A model reduction approach to numerical inversion for a parabolic partial differential equation

Alexander V. Mamonov[1],
Liliana Borcea[2], Vladimir Druskin[3]
and Mikhail Zaslavsky[3]

[1]Schlumberger and the University of Texas at Austin (ICES)
[2]University of Michigan, Ann Arbor
[3]Schlumberger-Doll Research Center

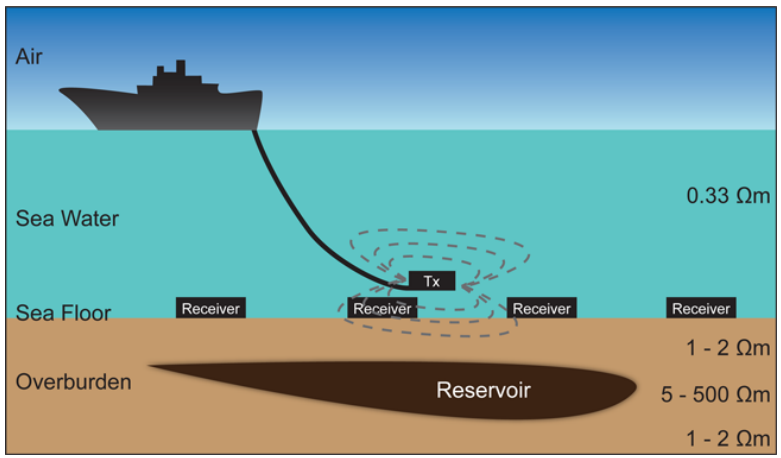
Support: NSF DMS-0934594.

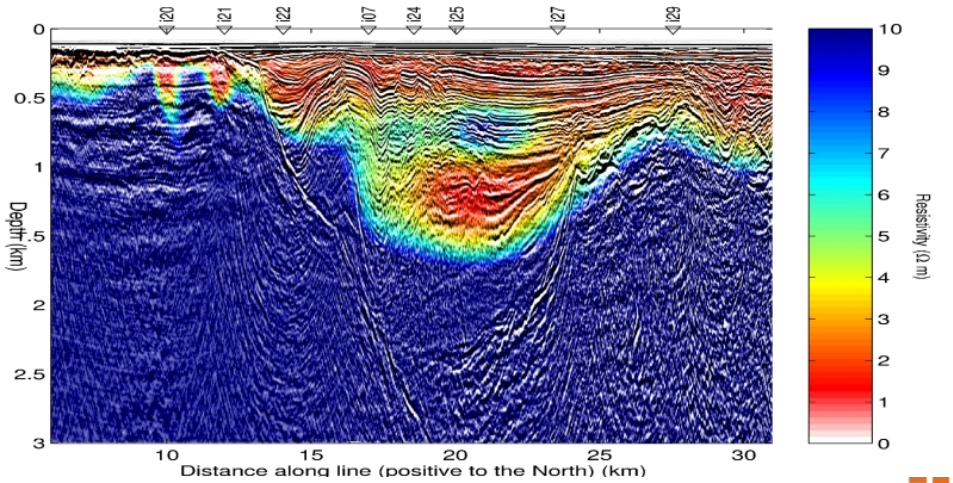
A model reduction approach to numerical inversion for a parabolic partial differential equation

Motivation

Problem: Time-domain Controlled Source Electromagnetic (CSEM) method in oil and gas exploration

- Determine the resistivity in the subsurface from time-resolved surface measurements
- Quasi-stationary parabolic Maxwell system
- Highly non-linear inverse problem, local minima, slow convergence
- Expensive forward modeling





Figures from: `http://www.acceleware.com`, `http://www.engineerlive.com/Mining-Engineer/CSEM`

Continuum problem

- Layered medium, one-dimensional equation

$$\frac{\partial}{\partial x}\left[r(x)\frac{\partial u(t,x)}{\partial x}\right] = \frac{\partial u(t,x)}{\partial t}, \quad x \in [0,1], \quad t > 0$$

- Boundary and initial conditions

$$u_x(t,0) = u(t,1) = 0, \quad u(0,x) = \delta(x)$$

- Measurements: boundary time-domain response

$$y(t) = u(t,0), \quad t > 0$$

- **Inverse problem**: given $y(t)$ for $t > 0$ find $r(x)$
- Inverse boundary value problems for parabolic (and elliptic) equations are typically ill-posed
- Ill-posedness in continuum leads to poor conditioning in numerics

Model reduction framework: semi-discrete system

- Discretize on a fine (uniform) grid in space with spacing $h = 1/(N+1)$

$$\frac{\partial \mathbf{u}(t)}{\partial t} = A(\mathbf{r})\mathbf{u}(t), \quad \mathbf{u}(0) = \frac{1}{h}\mathbf{e}_1 \tag{1}$$

- Discretization of the differential operator

$$A(\mathbf{r}) = -D^T \mathrm{diag}(\mathbf{r}) D, \quad \mathbf{r} \in \mathbb{R}^N_+$$

- Time-domain response

$$y(t;\mathbf{r}) = \mathbf{e}_1^T \mathbf{u}(t)$$

- **Semi-discrete inverse problem**: given $y(t;\mathbf{r})$ find $\mathbf{r} \in \mathbb{R}^N_+$
- Treat (1) as a dynamical system, apply model reduction techniques
- Model reduction: given $\mathbf{r}$ find a reduced model that approximates y
- **"Inverse" model reduction**: obtain a reduced model from the y, then find $\mathbf{r}$ which has this reduced model

Optimization formulation: non-linear preconditioning

- Noisy data
$$d(t) = y(t; \mathbf{r}^{\text{true}}) + \xi(t), \quad t > 0,$$
where $\xi(t)$ is due to noise and discretization errors.
- **Optimization formulation**
$$\mathbf{r}^\star = \arg\min_{\mathbf{r}\in\mathcal{S}} \frac{1}{2} \|\mathcal{Q}(d(t)) - \mathcal{Q}(y(t;\mathbf{r}))\|_2^2 + \frac{\alpha}{2}\mathcal{P}(\mathbf{r})$$
- Solution method: **Gauss-Newton** or non-linear CG
- Traditional approach: output least squares, no preconditioner $\mathcal{Q}$, regularization term $\mathcal{P}$ Tikhonov or TV
- Drawbacks: non-convexity, easy to get stuck in local minima, slow convergence, difficulties with high contrast, expensive
- **Non-linear preconditioner** $\mathcal{Q} : C(0, +\infty) \to \mathbb{R}^q$ with small q, the dimension of parameters of reduced models
- The mapping $\mathcal{Q}(y(\,\cdot\,;\mathbf{r})) : \mathbb{R}^N \to \mathbb{R}^q$ is an **approximate identity**

Projection-based model reduction

- Transfer function of the full model $A(\mathbf{r}) \in \mathbb{R}^{N\times N}$, $\mathbf{b} \in \mathbb{R}^N$

$$G(s;\mathbf{r}) = \int_0^{+\infty} y(t;\mathbf{r})e^{-st}dt = \mathbf{b}^T(sI - A(\mathbf{r}))^{-1}\mathbf{b}, \quad s > 0, \quad \mathbf{b} = \frac{1}{\sqrt{h}}\mathbf{e}_1$$

- Transfer function of a reduced model $A_m \in \mathbb{R}^{m\times m}$, $\mathbf{b}_m \in \mathbb{R}^m$

$$G_m(s) = \mathbf{b}_m^T(sI_m - A_m)^{-1}\mathbf{b}_m$$

- Projection-based model reduction

$$A_m = V^T AV, \quad \mathbf{b}_m = V^T\mathbf{b}, \quad V^T V = I_m$$

- Columns of $V \in \mathbb{R}^{N\times m}$ span the projection **subspace**
- Choice of subspace is dictated by **matching conditions**

$$\left.\frac{\partial^k G_m}{\partial s^k}\right|_{s=\sigma_j} = \left.\frac{\partial^k G}{\partial s^k}\right|_{s=\sigma_j}, \quad j = 1,\ldots,m, \quad k = 0,\ldots,2M_j - 1$$

at **interpolation nodes** $\sigma_j \in [0,+\infty)$

Rational Krylov Model Reduction

- Reduced order transfer function admits a partial fraction expansion

$$G_m(s) = \sum_{j=1}^{m} \frac{c_j}{s+\theta_j}, \quad c_j > 0, \quad \theta_j > 0,$$

  with negative **poles** $-\theta_j$ and positive **residues** c_j
- Rational G_m, hence **rational interpolation**
- Typical choices of projection subspaces in model reduction: rational **Krylov** subspaces

$$\mathcal{K}_m(\boldsymbol{\sigma}) = \text{span}\left\{(\sigma_j I - A)^{-k}\mathbf{b} \mid j = 1,\ldots,m;\ k = 1,\ldots,M_j\right\}$$

- Popular special cases for **forward** modeling: moment matching

$$\begin{aligned} \mathcal{K}_m(+\infty) &= \text{span}\left\{\mathbf{b}, A\mathbf{b}, \ldots, A^{m-1}\mathbf{b}\right\} \\ \mathcal{K}_m(0) &= \text{span}\left\{A^{-1}\mathbf{b}, A^{-2}\mathbf{b}, \ldots, A^{-m}\mathbf{b}\right\} \end{aligned}$$

- $K_m(+\infty)$ is bad for inversion

Inversion via model reduction: $\mathcal{H}_2$-optimality

- Proposed in [Druskin, Simoncini, Zaslavsky, 2011]
- Based on $\mathcal{H}_2$-optimal reduced models
- View reduced model as a function of interpolation nodes $\boldsymbol{\sigma}$

$$y_m(t;\boldsymbol{\sigma}) = \mathbf{b}^T V(\boldsymbol{\sigma}) e^{A_m(\boldsymbol{\sigma})t} V(\boldsymbol{\sigma})^T \mathbf{b}, \quad \text{for} \quad A_m(\boldsymbol{\sigma}) = V(\boldsymbol{\sigma})^T A(\mathbf{r}) V(\boldsymbol{\sigma})$$

- Minimize time-domain error in L_2 sense

$$\boldsymbol{\sigma}^\star = \arg\min \| y(t;\mathbf{r}) - y_m(t;\boldsymbol{\sigma}) \|_{L_2[0,+\infty)}$$

- Equivalent to $\mathcal{H}_2$-optimality in Laplace domain (solve with IRKA)

$$\boldsymbol{\sigma}^\star = \arg\min \| G(s;\mathbf{r}) - G_m(s) \|_{\mathcal{H}_2}$$

- Use poles and residues of $G_m(s)$ as parameters in inversion
- Define the non-linear preconditioner with a **chain of mappings**

$$\mathcal{Q}_{\mathcal{H}_2}(y(\,\cdot\,;\mathbf{r})): \ \mathbf{r} \overset{\text{(a)}}{\to} A(\mathbf{r}) \overset{\text{(b)}}{\to} \boldsymbol{\sigma}^\star \overset{\text{(c)}}{\to} V(\boldsymbol{\sigma}^\star) \overset{\text{(d)}}{\to} A_m \overset{\text{(e)}}{\to} \{(c_j, \theta_j)\}_{j=1}^m$$

Inversion via model reduction: continued fraction

- We propose a different choice of parameters and matching conditions
- Write the reduced model response as a **continued fraction**

$$G_m(s) = \cfrac{1}{\widehat{\kappa}_1 s + \cfrac{1}{\kappa_1 + \cfrac{1}{\ddots + \cfrac{1}{\widehat{\kappa}_m s + \cfrac{1}{\kappa_m}}}}}$$

- This is a response $w_1(s)$ of a second-order finite difference scheme

$$\frac{1}{\widehat{\kappa}_j}\left(\frac{w_{j+1}-w_j}{\kappa_j} - \frac{w_j - w_{j-1}}{\kappa_{j-1}}\right) - s w_j = 0$$

- Continued fraction coefficient are **discrete resistivities**
- Use $\{(\kappa_j, \widehat{\kappa}_j)\}_{j=1}^m$ as parameters in inversion, define $\mathcal{Q}$ as a chain

$$\mathcal{Q}(y(\,\cdot\,;\mathbf{r})):\ \mathbf{r} \overset{\text{(a)}}{\to} A(\mathbf{r}) \overset{\text{(b)}}{\to} V \overset{\text{(c)}}{\to} A_m \overset{\text{(d)}}{\to} \{(c_j,\theta_j)\}_{j=1}^m \overset{\text{(e)}}{\to} \{(\kappa_j,\widehat{\kappa}_j)\}_{j=1}^m$$

Matching conditions for inversion

- The choice of matching conditions is important: what is good for forward modeling is not necessarily good for inversion
- Use the Jacobian of $\mathcal{Q}$ to quantify the quality of matching conditions

$$(\mathcal{D}\mathcal{Q})_{j,k} = \begin{cases} \dfrac{\partial \kappa_j}{\partial r_k}, & \text{for } j = 1, \ldots, m \\ \dfrac{\partial \widehat{\kappa}_j}{\partial r_k}, & \text{for } j = m+1, \ldots, 2m \end{cases}, \quad k = 1, \ldots, N.$$

- Proper matching conditions should give good **conditioning** and **resolution**
- Good conditioning of $\mathcal{D}\mathcal{Q}$ is desirable for fast and robust performance of Gauss-Newton iteration, possible to achieve $\text{cond}(\mathcal{D}\mathcal{Q}) \approx O(1)$
- Conditioning of $\mathcal{Q}(d(\,\cdot\,))$ always grows exponentially (unavoidable ill-posedness), slower growth is preferential
- Resolution is connected to **optimal grids**

Connection to optimal grids

- Optimal (spectrally matched) grids were introduced in [Druskin, Knizhnerman, 2000] for exponential superconvergence of DtN maps
- Typically defined for $r(x) \equiv 1$

$$(\kappa_j^{(0)}, \widehat{\kappa}_j^{(0)})_{j=1}^m = \mathcal{Q}(y(\,\cdot\,, \mathbf{r}^{(0)})), \quad \mathbf{r}^{(0)} = (1, 1, \ldots, 1)^T$$

- Continued fraction coefficients are the grid steps of FD scheme for

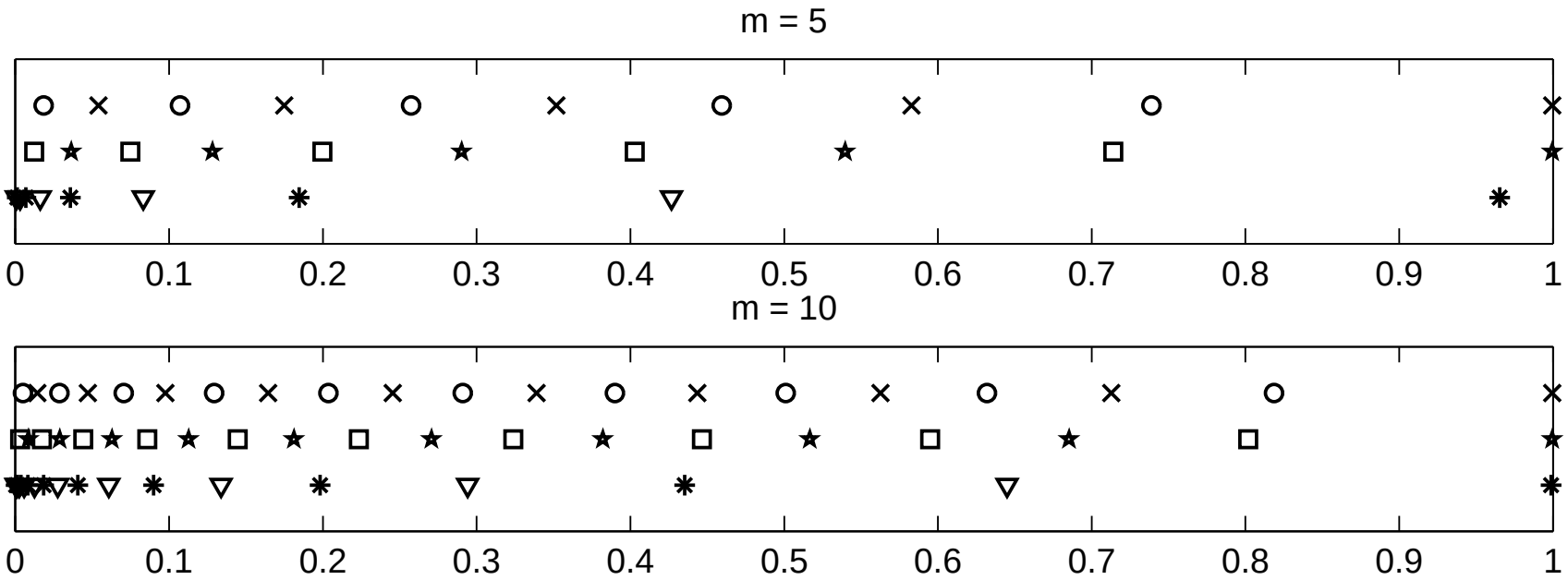
$$\Delta w - sw = 0$$

- Staggered grid with primary and dual nodes

$$x_j^{(0)} = \sum_{k=1}^{j} \kappa_k^{(0)}, \quad \widehat{x}_j^{(0)} = \sum_{k=1}^{j} \widehat{\kappa}_k^{(0)}, \quad j = 1, \ldots, m$$

- Optimal grids were used for inversion (Sturm-Liouville, EIT)
- Here we do not explicitly use optimal grids for inversion, but to quantify the resolution

Optimal grids for different matching conditions



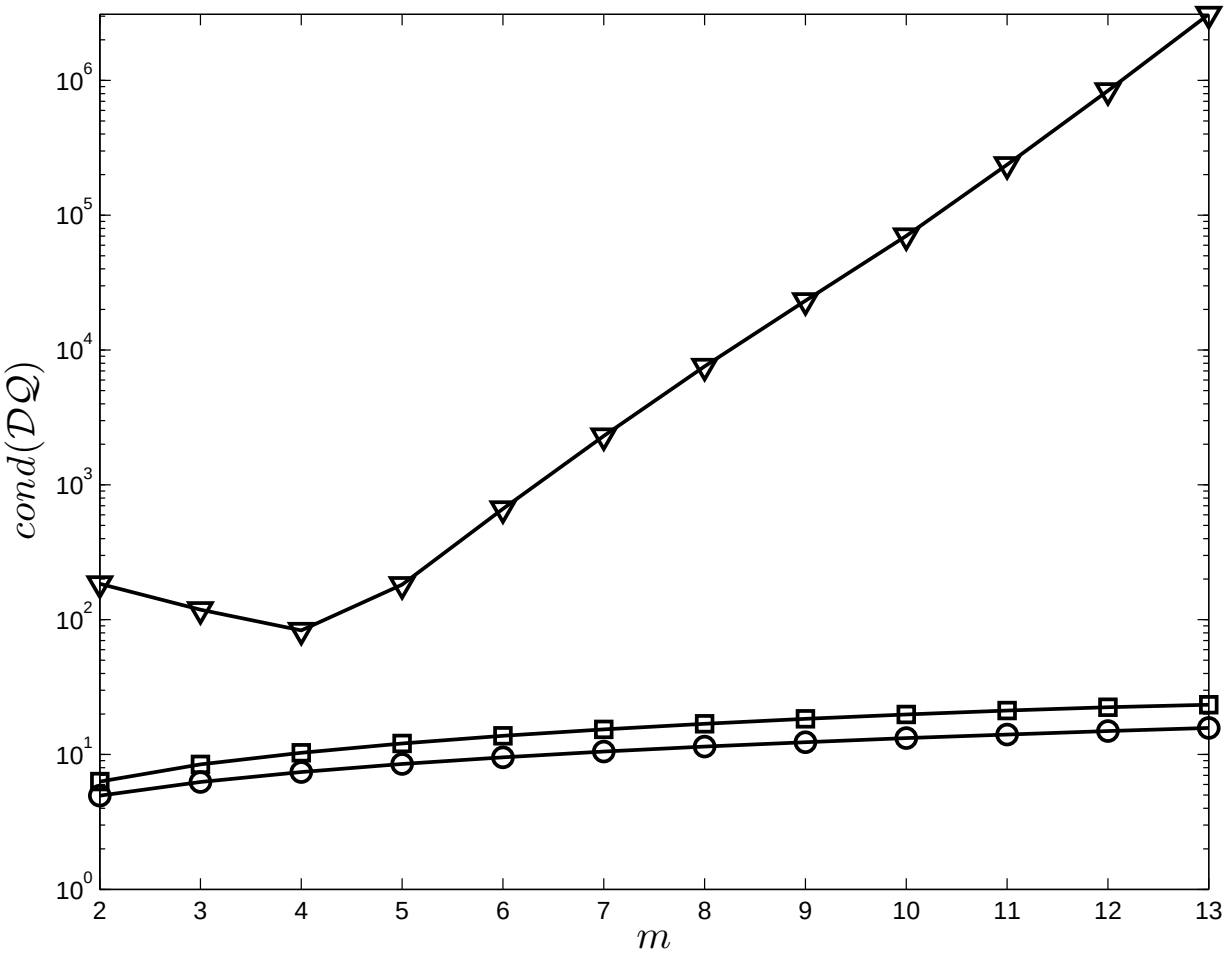
Compare three choices of matching conditions:

$(\circ, \times)$ Moment matching at zero, $\mathcal{K}_m(0) = \text{span}\left\{A^{-1}\mathbf{b}, A^{-2}\mathbf{b}, \ldots, A^{-m}\mathbf{b}\right\}$

$(\square, \star)$ Interpolation $\mathcal{K}_m(\widetilde{\boldsymbol{\sigma}}) = \text{span}\left\{(\widetilde{\sigma}_j I - A)^{-1}\mathbf{b} \mid j = 1, \ldots, m\right\}$ at geometrically spaced nodes $\widetilde{\sigma}_j = \widetilde{\sigma}_1 \left(1 + C/m\right)^{j-1}$

$(*, \bigtriangledown)$ Interpolation at fast growing nodes $\boldsymbol{\sigma}^\star$

Conditioning of the Jacobian



- Rows of $\mathcal{DQ}$ have the meaning of **sensitivity functions**
- Each row corresponds to one grid cell
- Sensitivity functions are localized
- Peak locations are at the optimal grid nodes
- Clustered nodes lead to (almost) linearly dependent rows of $\mathcal{DQ}$

Condition number growth for moment matching at zero ($\circ$), interpolation at $\widetilde{\sigma}$ ($\square$), interpolation at $\sigma^\star$ ($\bigtriangledown$)

Conditioning of rational approximation

- Optimization objective requires computing

$$(\kappa_j, \widehat{\kappa}_j)_{j=1}^{m} = \mathcal{Q}(d(\,\cdot\,))$$

- The most **unstable** step in inversion procedure
- Rational approximation (interpolation) problem
- **Simple Padé** for moment matching $\mathcal{K}_m(0)$

$$\left.\frac{\partial^j G_m}{\partial s^j}\right|_{s=0} = \left.\frac{\partial^j G}{\partial s^j}\right|_{s=0} = (-1)^j \int_0^{+\infty} y(t;\mathbf{r}) t^j dt, \quad j = 0, 1, \ldots, 2m-1$$

- **Multipoint Padé** for osculatory interpolation $\mathcal{K}_m(\widetilde{\sigma})$

$$G_m(s) = \frac{p(s)}{q(s)}, \quad \begin{cases} p(\widetilde{\sigma}_j) - G_m(\widetilde{\sigma}_j) q(\widetilde{\sigma}_j) = 0 \\ p'(\widetilde{\sigma}_j) - G_m'(\widetilde{\sigma}_j) q(\widetilde{\sigma}_j) - G_m(\widetilde{\sigma}_j) q'(\widetilde{\sigma}_j) = 0 \end{cases},$$

  for $j = 1, \ldots, m$

- Once the coefficients of $p(s)$ and $q(s)$ are known, can compute the poles, residues and the continued fraction coefficients easily

Conditioning of rational approximation: comparison

m	2	3	4	5	6
cond(T_m)	$5.28 \cdot 10^1$	$1.26 \cdot 10^5$	$1.84 \cdot 10^9$	$9.14 \cdot 10^{13}$	$2.86 \cdot 10^{16}$
cond(R_m)	$4.43 \cdot 10^2$	$6.73 \cdot 10^4$	$1.85 \cdot 10^7$	$6.95 \cdot 10^9$	$3.83 \cdot 10^{12}$

- Both simple and multipoint Padé problems may be solved via SVD
- Simple Padé: SVD of **Toeplitz** matrix $T_m \in \mathbb{R}^{m\times(m+1)}$ of Taylor coefficients τ_j, $j = 1, \ldots, 2m-1$ of

$$G(s) = \tau_0 + \tau_1 s + \tau_2 s^2 + \ldots + \tau_{2m-1} s^{2m-1} + \ldots$$

- Multipoint Padé: SVD of $R_m \in \mathbb{R}^{2m\times(2m+1)}$

$$R_m = \begin{bmatrix} \Sigma_{1:m,\,1:m} & -F\Sigma \\ \Sigma'_{1:m,\,1:m} & -F'\Sigma - F\Sigma' \end{bmatrix},$$

  with **Vandermonde** Σ and $F = \mathrm{diag}(G(\widetilde{\sigma}))$
- Multipoint Padé is clearly **superior**

Computing $\mathcal{Q}$ and $\mathcal{D}\mathcal{Q}$: chain of mappings

- Chain of mappings for computing $\mathcal{Q}$ and the Jacobian $\mathcal{D}\mathcal{Q}$

$$\mathcal{Q}(y(\,\cdot\,;\mathbf{r})):\ \mathbf{r} \overset{\text{(a)}}{\to} A(\mathbf{r}) \overset{\text{(b)}}{\to} V \overset{\text{(c)}}{\to} A_m \overset{\text{(d)}}{\to} \{(c_j,\theta_j)\}_{j=1}^m \overset{\text{(e)}}{\to} \{(\kappa_j,\widehat{\kappa}_j)\}_{j=1}^m$$

(a) $A(\mathbf{r}) = -D^T \mathrm{diag}(\mathbf{r}) D$, trivial to differentiate.

(b) Requires differentiation of orthonormal basis V.
Here we differentiate QR decomposition.

(c) $A_m = V^T A V$, trivial to differentiate once $\mathcal{D}V$ is known.

(d) Differentiation of eigendecomposition

$$c_j = (\mathbf{b}_m^T \mathbf{z}_j)^2, \quad A_m \mathbf{z}_j + \theta_j \mathbf{z}_j = 0, \quad \|\mathbf{z}_j\| = 1, \quad j = 1,\ldots,m$$

(e) Lanczos iteration. Explicit differentiation formulas derived in [Borcea, Druskin, Knizhnerman, 2005]

- Steps (d)–(e) can be combined into one using a variant of Lanczos. No explicit formulas, requires differentiation of the iteration.

Differentiation of QR decomposition

- Krylov matrix $K = \left[(\widetilde{\sigma}_1 I - A)^{-1}\mathbf{b}, \ldots, (\widetilde{\sigma}_m I - A)^{-1}\mathbf{b}\right] \in \mathbb{R}^{N\times m}$
- Compute V from QR decomposition: $K = VR, \quad K^T K = R^T R = LL^T$
- Need to differentiate Cholesky to get

$$\frac{\partial V}{\partial r_k} = \left(\frac{\partial K}{\partial r_k} - V\frac{\partial L^T}{\partial r_k}\right) L^{-T}, \quad k = 1, \ldots, N$$

Proposition (Differentiation of Cholesky factorization)

Let $M \in \mathbb{R}^{n\times n}$ be a matrix with Cholesky factorization $M = LL^T$. Given the perturbation δM of M, the corresponding perturbation δL of the Cholesky factor is computed by the following algorithm.

For $k = 1, \ldots, n$

$$\delta L_{kk} = \frac{1}{L_{kk}}\left(\frac{\delta M_{kk}}{2} - \sum_{j=1}^{k-1} \delta L_{kj} L_{kj}\right)$$

For $i = k+1, \ldots, n$

$$\delta L_{ik} = \frac{1}{L_{kk}}\left(\delta M_{ik} - \sum_{j=1}^{k} \delta L_{kj} L_{ij} - \sum_{j=1}^{k-1} \delta L_{ij} L_{kj}\right)$$

Regularization

- Traditional approaches require regularization for stability
- We use regularization only to improve the reconstruction quality
- **Separate** the fitting $\|\mathcal{Q}(d(\,\cdot\,)) - \mathcal{Q}(y(\,\cdot\,;\mathbf{r})\|_2$ step from regularization step
- Jacobian $\mathcal{D}\mathcal{Q} \in \mathbb{R}^{2m\times N}$ has a **large null space** $2m \ll N$
- Minimze the regularization functional $\mathcal{P}(\mathbf{r})$ in null$(\mathcal{D}\mathcal{Q})$
- Weighted discrete H^1 seminorm

$$\mathcal{P}(\mathbf{r}) = \frac{1}{2}\|W^{1/2}\Delta\mathbf{r}\|_2^2,$$

  here Δ is the truncation of D (first derivative, not second)
- Works for both smooth ($W = I$) and piecewise constant resistivities (non-linear re-weighting at every iteration)
- Constrained optimization subproblem at each Gauss-Newton iteration

$$\begin{aligned} &\text{minimize} && \boldsymbol{\rho}^T\Delta^T W\Delta\,\boldsymbol{\rho} \\ &\text{s.t.} && [\mathcal{D}\mathcal{Q}](\mathbf{r}-\boldsymbol{\rho})=0 \end{aligned}$$

Regularization

- Regularization subproblem: quadratic with linear constraints
- First order optimality conditions, **explicit solution** from

$$\begin{aligned} \Delta^T W \Delta\, \boldsymbol{\rho} + [\mathcal{D}\mathcal{Q}]^T \boldsymbol{\lambda} &= 0 \\ [\mathcal{D}\mathcal{Q}]\boldsymbol{\rho} &= [\mathcal{D}\mathcal{Q}]\mathbf{r} \end{aligned}$$

- May be ill-conditioned, SVD truncation, drop the smallest singular value
- Non-trivial weight ($W \neq I$) is needed for **piecewise constant** resistivities
- Use weight introduced in [Abubakar, Habashy, Druskin, Knizhnerman, Alumbaugh, 2008]

$$w_j = \left(([\Delta\, \mathbf{r}]_j)^2 + \phi(\mathbf{r})^2 \right)^{-1}, \quad j = 1, \ldots, N-1,$$

where

$$\phi(\mathbf{r}) = C_\phi \| \mathcal{Q}(d(\,\cdot\,)) - \mathcal{Q}(y(\,\cdot\,; \mathbf{r}) \|_2$$

- Sharp resolution of interfaces

Inversion algorithm

1. Solve the data fitting rational interpolation problem $(\kappa_j^\star, \widehat{\kappa}_j^\star)_{j=1}^m = \mathcal{Q}(d(\cdot))$
2. Work with logarithms $\mathbf{l}^\star = (\log \kappa_1^\star, \ldots, \log \kappa_m^\star, \log \widehat{\kappa}_1^\star, \ldots, \log \widehat{\kappa}_m^\star)^T$
3. Choose on initial guess $\mathbf{r}^{(1)} \in \mathcal{R}_+^N$
4. For $p = 1, \ldots, n_{GN}$ do
   1. Compute the non-linear preconditioner $(\kappa_j^{(p)}, \widehat{\kappa}_j^{(p)})_{j=1}^m = \mathcal{Q}(y(t; \mathbf{r}^{(p)}))$ and its Jacobian $\mathcal{DQ}^{(p)} = \mathcal{DQ}(y(t; \mathbf{r}^{(p)}))$
   2. Work with logs $\mathbf{l}^{(p)} = (\log \kappa_1^{(p)}, \ldots, \log \kappa_m^{(p)}, \log \widehat{\kappa}_1^{(p)}, \ldots, \log \widehat{\kappa}_m^{(p)})^T$
   3. Gauss-Newton step

$$\boldsymbol{\rho}^{(p)} = -\left(\mathcal{DQ}^{(p)}\right)^\dagger \operatorname{diag}\left(\kappa_1^{(p)}, \ldots, \kappa_m^{(p)}, \widehat{\kappa}_1^{(p)}, \ldots, \widehat{\kappa}_m^{(p)}\right) (\mathbf{l}^{(p)} - \mathbf{l}^\star)$$

   4. Gauss-Newton update $\mathbf{r}^{GN} = \mathbf{r}^{(p)} + \zeta^{(p)} \boldsymbol{\rho}^{(p)}$
   5. Compute the regularization weight for $\mathbf{r}^{GN}$
   6. Solve for next iterate $\mathbf{r}^{(p+1)}$ from

$$\begin{bmatrix} \Delta^T W \Delta & (\mathcal{DQ}^{(p)})^T \\ \mathcal{DQ}^{(p)} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{r}^{(p+1)} \\ \boldsymbol{\lambda}^{(p)} \end{bmatrix} = \begin{bmatrix} 0 \\ (\mathcal{DQ}^{(p)})\mathbf{r}^{GN} \end{bmatrix}$$

Numerical experiments: setup

- Time stepping over $[0, T_{\max}]$ with N_T steps to compute $\mathbf{y} \in \mathbb{R}^{N_T}$
- Systematic discretization errors even in the absence of noise

$$y_j = y(t_j; \mathbf{r}^{\text{true}}) + \xi^{(s)}(t_j), \quad j = 1, \ldots, N_T$$

- Generate the data by adding noise $\mathbf{d} = \mathbf{y} + \boldsymbol{\xi}^{(n)}$
- Noise model

$$\boldsymbol{\xi}^{(n)} = \epsilon \, \text{diag}(\chi_1, \ldots, \chi_{N_T})\mathbf{y},$$

with independent $\chi_k \in \mathcal{N}(0, 1)$

- Reduced model size m chosen based on noise level ϵ

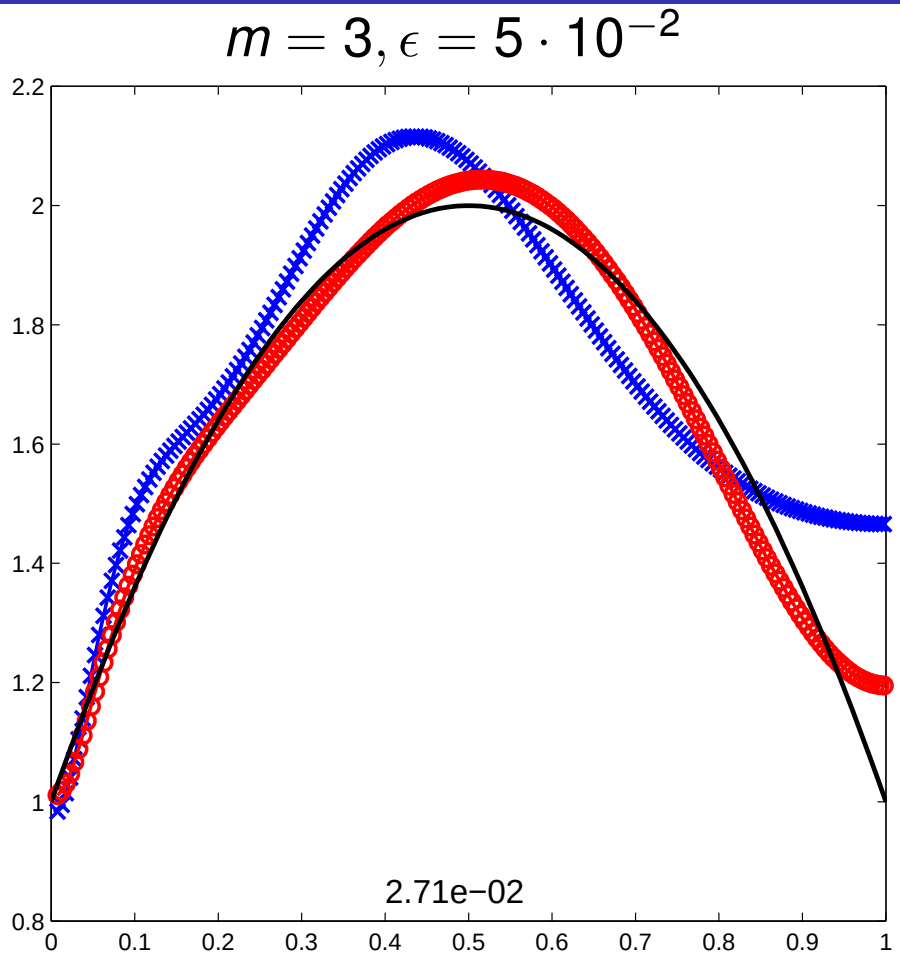
m	3	4	5	6
ϵ	$5 \cdot 10^{-2}$	$5 \cdot 10^{-3}$	10^{-4}	0 (noiseless)

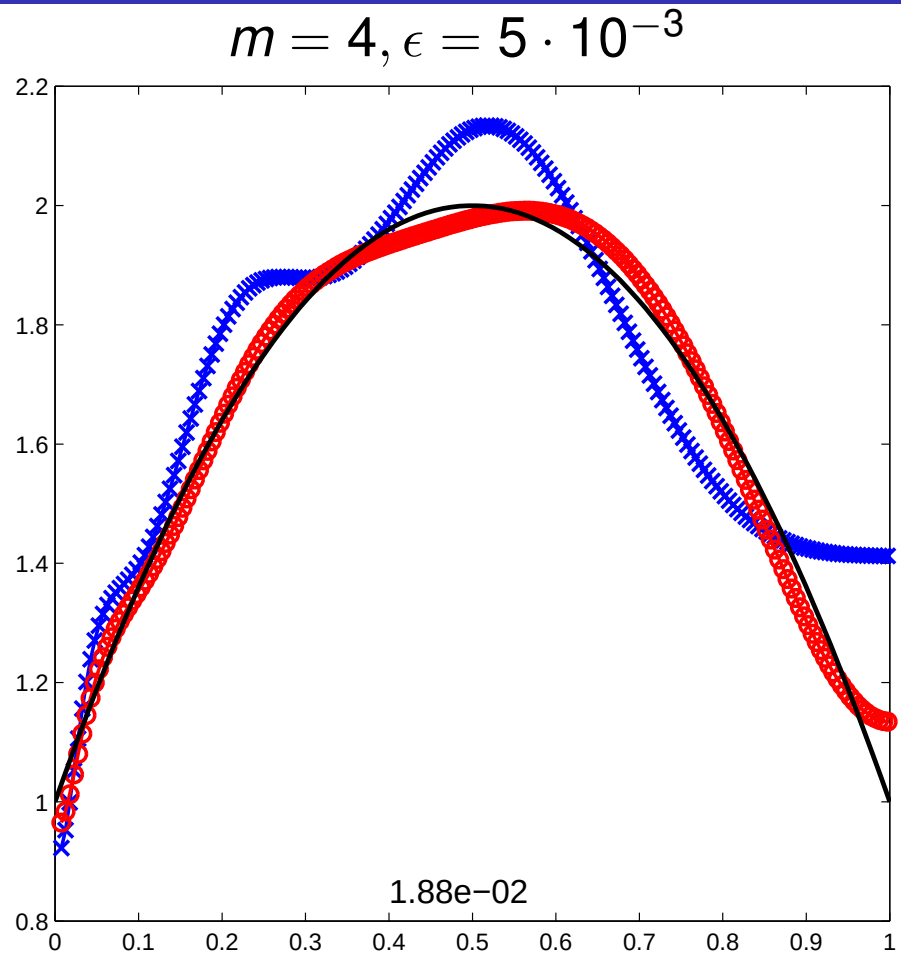
- Relative ℓ_2 error to measure the quality

$$\mathcal{E} = \frac{\|\mathbf{r}^\star - \mathbf{r}^{\text{true}}\|_2}{\|\mathbf{r}^{\text{true}}\|_2}$$

- Initial guess $\mathbf{r}^1 = \mathbf{1}$, five Gauss-Newton iterations $n_{GN} = 5$

Numerical results: smooth resistivity





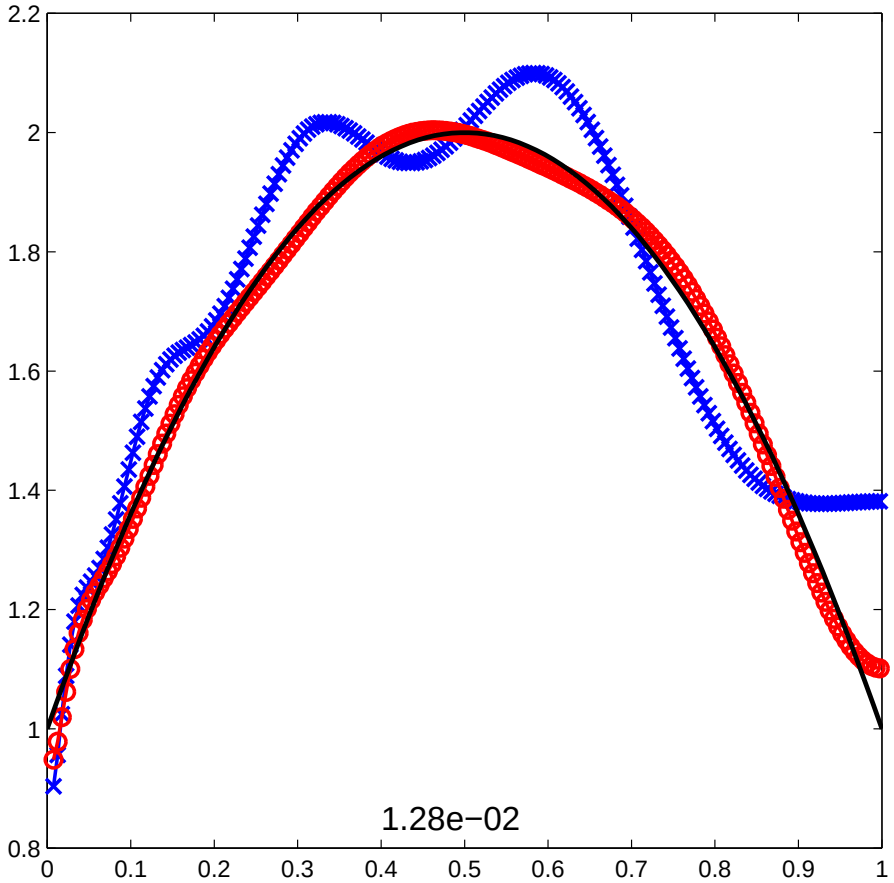
– True resistivity $\mathbf{r}^{\text{true}}$ (quadratic)

× Reconstruction $\mathbf{r}^{(2)}$ after one Gauss-Newton iteration

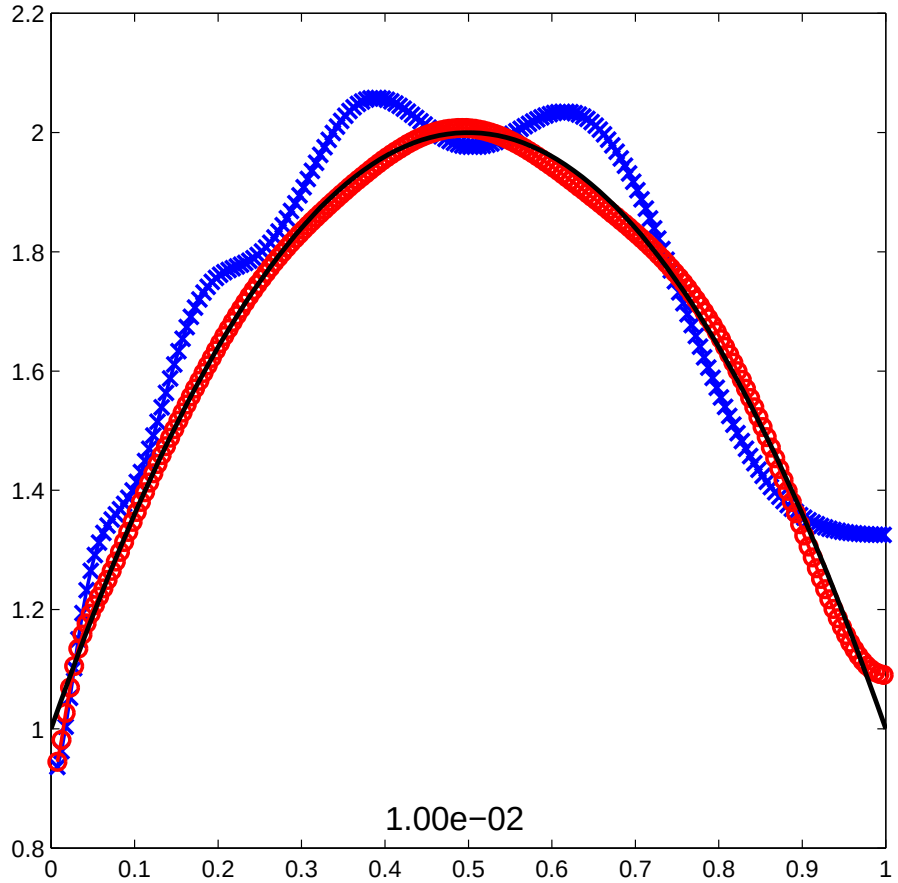
○ Reconstruction $\mathbf{r}^{(6)}$ after five Gauss-Newton iterations

Numerical results: smooth resistivity

$m = 5, \epsilon = 10^{-4}$



$m = 6, \epsilon = 0$ (noiseless)

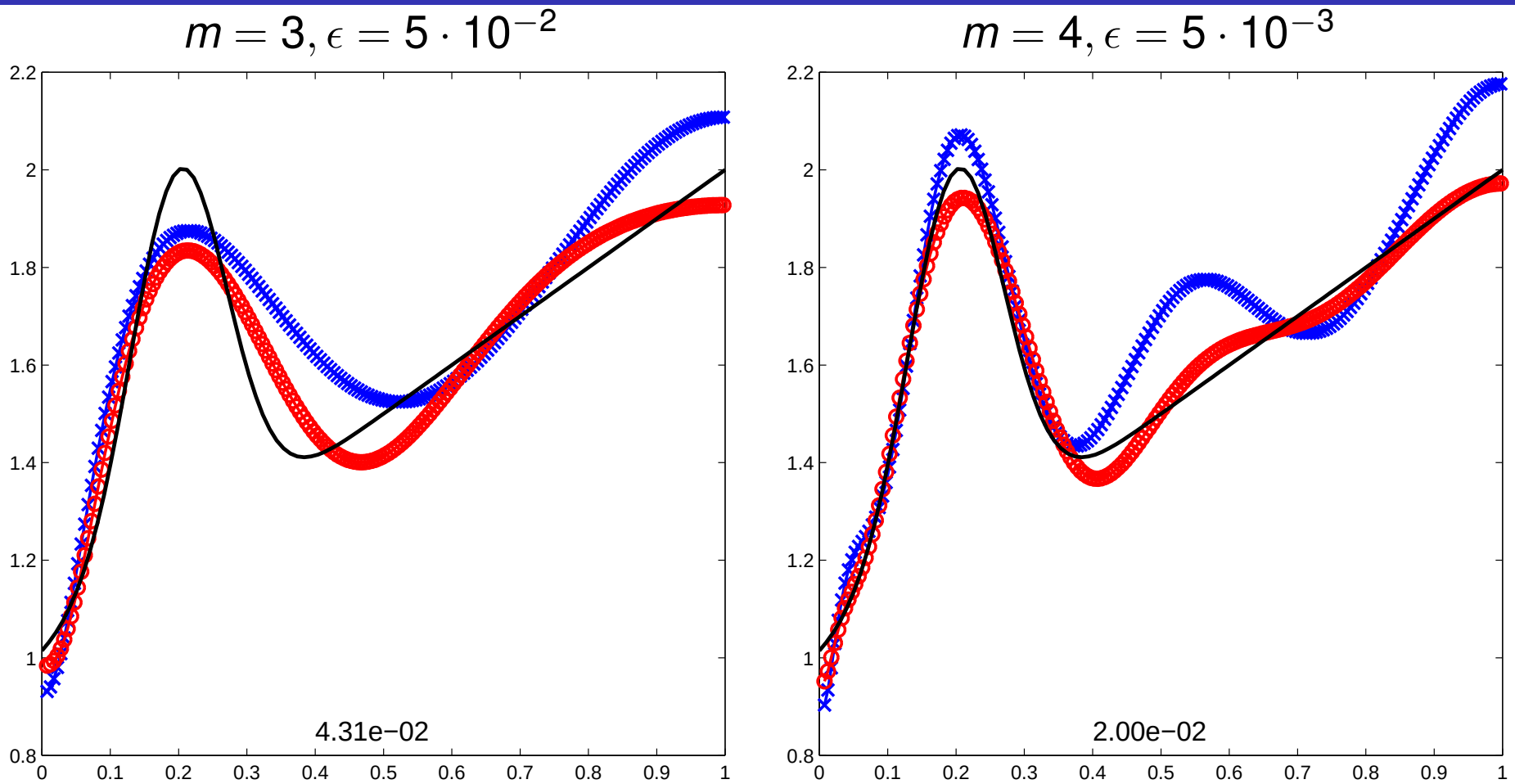


– True resistivity $\mathbf{r}^{\text{true}}$ (quadratic)

× Reconstruction $\mathbf{r}^{(2)}$ after one Gauss-Newton iteration

○ Reconstruction $\mathbf{r}^{(6)}$ after five Gauss-Newton iterations

Numerical results: smooth resistivity



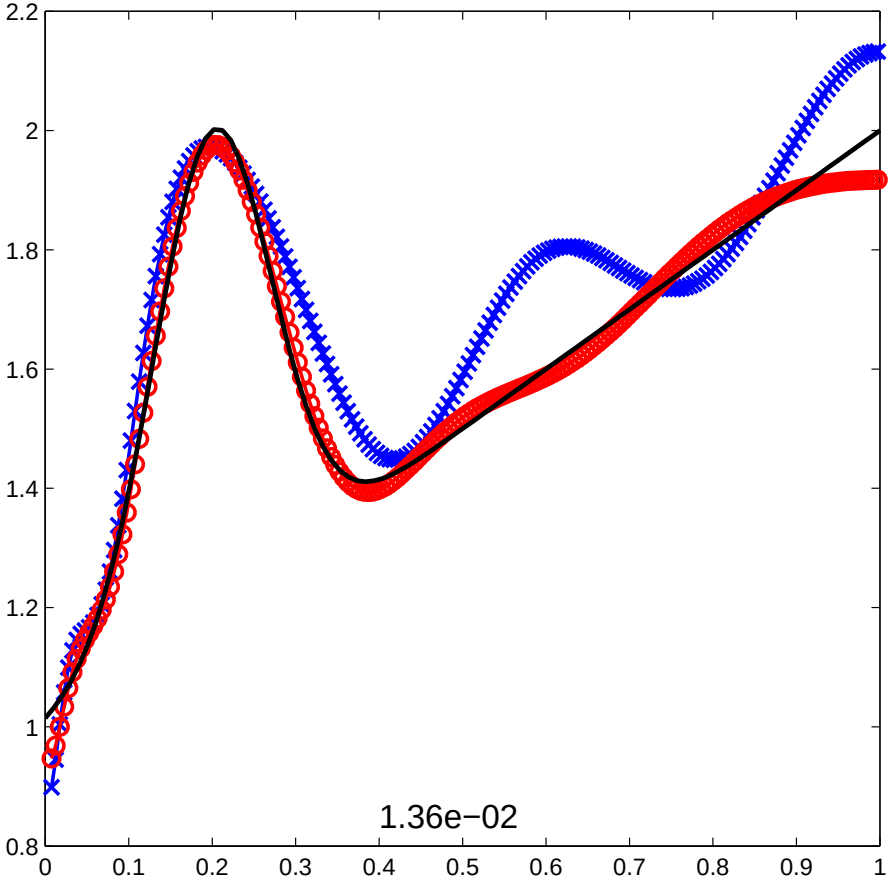
– True resistivity $\mathbf{r}^{\text{true}}$ (linear + Gaussian)

× Reconstruction $\mathbf{r}^{(2)}$ after one Gauss-Newton iteration

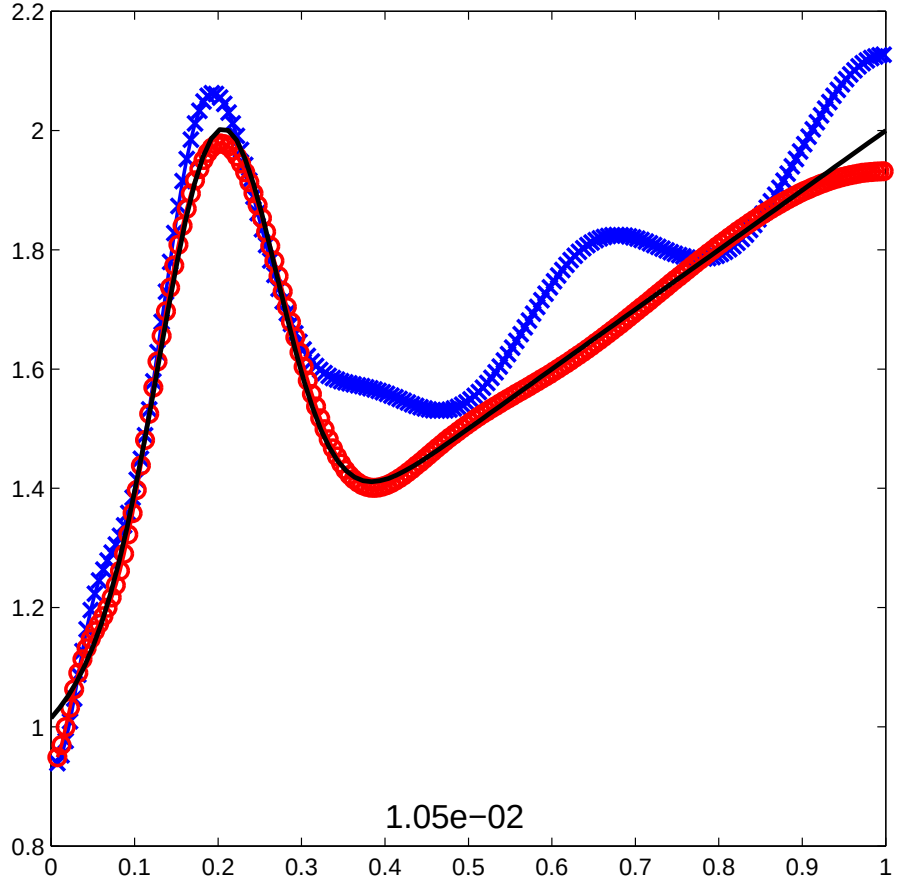
○ Reconstruction $\mathbf{r}^{(6)}$ after five Gauss-Newton iterations

Numerical results: smooth resistivity

$m = 5, \epsilon = 10^{-4}$



$m = 6, \epsilon = 0$ (noiseless)

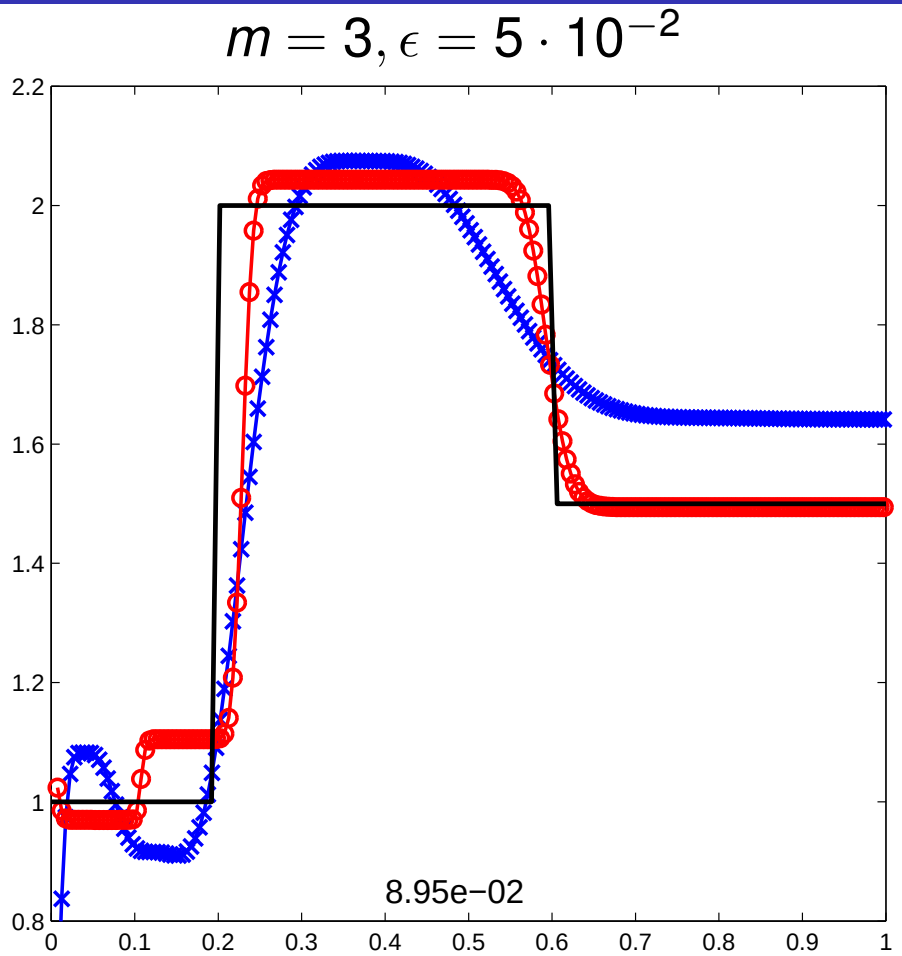


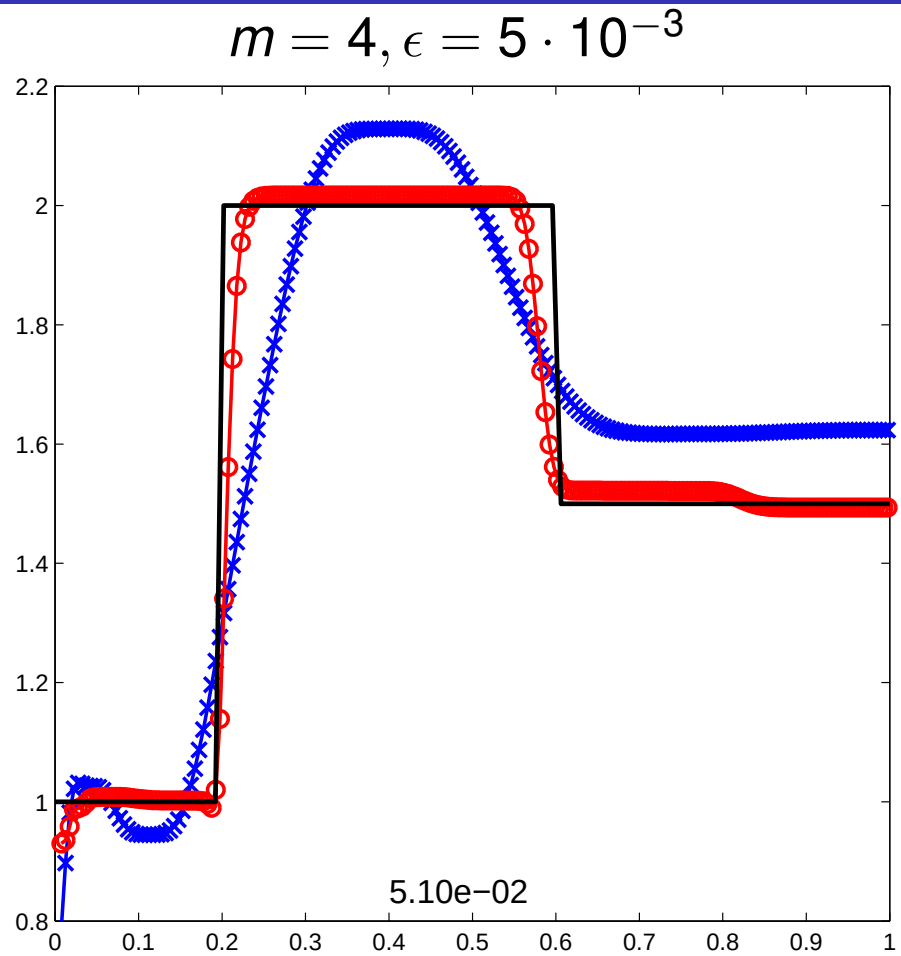
– True resistivity $\mathbf{r}^{\text{true}}$ (linear + Gaussian)

× Reconstruction $\mathbf{r}^{(2)}$ after one Gauss-Newton iteration

○ Reconstruction $\mathbf{r}^{(6)}$ after five Gauss-Newton iterations

Numerical results: piecewise constant resistivity





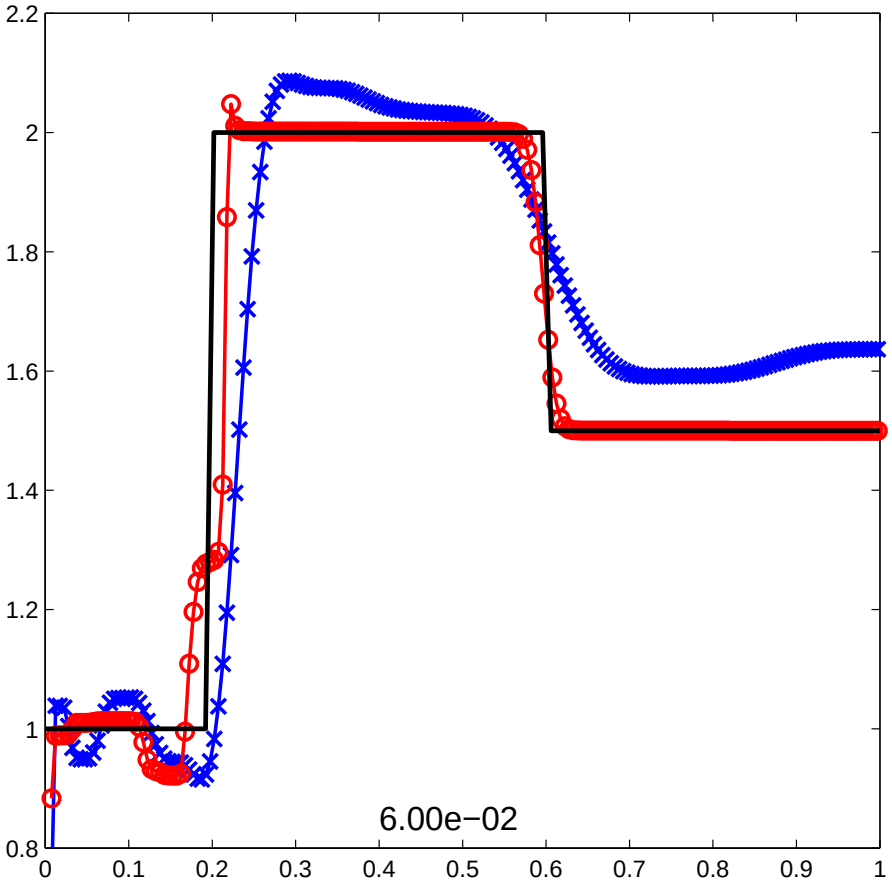
– True resistivity $\mathbf{r}^{\text{true}}$ (jump of contrast 2)

× Reconstruction $\mathbf{r}^{(2)}$ after one Gauss-Newton iteration

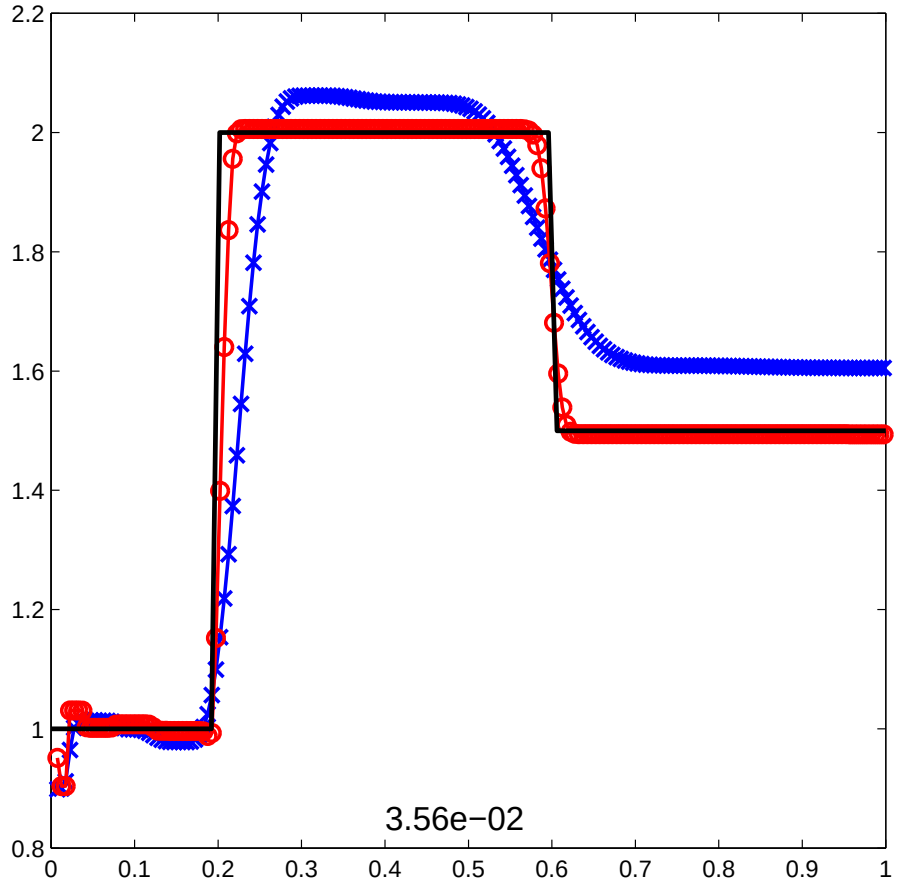
○ Reconstruction $\mathbf{r}^{(6)}$ after five Gauss-Newton iterations

Numerical results: piecewise constant resistivity

$m = 5, \epsilon = 10^{-4}$



$m = 6, \epsilon = 0$ (noiseless)



– True resistivity $\mathbf{r}^{\text{true}}$ (jump of contrast 2)

× Reconstruction $\mathbf{r}^{(2)}$ after one Gauss-Newton iteration

○ Reconstruction $\mathbf{r}^{(6)}$ after five Gauss-Newton iterations

Approaches to higher dimensions

Higher dimensions:

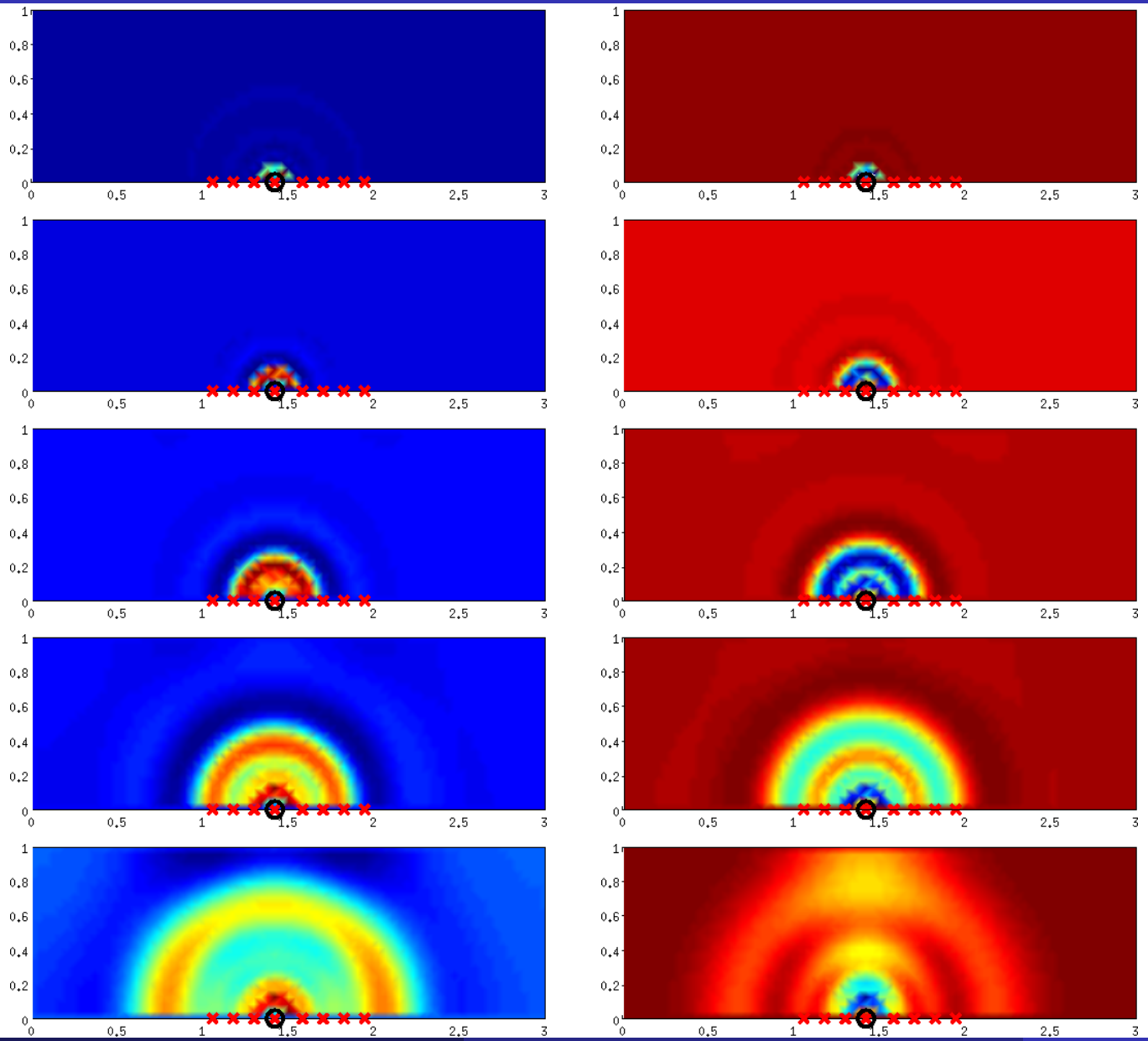
- One-dimensional problem is formally determined: 1D unknown $r(x)$ and 1D data $y(t)$
- In dimension two the problem is already **overdetermined**: 2D unknown $r(x, y)$, but 3D data $y_{i,j}(t)$, where (i, j) are source-detector pairs
- Straightforward generalization: block Lanczos, block Krylov subspaces, matrix-valued continued fractions
- Block tridiagonal matrix with **dense** blocks, consequence of an overdetermined problem
- Dense blocks do not correspond directly to a finite-difference scheme
- Work with a subset of the data, make the problem formally determined

Coinciding sources and receivers (transducers)

Using one scalar continued fraction per transducer:

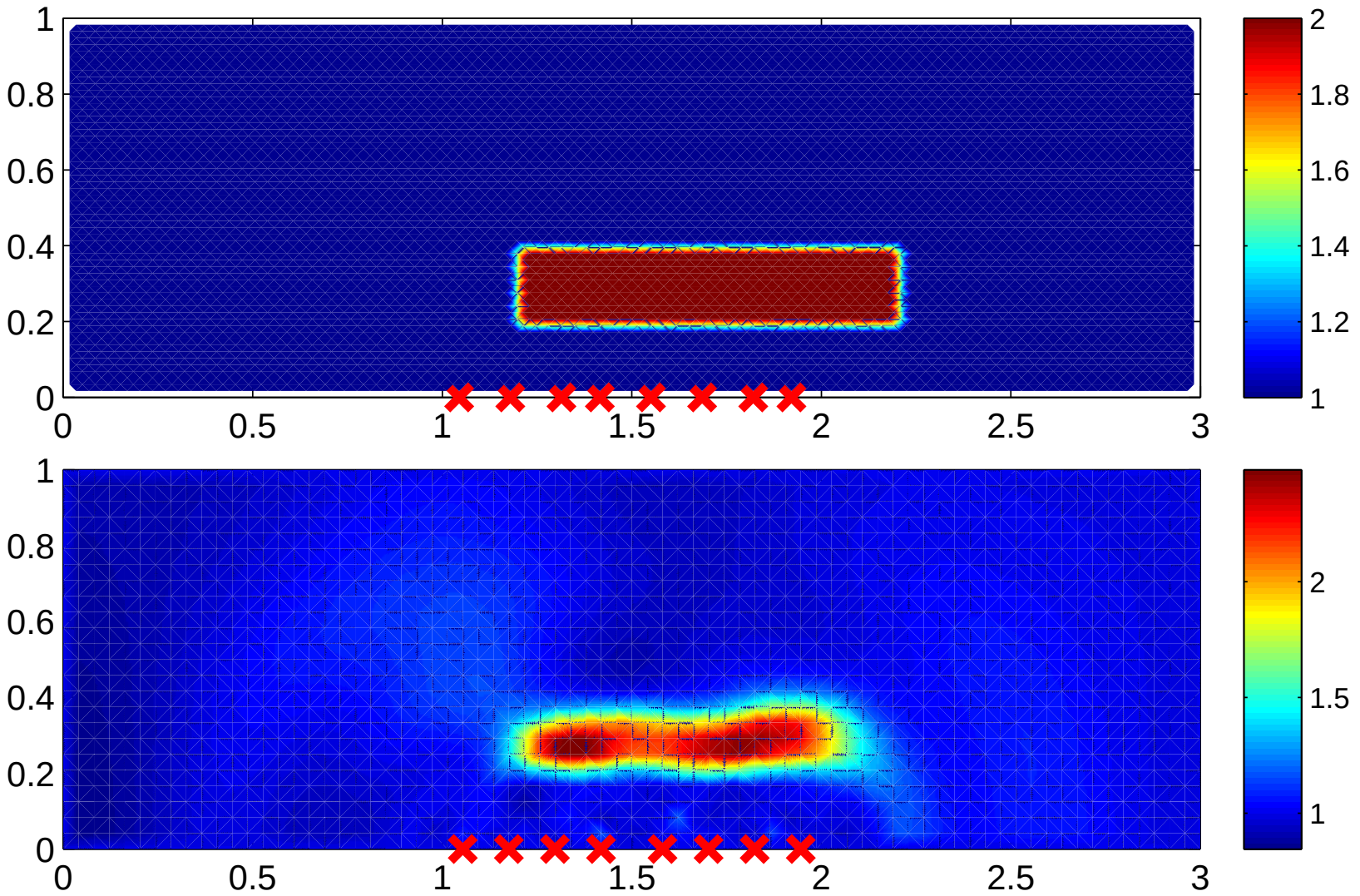
- Simplest reduction of the data $y_{i,j}(t)$: take the diagonal $i = j$
- Sources and receivers coincide
- Excite at a point on the boundary at $t = 0$, measure $y_j(t)$ at the same location for $t > 0$ for each transducer $j = 1, \ldots, n$
- Not the best setting in practice, measurements at source locations may be noisy
- Easier to work with theoretically, coinciding sources/receivers preserve the symmetry
- Construct separate scalar continued fractions interpolating each $y_j(t)$, $j = 1, \ldots, n$
- Continued fractions are again Stieltjes due to the symmetry

Sensitivities



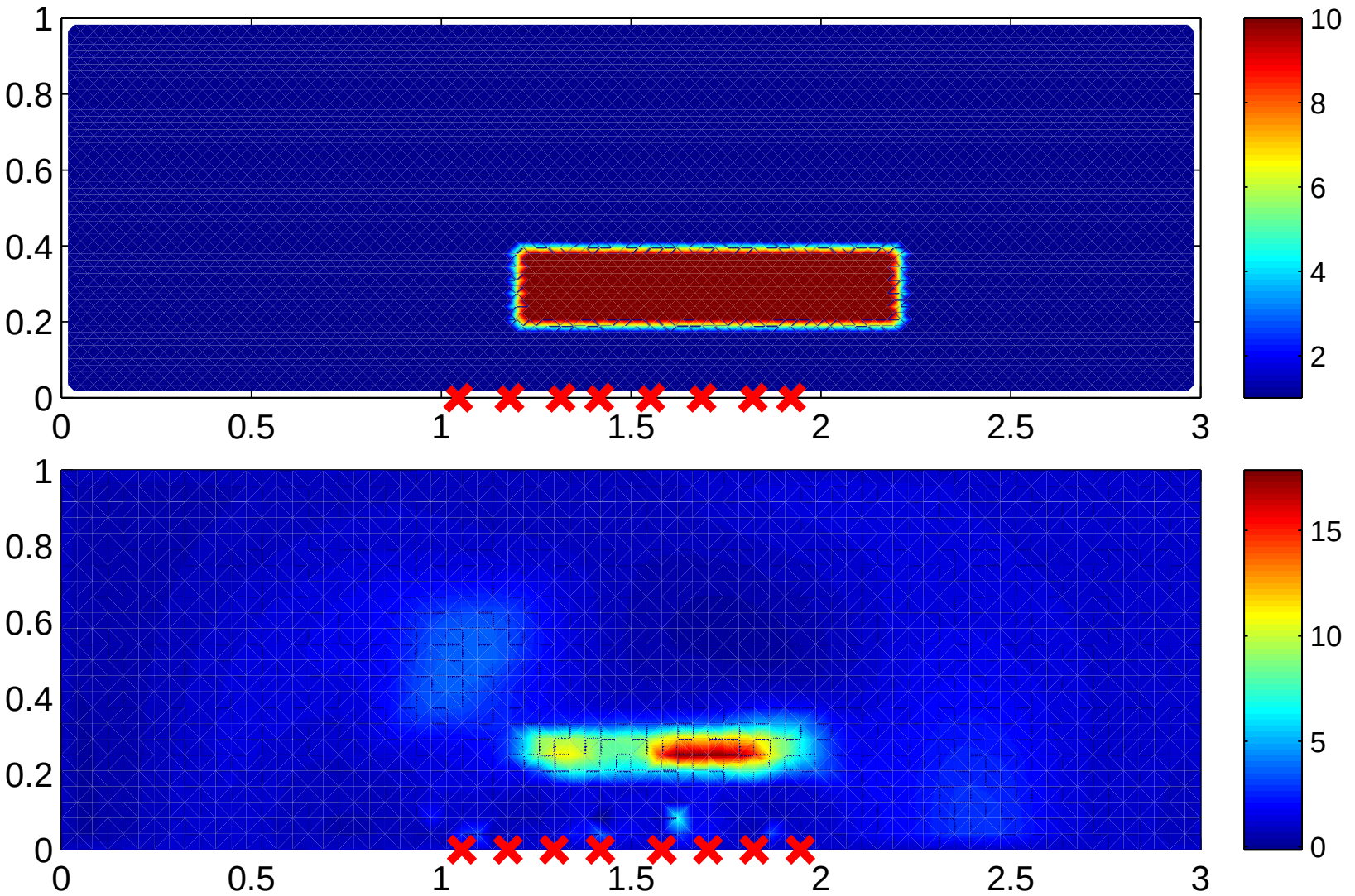
Sensitivity functions of $\widehat{\kappa}_j$ (left) and κ_j (right) for $j = 1, \ldots, m$ (top to bottom), $m = 5$ for a single transducer (black $\circ$) out of $n = 8$ (red $\times$).
Simple Pade approximant at $\sigma = 60$. Sensitivities resemble propagating spherical waves. Higher σ means lower speed of propagation. Should avoid reflections from boundaries.

Reconstructions: single low contrast inclusion



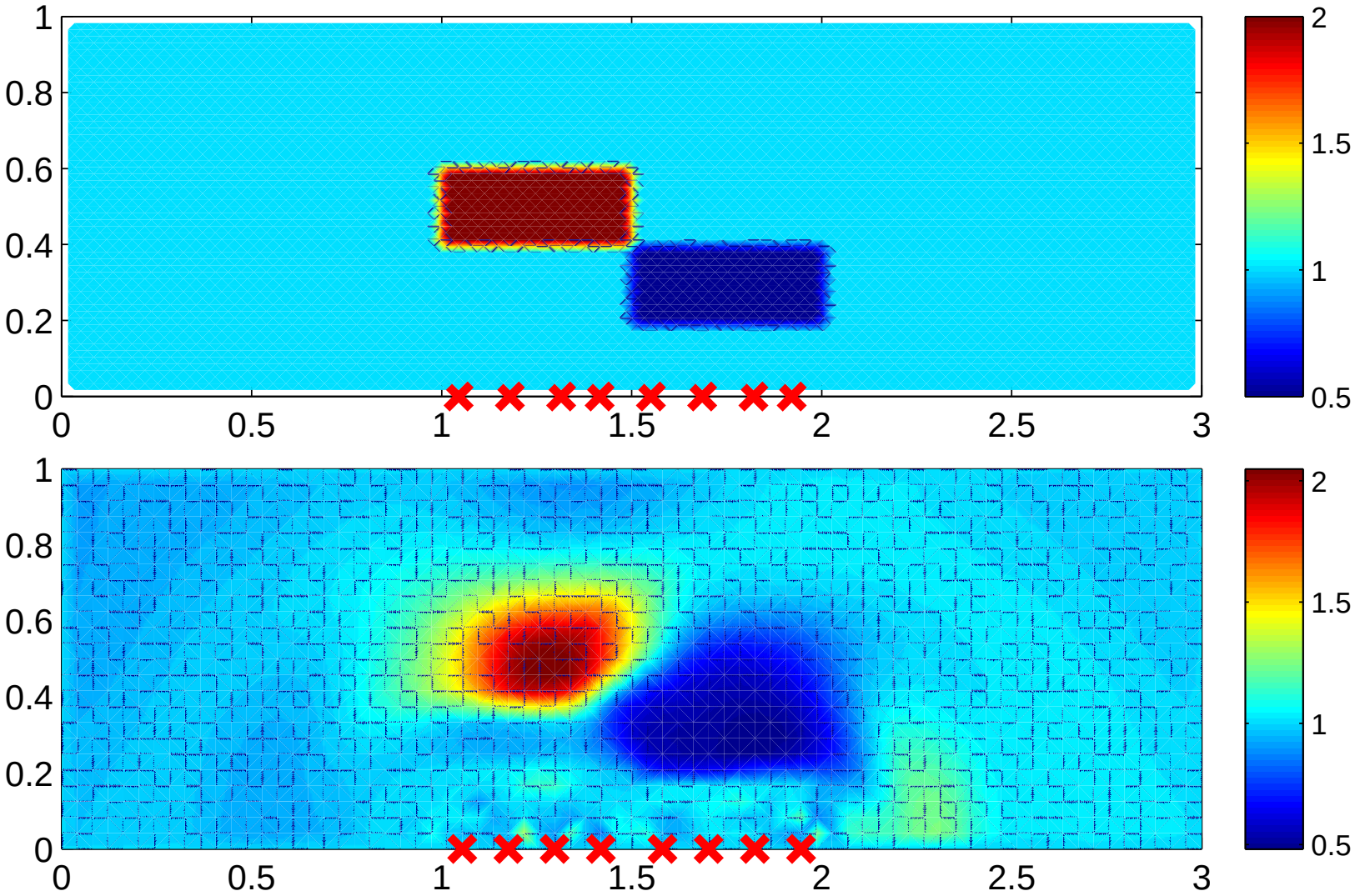
Top: true $r(x, y)$. Bottom: reconstruction after a single Gauss-Newton iteration. Constant initial guess $r_0(x, y) \equiv 1$. Transducers: red $\times$.

Reconstructions: single high contrast inclusion



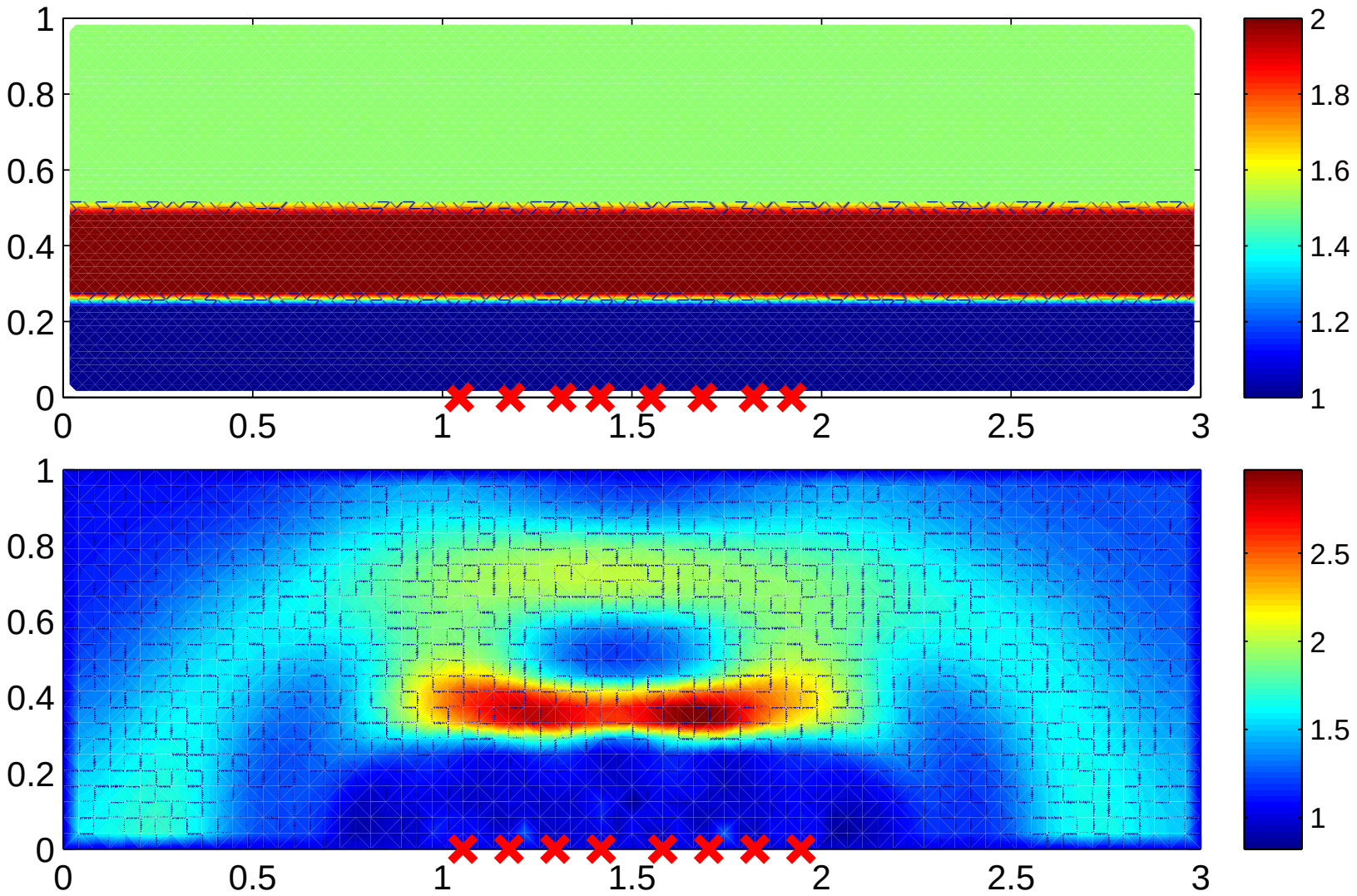
Top: true $r(x, y)$. Bottom: reconstruction after a single Gauss-Newton iteration. Constant initial guess $r_0(x, y) \equiv 1$. Transducers: red $\times$.

Reconstructions: two adjacent inclusions



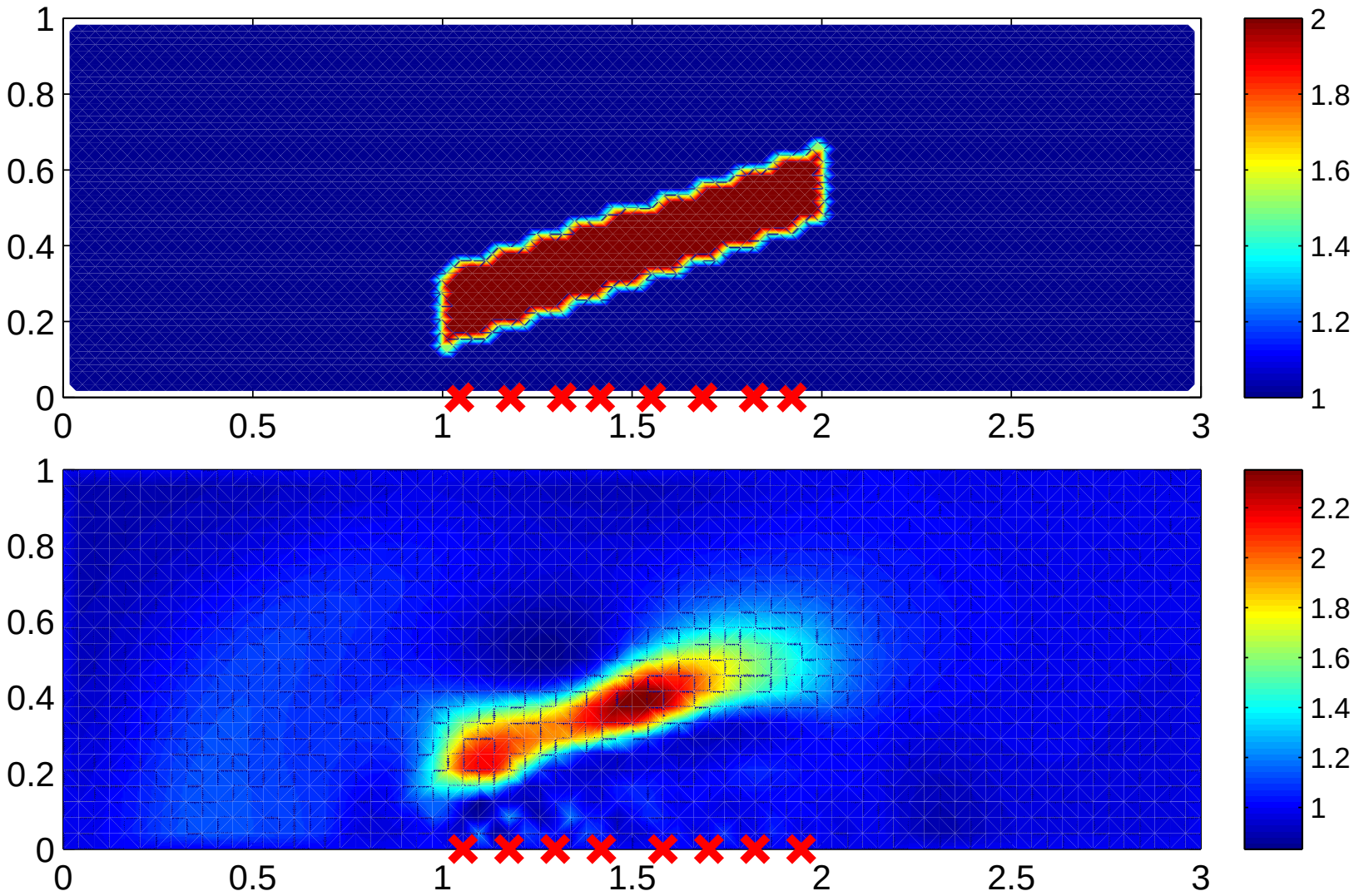
Top: true $r(x, y)$. Bottom: reconstruction after a single Gauss-Newton iteration. Constant initial guess $r_0(x, y) \equiv 1$. Transducers: red $\times$.

Reconstructions: layered medium



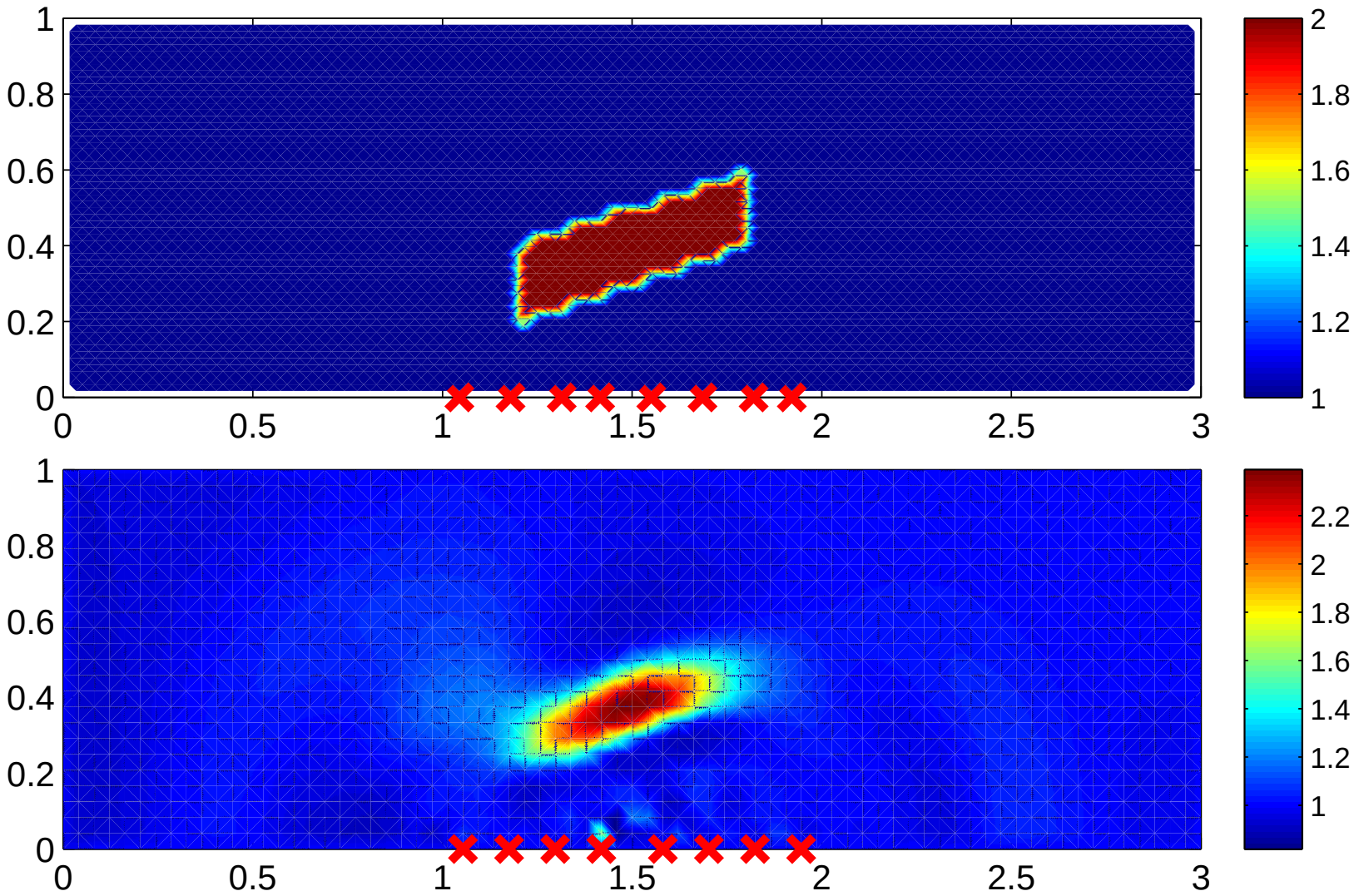
Top: true $r(x, y)$. Bottom: reconstruction after a single Gauss-Newton iteration. Constant initial guess $r_0(x, y) \equiv 1$. Transducers: red $\times$.

Reconstructions: skewed inclusion, low aperture



Top: true $r(x, y)$. Bottom: reconstruction after a single Gauss-Newton iteration. Constant initial guess $r_0(x, y) \equiv 1$. Transducers: red $\times$.

Reconstructions: skewed inclusion, high aperture



Top: true $r(x, y)$. Bottom: reconstruction after a single Gauss-Newton iteration. Constant initial guess $r_0(x, y) \equiv 1$. Transducers: red $\times$.

Conclusions and future work

Conclusions:

- Non-linear preconditioning based on model reduction
- Data fitting: rational interpolation (unavoidably ill-conditioned)
- Reconstruction: well-conditioned
- Fast convergence, inexpensive
- Possible to extend to higher dimensions

Future work:

- More work on 2D and 3D
- Non-coinciding source-receiver pairs
- Deal with the loss of symmetry

Preprint: *A model reduction approach to numerical inversion for a parabolic partial differential equation*. L. Borcea, V. Druskin, A.V. Mamonov and M. Zaslavsky, 2012, arXiv:1210.1257 [math.NA]