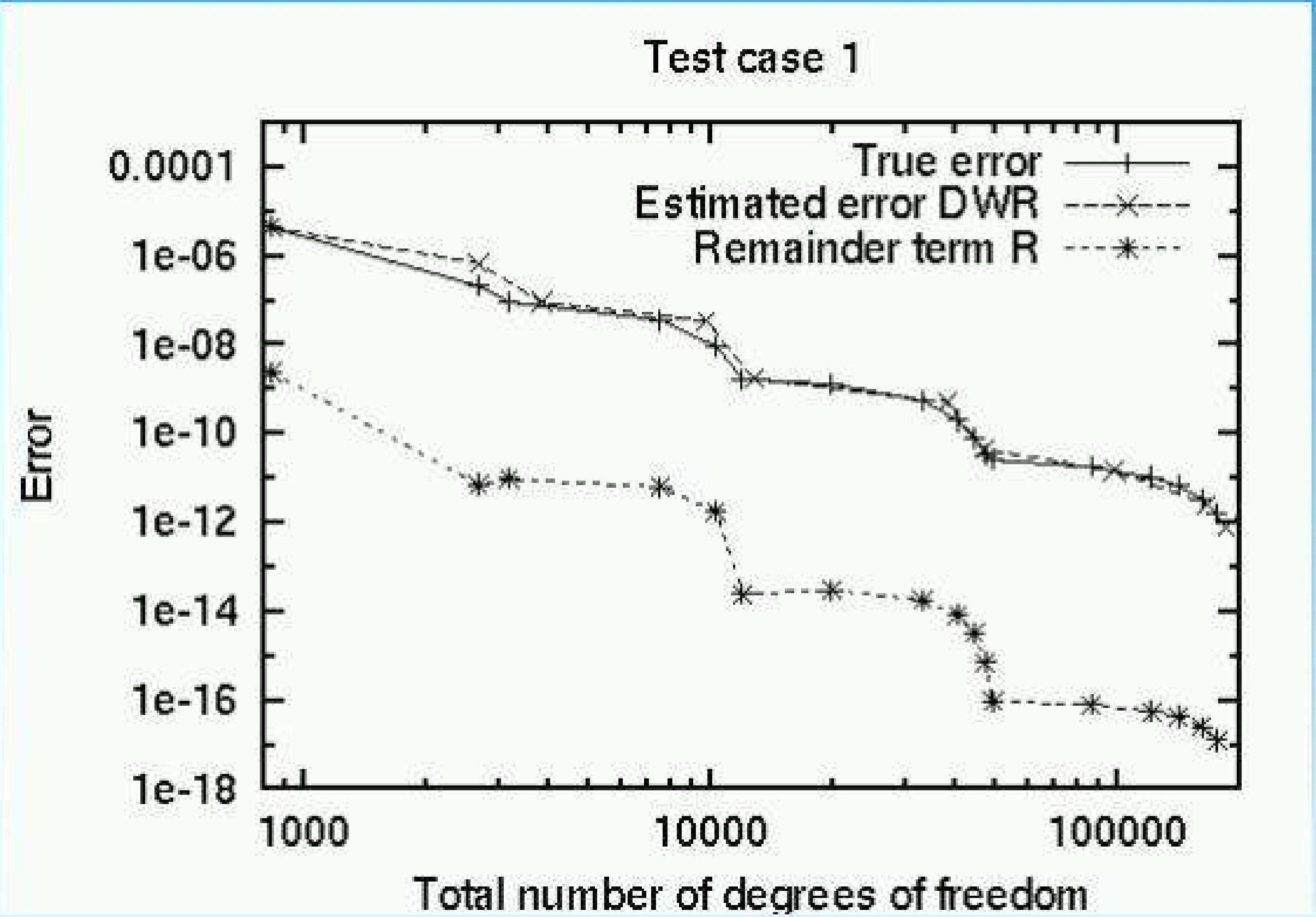
Exact displacement u



Recovered coefficient a

Example 1: Smooth Coefficient

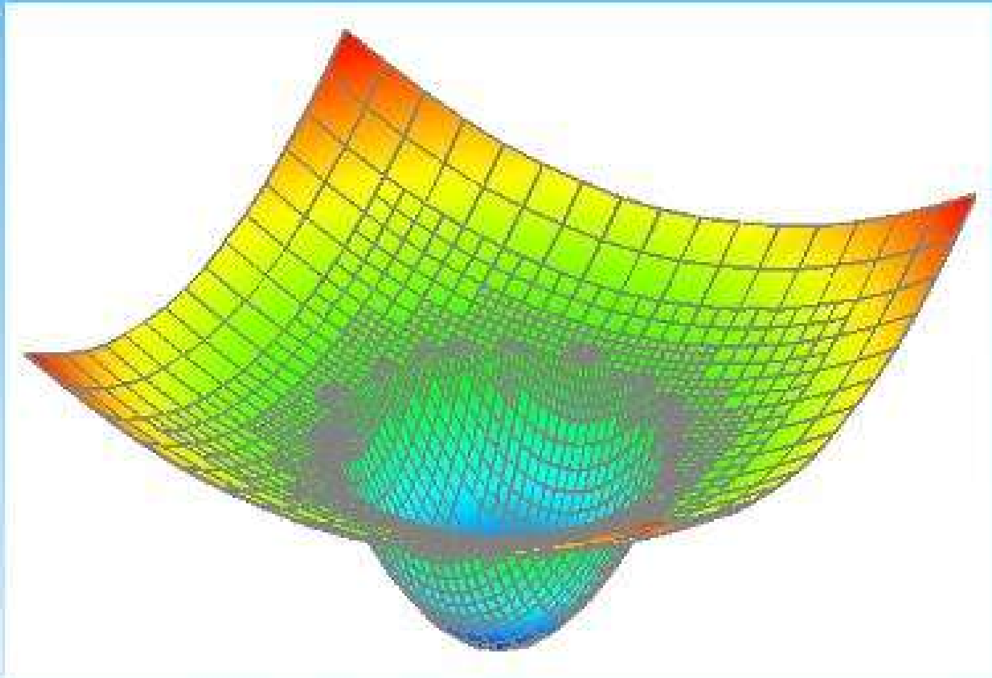
Quality of error estimates:



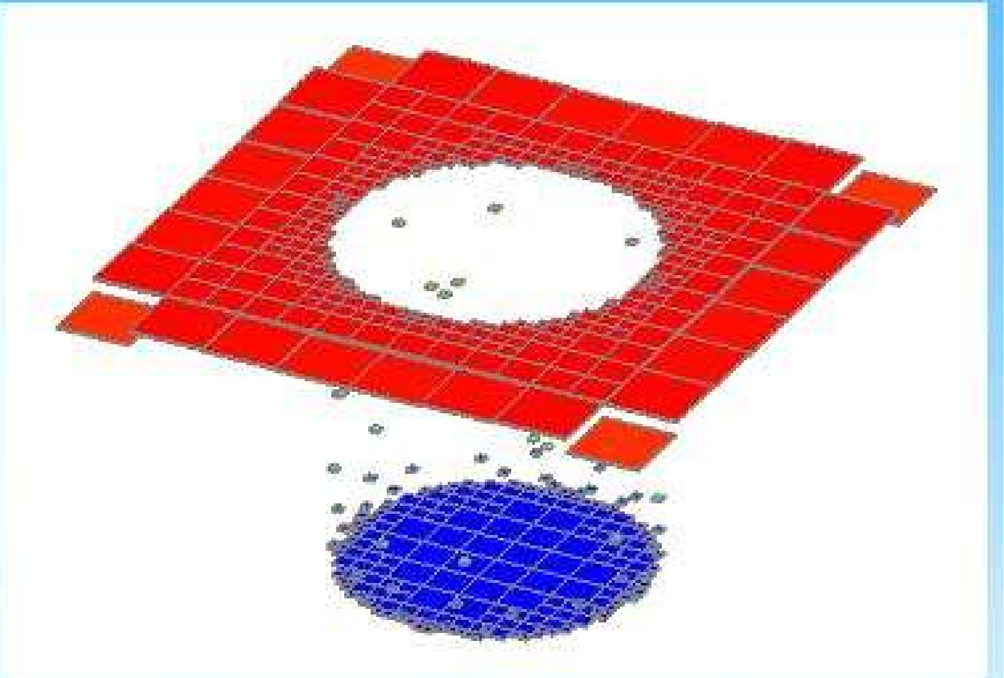
Example 2: Discontinuous Coefficient

Exact solution:

$$u = \begin{cases} r^2, & \text{if } r < \dfrac{1}{2} \\ r^2/8 + 7/32, & \text{otherwise} \end{cases}, \qquad a = \begin{cases} 1, \text{ if } r < \dfrac{1}{2} \\ 8, \text{otherwise} \end{cases}, \qquad z = u$$



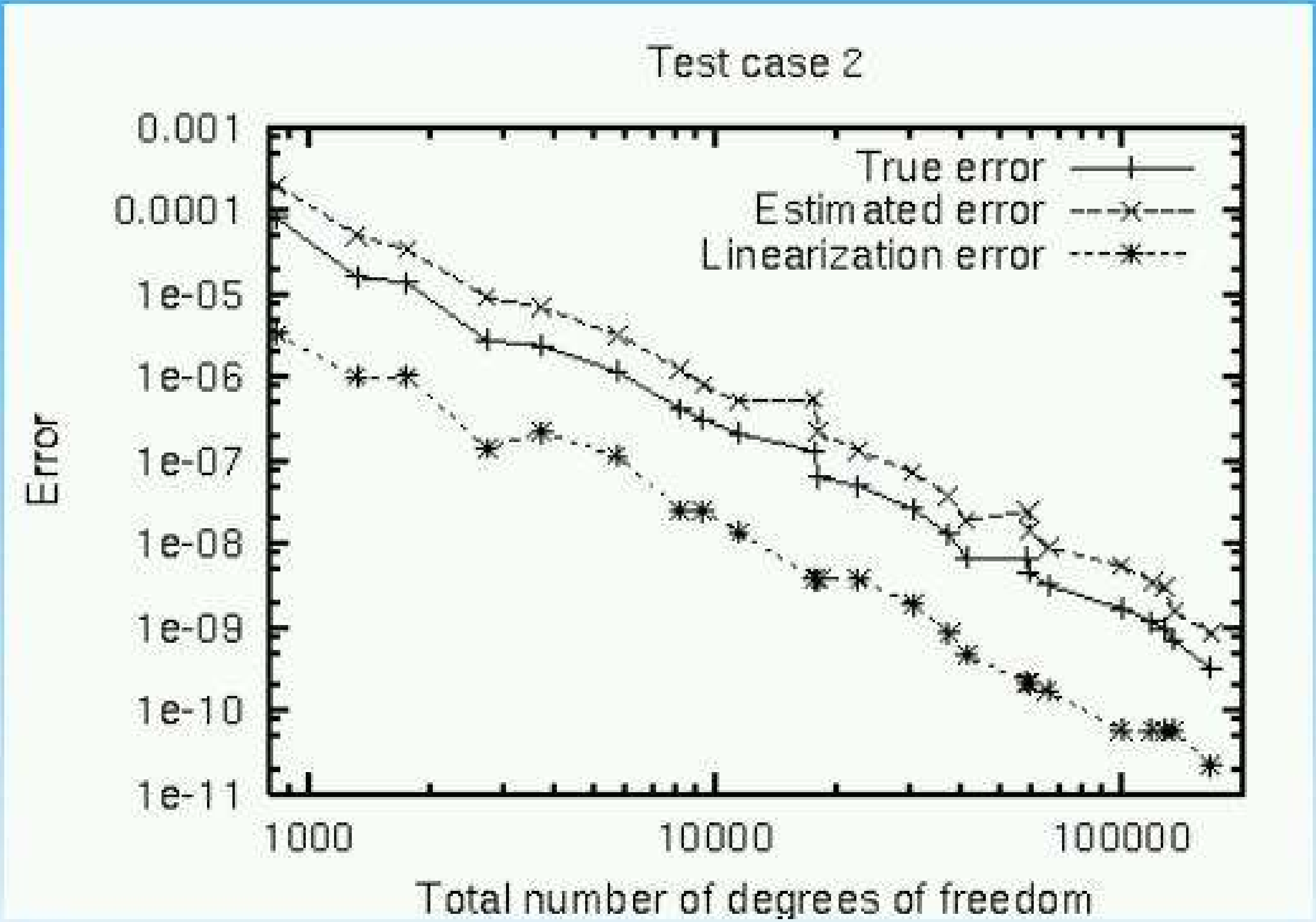
Exact displacement u



Recovered coefficient a

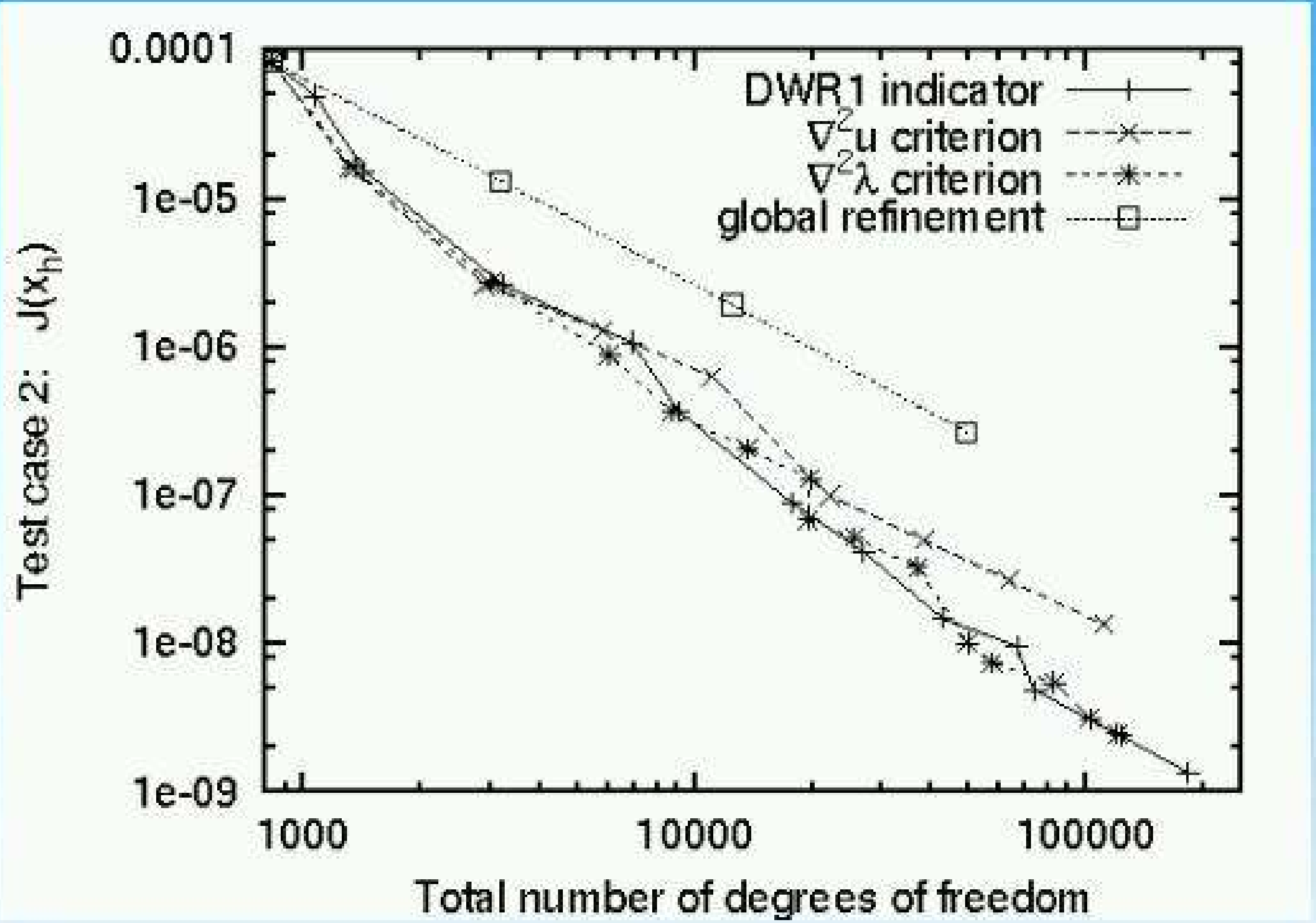
Example 2: Discontinuous Coefficient

Quality of error estimates:



Example 2: Discontinuous Coefficient

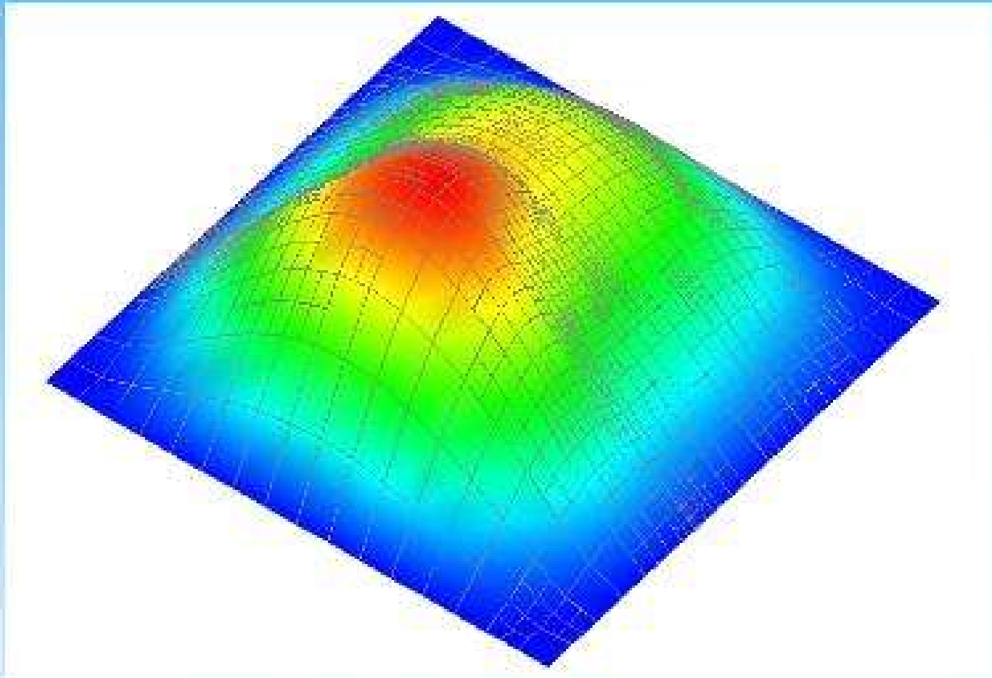
Comparison of different refinement criteria:



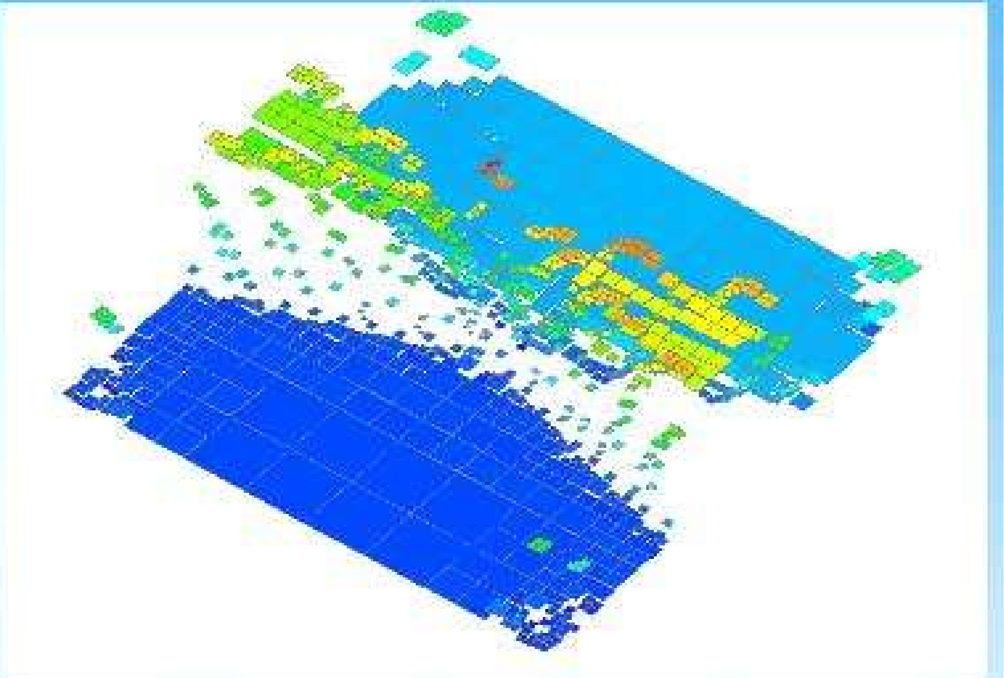
Example 3: Criss-cross Coefficient

Exact solution:

$$u = (-\nabla \cdot a \nabla)^{-1} f, \qquad a = \text{constant in 4 sectors}, \qquad z = u$$



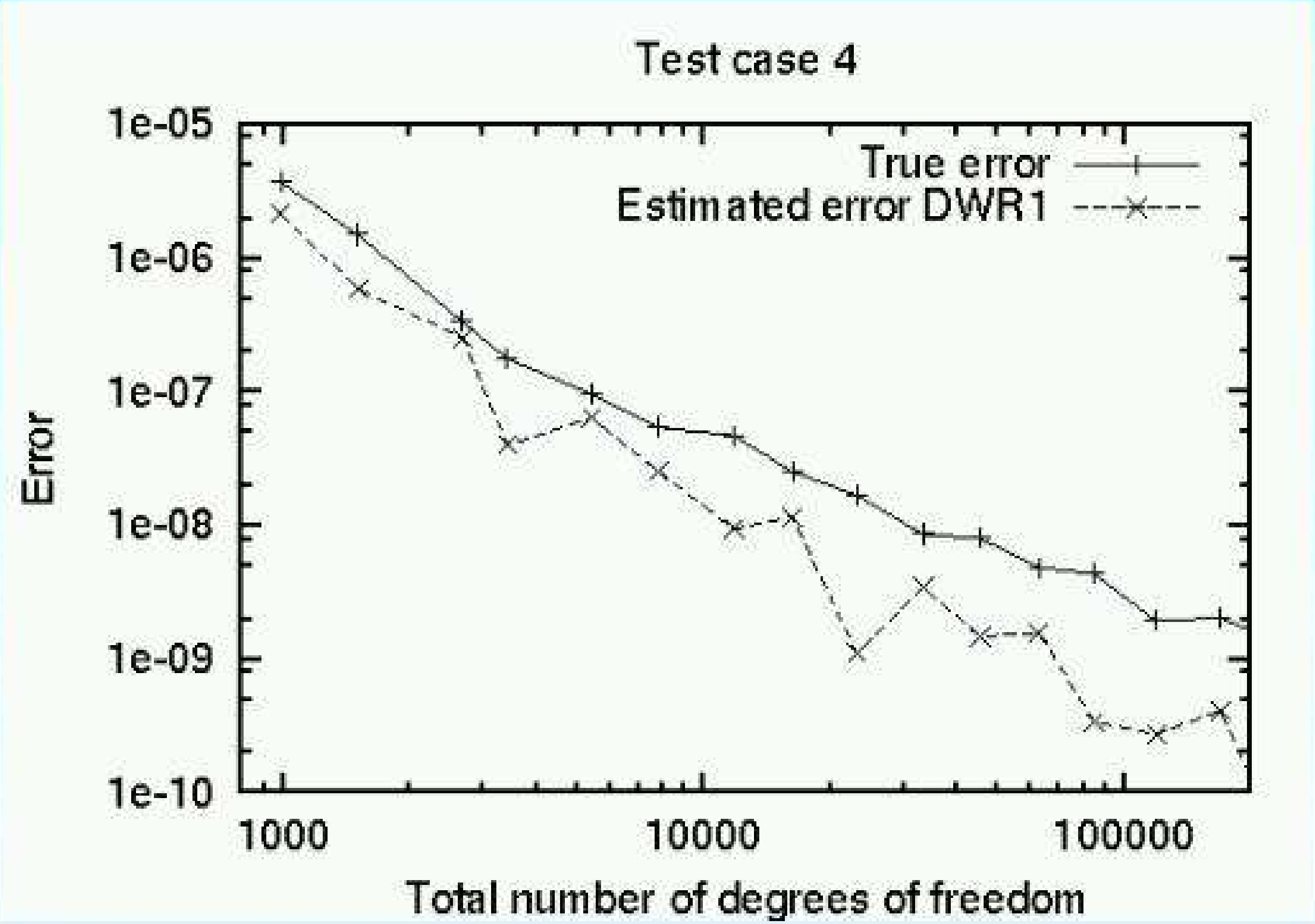
Exact displacement u



Recovered coefficient a

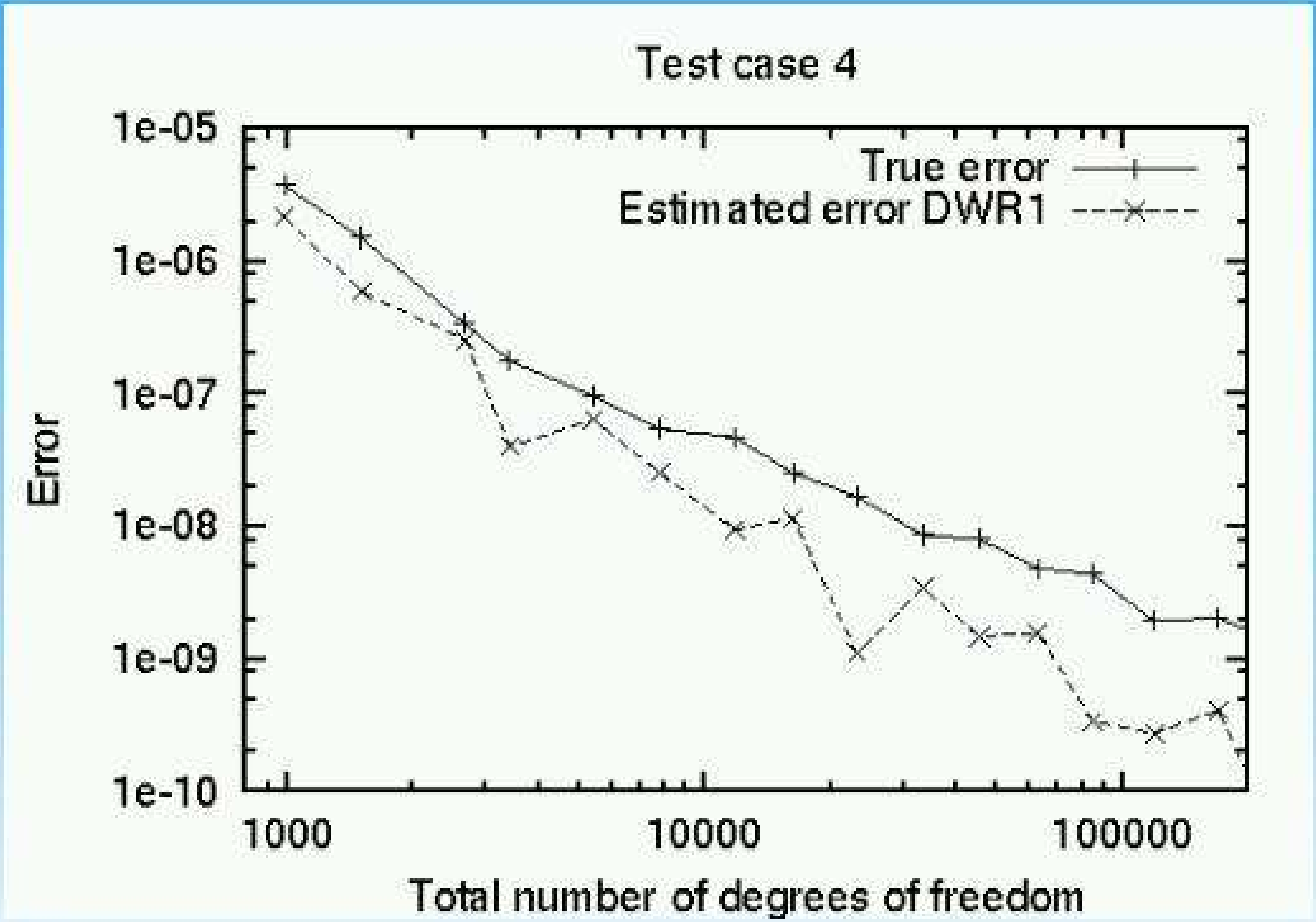
Example 3: Criss-cross Coefficient

Quality of error estimates:



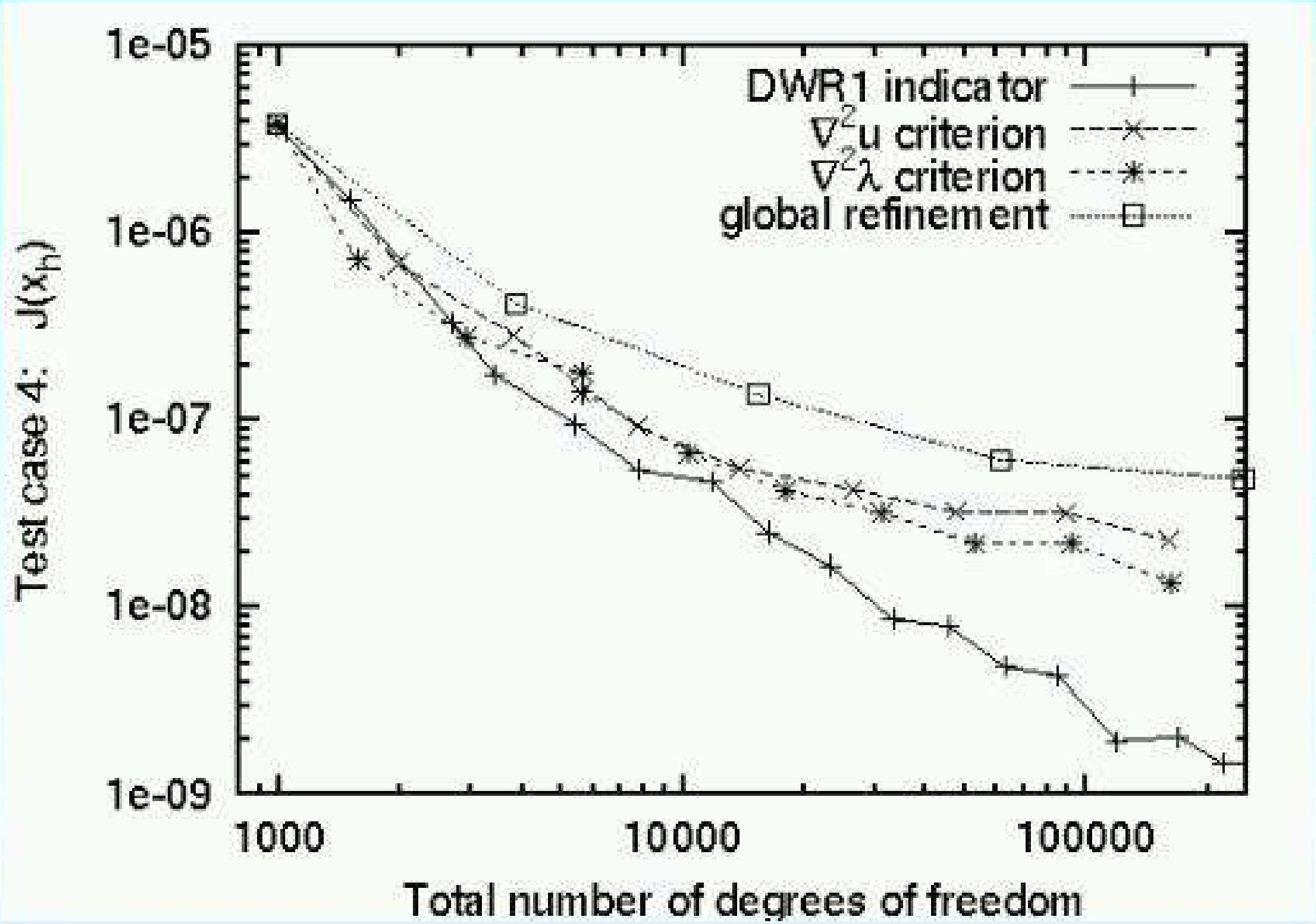
Example 3: Criss-cross Coefficient

Quality of error estimates:



Example 3: Criss-cross Coefficient

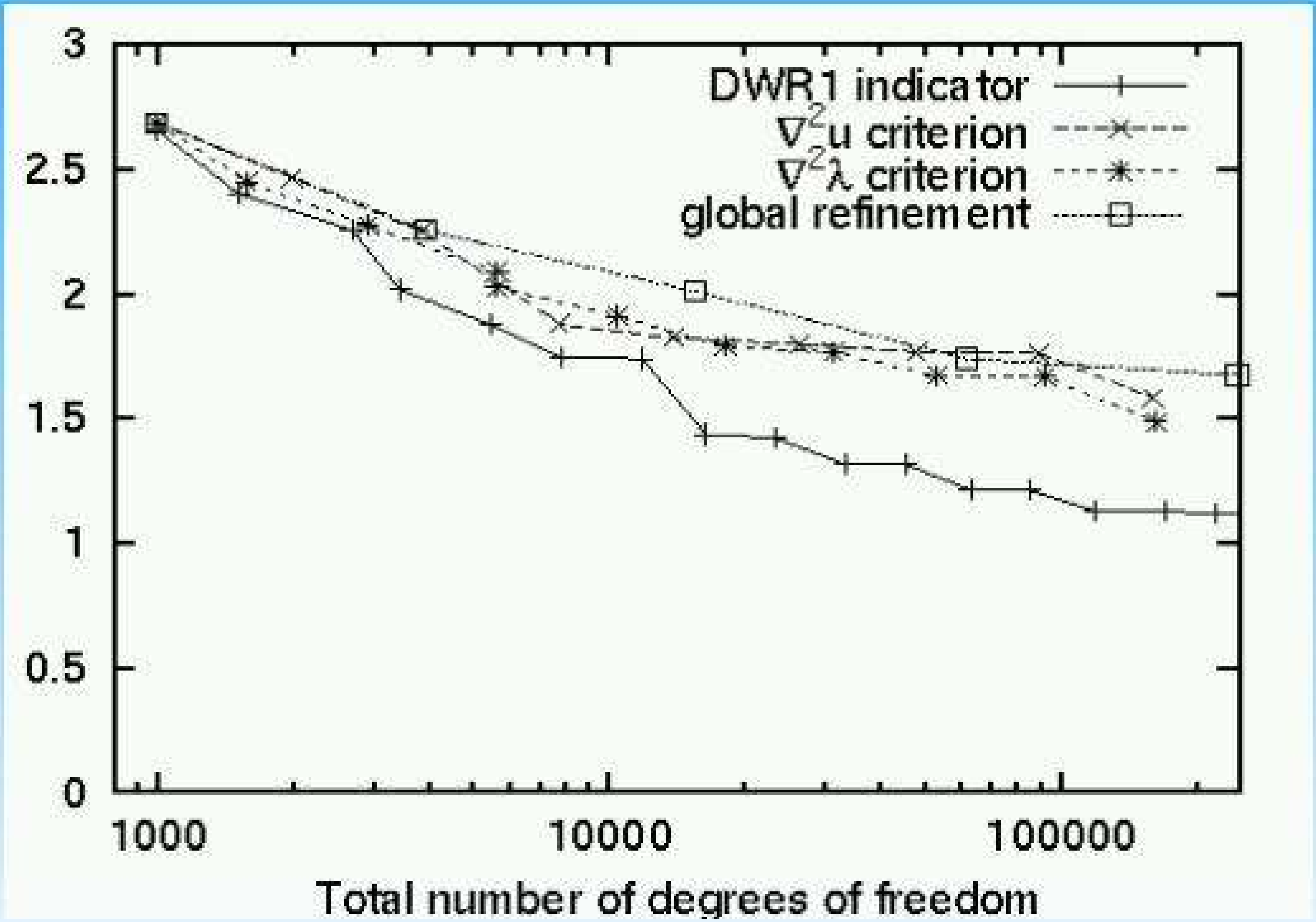
Comparison of different refinement criteria:



Example 3: Criss-cross Coefficient

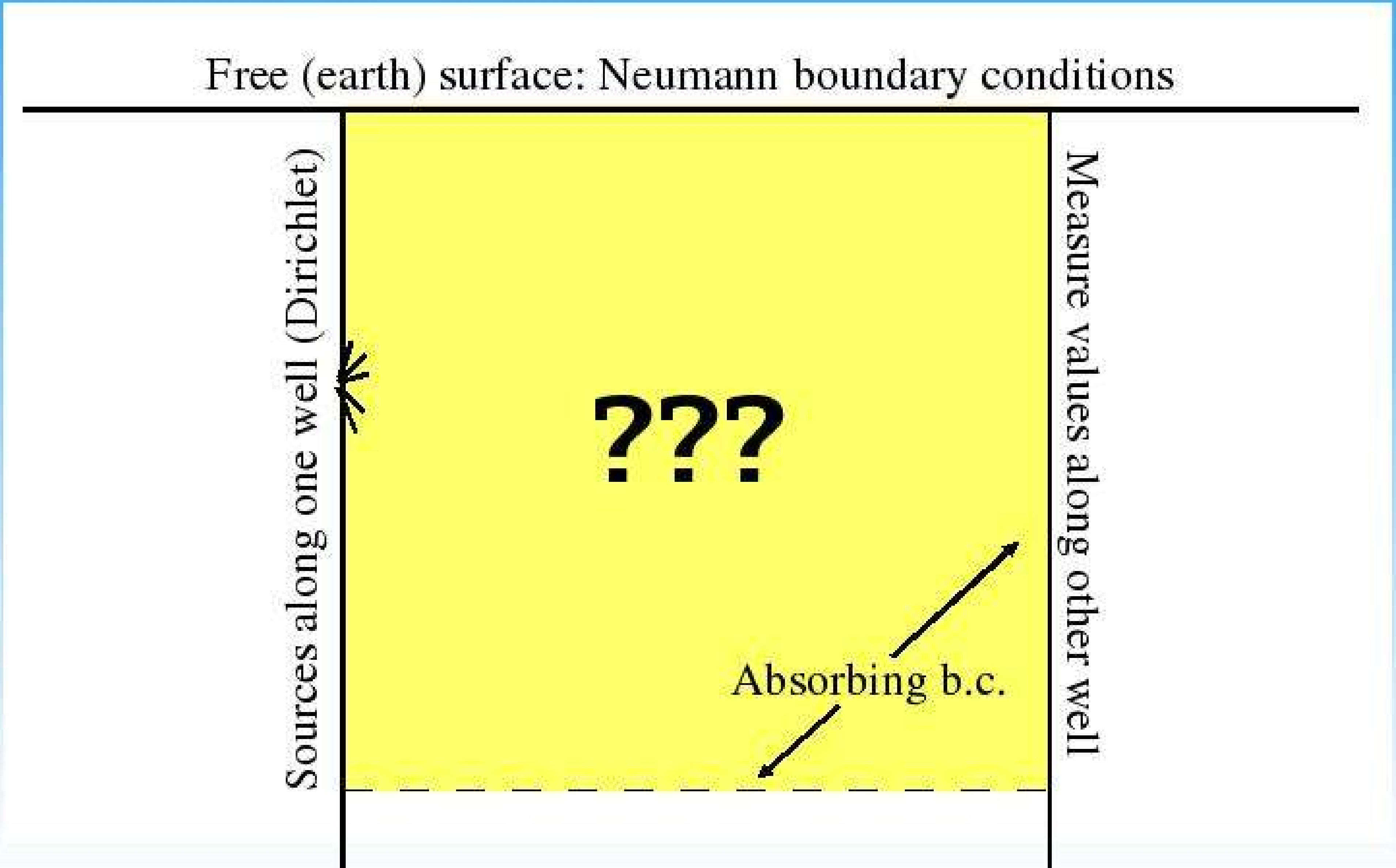
Comparison of different refinement criteria with respect to error in parameter:

$\|q - q_{\text{exact}}\|$



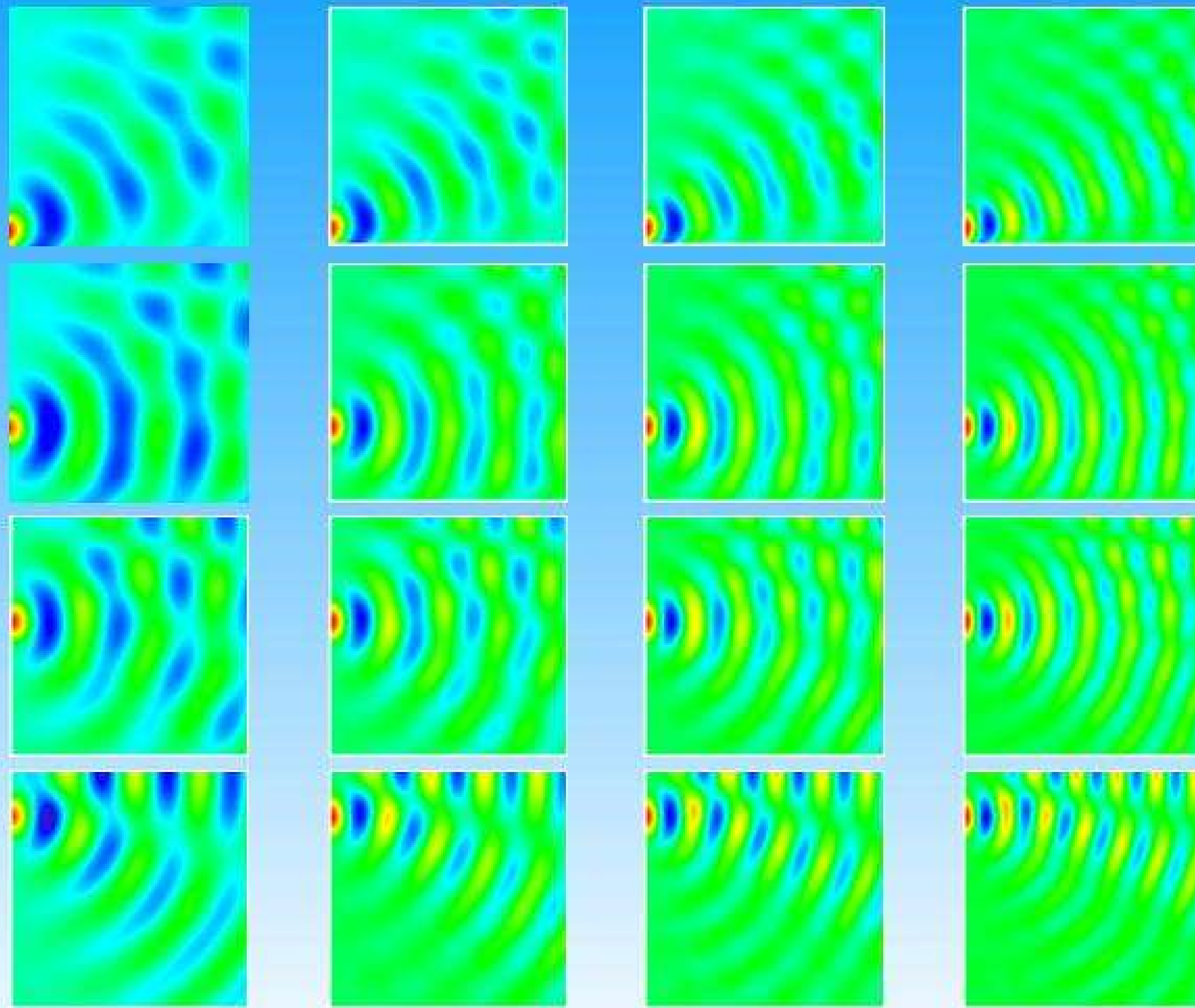
Helmholtz-Example: Layout

Abstraction of cross-well logging:



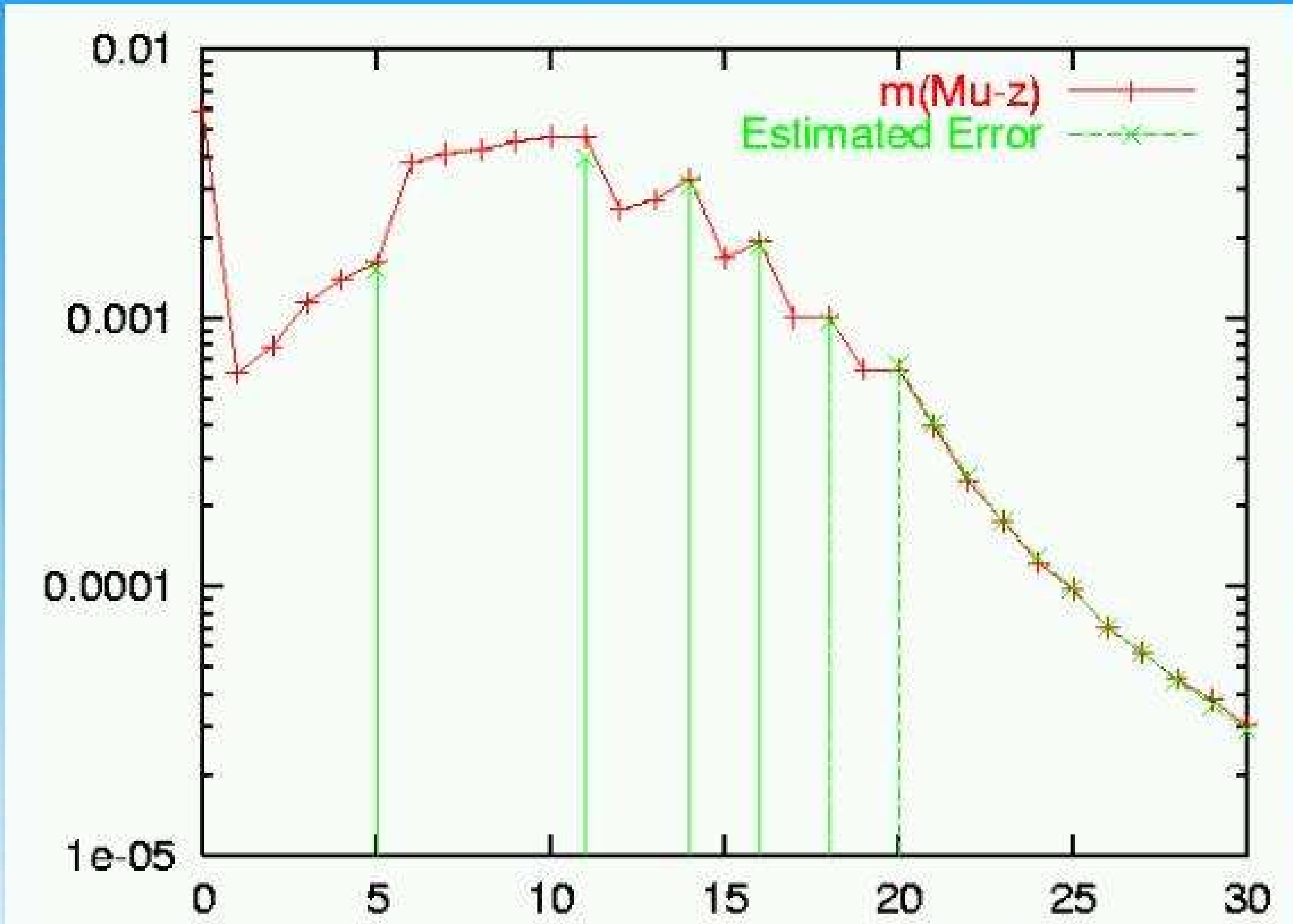
4 different frequencies: k^2=25, 30, 35 40

8 different source locations along left boundary



Total:
32 different
experiments

Helmholtz-Example: Error Estimation

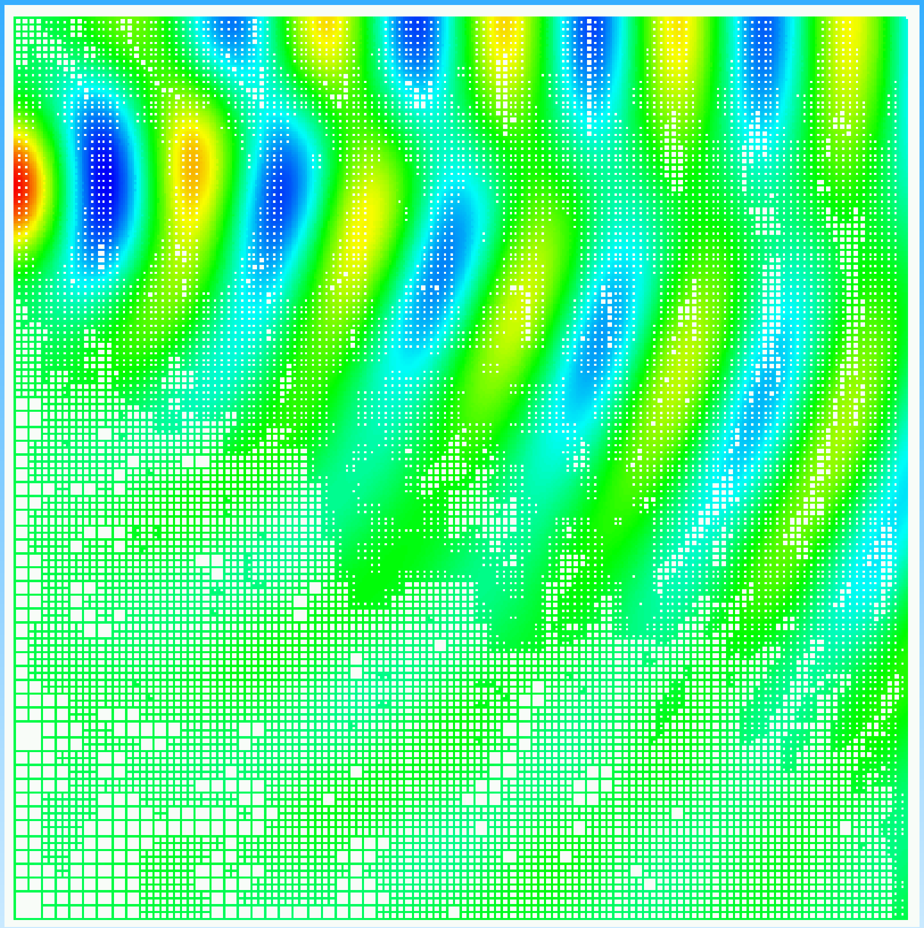


No noise:

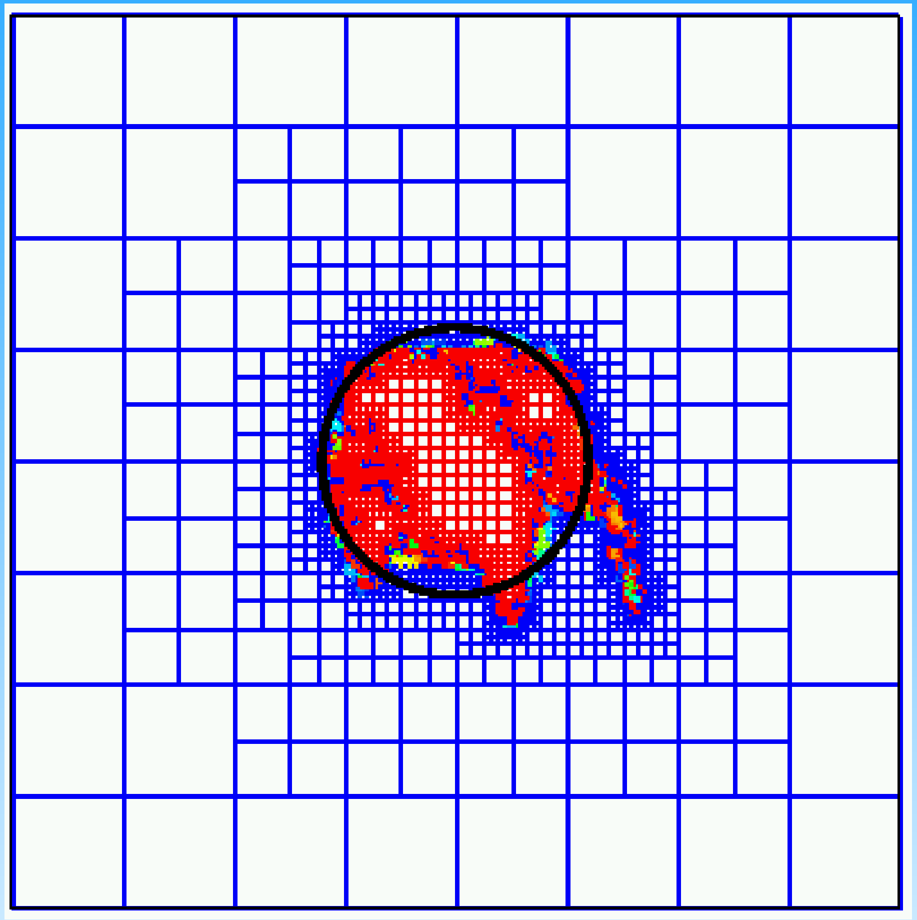
$$J(x) - J(x_h)$$
$$= -J(x_h)$$
$$= -m(Mu_h - z) - \beta r(q)$$

Estimated error astonishingly close to true error, so can be used as a stopping criterion: when error drops below noise level, then stop!

Helmholtz example: Grids



Grid for state/adjoint variables



Grid for coefficient

Summary

- Using duality arguments, error estimates for finite element discretizations can be formulated for arbitrary functionals of the solution
- These estimates do not need stability estimates, but compute stability properties using a dual solution
- Estimates are usually very accurate and yield very good adaptive meshes
- Estimates are even possible for equations that have very bad stability properties, such as inverse problems
- They nevertheless yield accurate and efficient estimates!