Using Neumann Series to Solve Inverse Problems in Imaging

Christopher Kumar Anand

Advanced Optimization Laboratory

McMaster University
Inspiring Innovation and Discovery

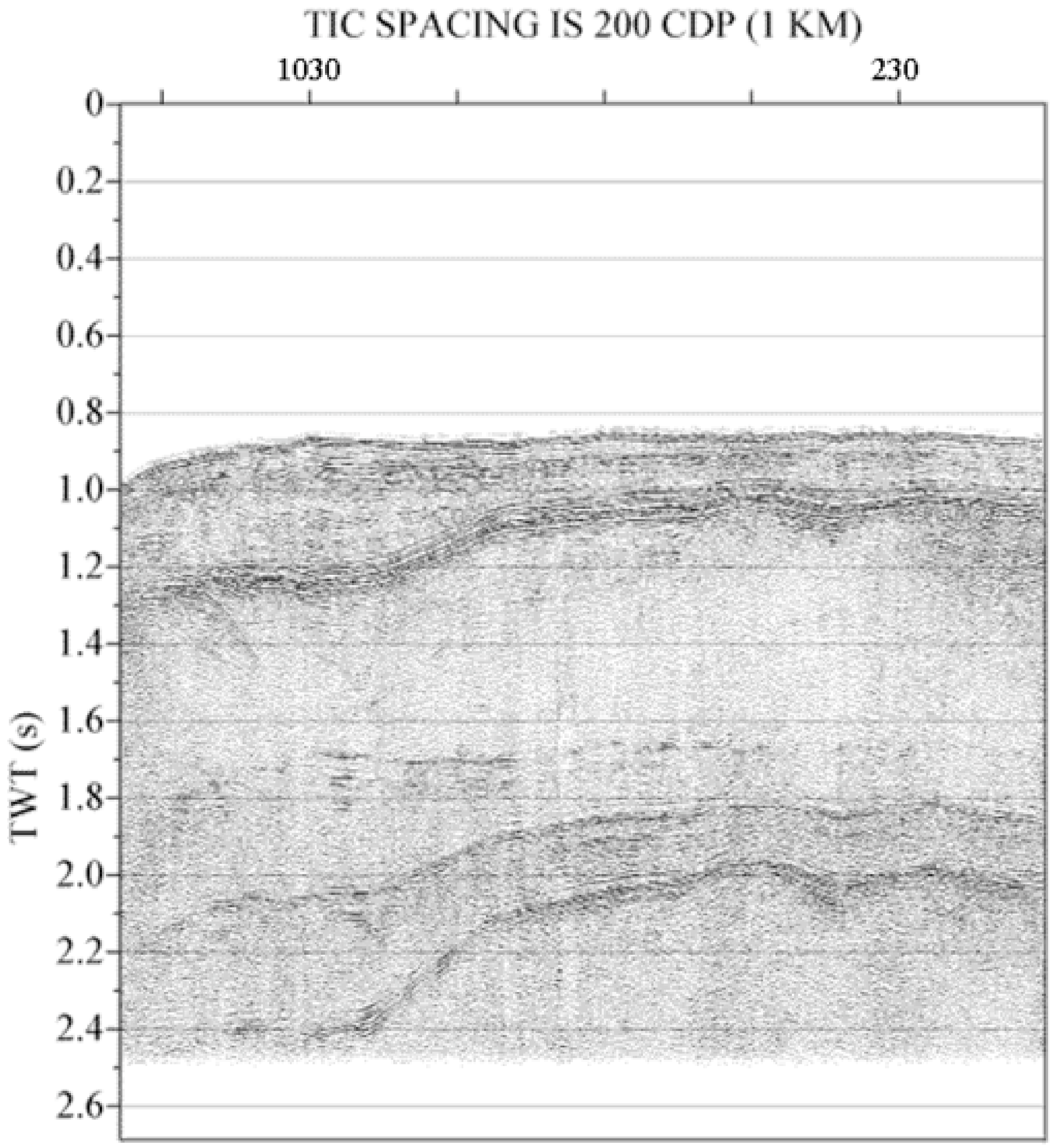
Inverse Problem

$$Tf = m$$

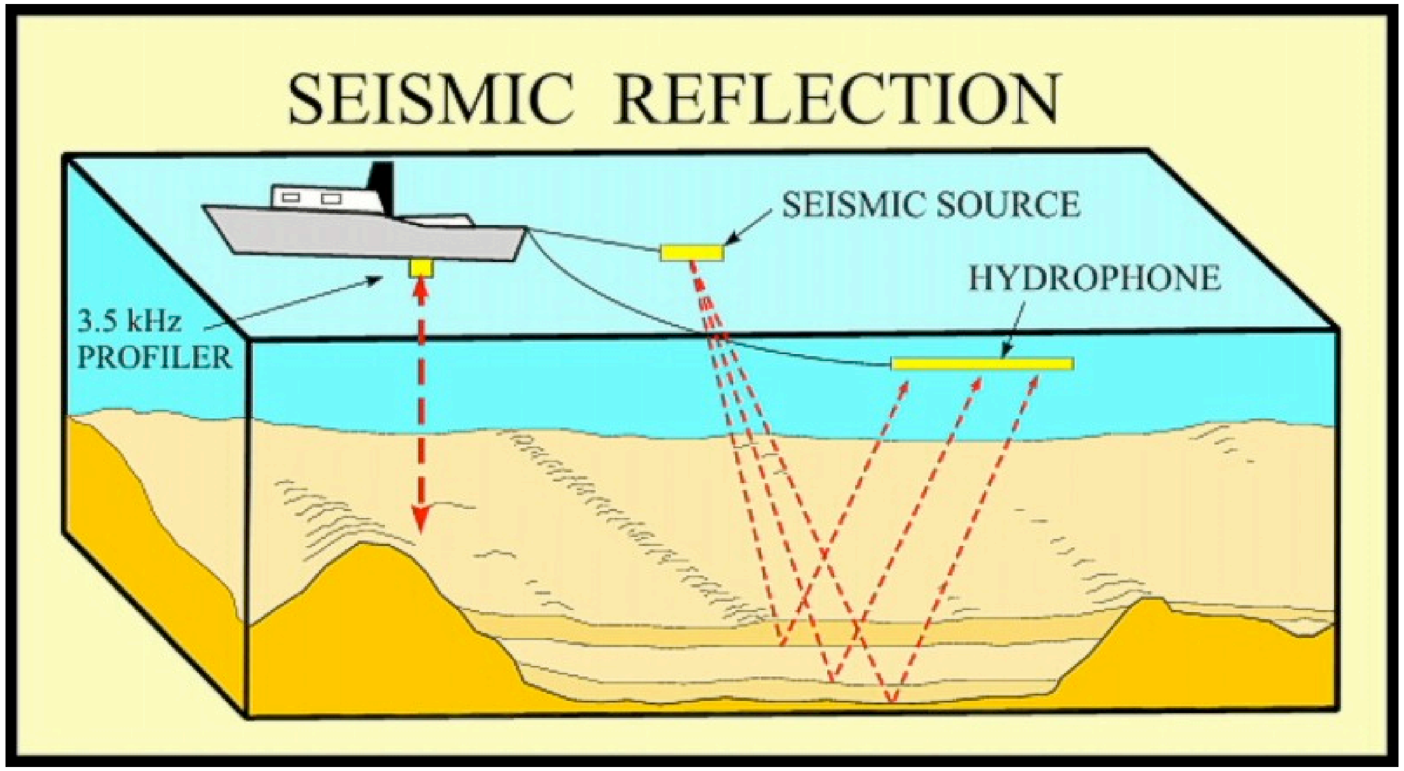
Solve for f.

- given
  - measurements (m)
  - model (T)

Seismic Imaging



1999 Gulf of Mexico Line 7

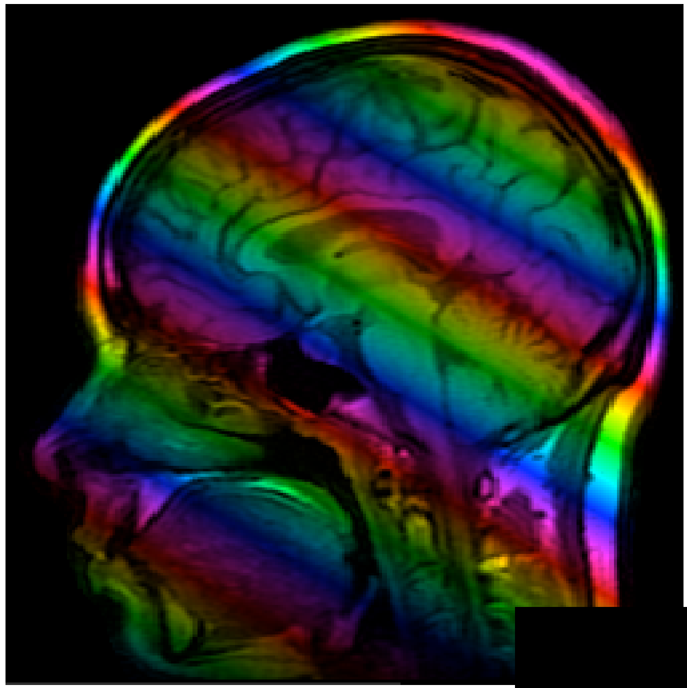


1. Bang
2. Listen
3. Solve Acoustic Equations

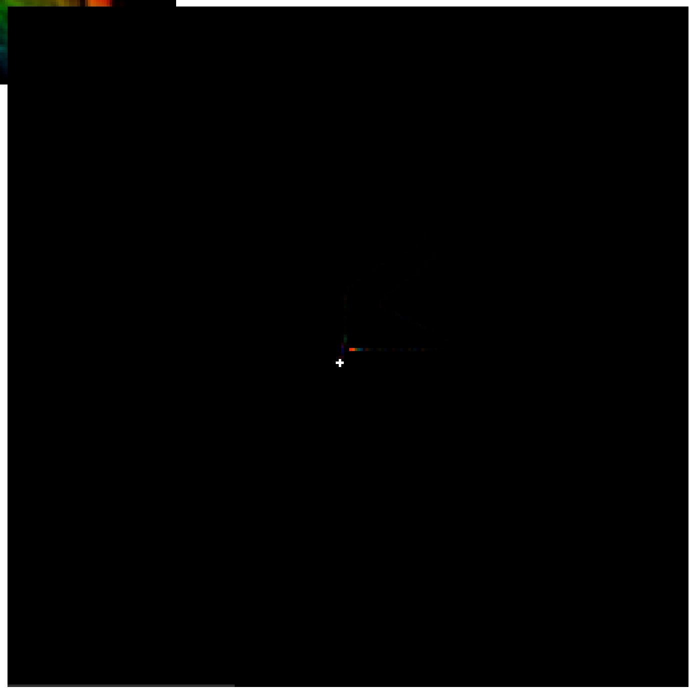
http://en.wikipedia.org/wiki/Seismology

Magnetic Resonance Imaging

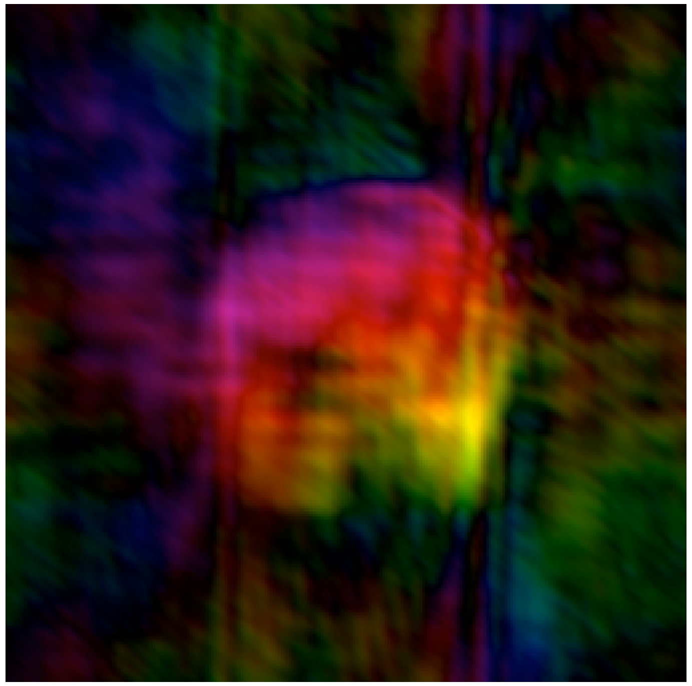
0. Tissue Density



1. Phase Modulation



2. Sample Fourier Transform



3. Invert Linear System

Challenging when...

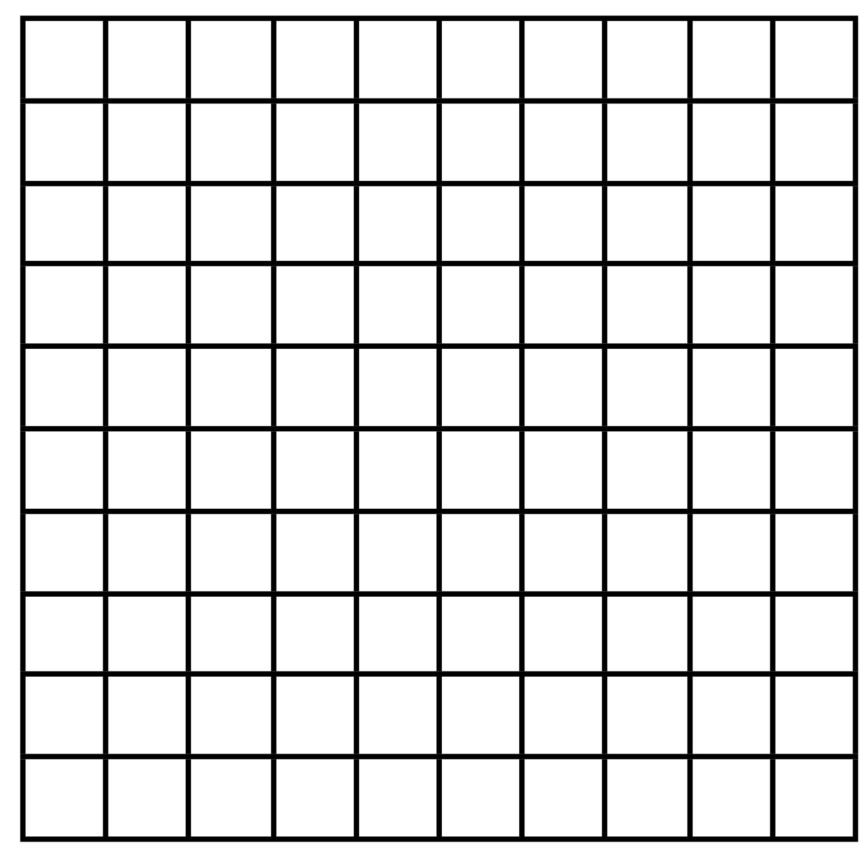
- model is big
  - 1 000 000 000 variables
- model is nonlinear
- data is inexact

  (usually know error probabilistically)

Imaging

- discretize continuous model
  - regular volume/area elements
    - sparse structure

$\rho_{i,j}$



Solutions: Noise

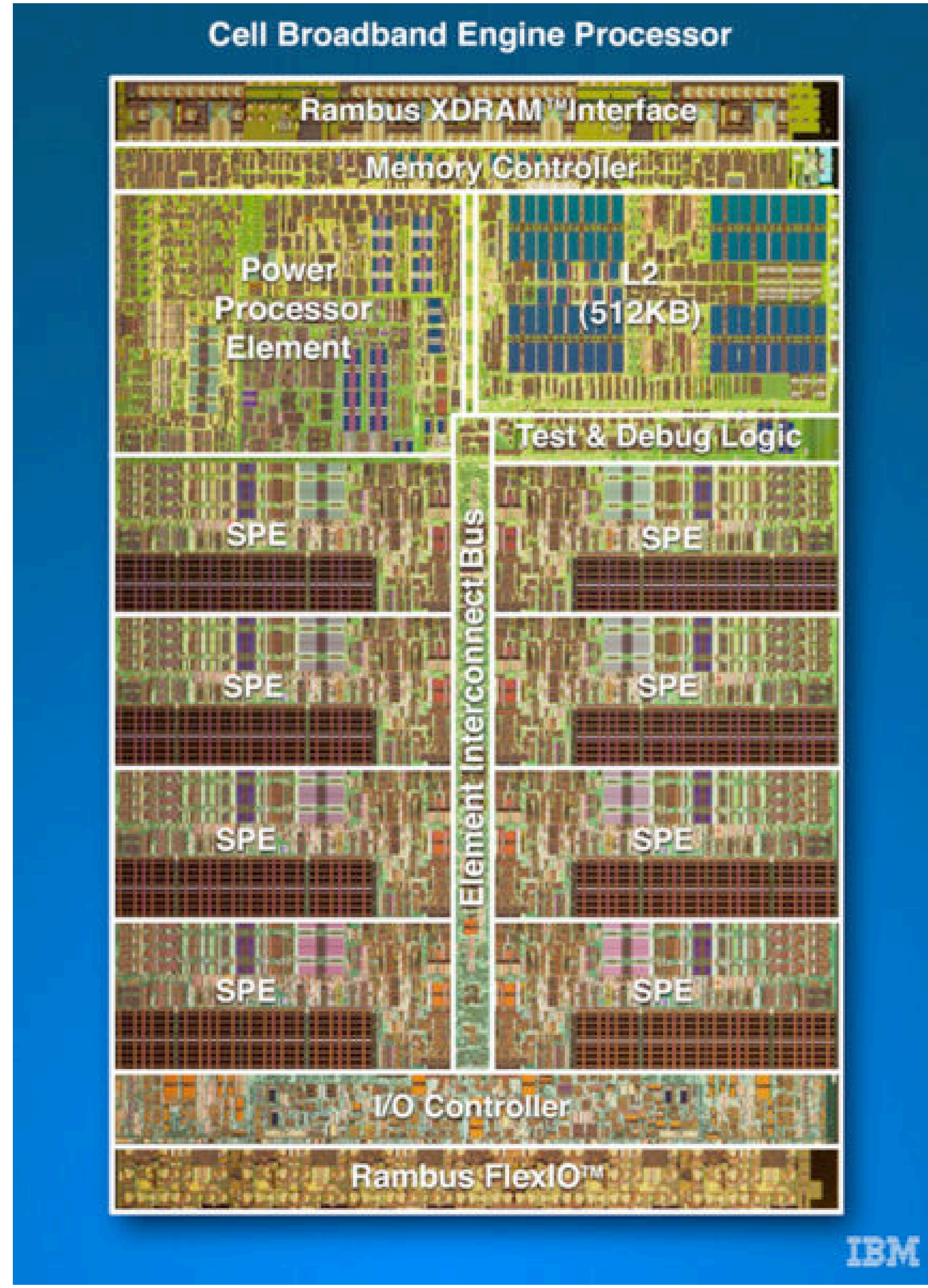
- filter noisy solution
  1. convolution filter
  2. bilateral filter
  3. Anisotropic Diffusion (uses pde)
- regularize via penalty
  1. energy
  2. Total Variation
  3. something new

Solutions: Problem Size

I. use a fast method (i.e. based on FFT)
II. use (parallelizable) iterative method
   a. Conjugate Gradient
   b. Neumann series
III. use sparsity
   a. choose penalties with sparse Hessians
IV. use fast hardware
   a. 1000-way parallelizable
   b. single precision

Cell BE

- 25 GFlops DP
- 200 GFlops SP
- need 384-way ||ism
  - 4-way SIMD
  - 8-way cores
  - 6-times unrolling
  - double buffering



Solutions: Nonlinearity

use iterative method

A. sequential projection onto convex sets

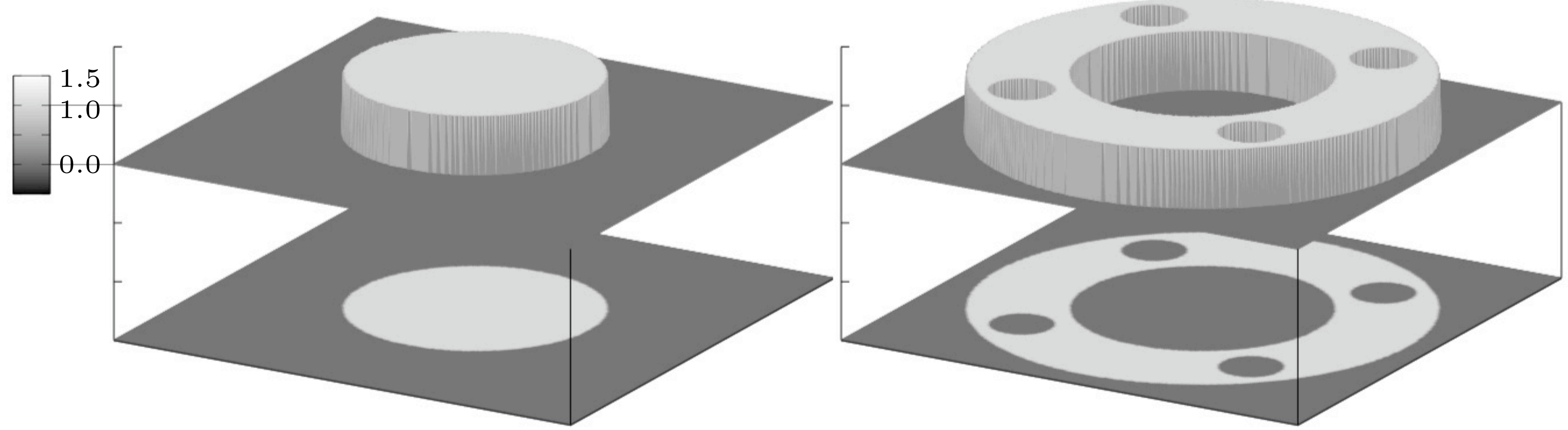
B. trust region

C. sequential quadratic approximations

Plan of Talk

A. example/benchmark
B. optimization
   1. fit to data
   2. regularization
      i. new penalty (with optimized gradient)
      ii. nonlinear penalties
C. solution
   1. operator decomposition
   2. Neumann series
D. proof of convergence
E. numerical example
   1. noise reduction
   2. linear convergence

Example/Benchmark



$\rho : \Omega \to \mathbb{C}$ complex image

$\mu_1, \mu_2, \mu_3, \mu_4 : \Omega/2 \to \mathbb{C}$ complex data

$$\mu_{m;i,j} = S_{m;i,j}\rho_{i,j} + S_{m;i+128,j}\rho_{i,j} + \epsilon_{m;i,j}$$

model

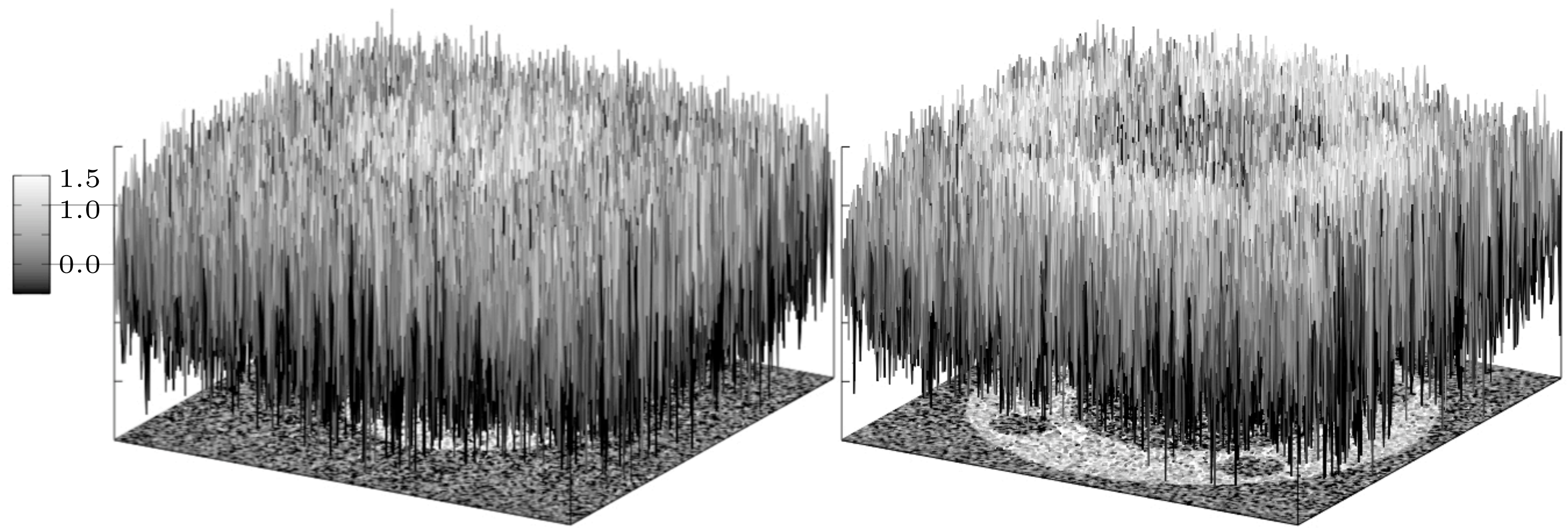
Example/Benchmark

$$\mu_{m;i,j} = S_{m;i,j}\rho_{i,j} + S_{m;i+128,j}\rho_{i,j} + \epsilon_{m;i,j}$$

$$T = \begin{pmatrix} S_{1;\mathrm{top}} & S_{1;\mathrm{bottom}} \\ S_{2;\mathrm{top}} & S_{2;\mathrm{bottom}} \\ S_{3;\mathrm{top}} & S_{3;\mathrm{bottom}} \\ S_{4;\mathrm{top}} & S_{4;\mathrm{bottom}} \end{pmatrix}$$ diagonal blocks

$$T^T T$$ easily invertible

Example/Benchmark



direct inverse

$$(T^T T)^{-1} T^T \mu$$

Optimization

fit-to-data

$$\min_{\rho:\Omega\to\mathbb{C}} \lambda\phi_{\mathrm{data}} + \lambda_{\mathrm{bi2}}\phi_{\mathrm{bi2}} + \lambda_{\mathrm{biTv}}\phi_{\mathrm{biTv}} + \lambda_{\mathrm{mask}}\phi_{\mathrm{mask}} + \lambda_{\mathrm{magnet}}\phi_{\mathrm{magnet}} + \lambda_{\mathrm{seg}}\phi_{\mathrm{seg}}$$

quadratic penalties

nonlinear penalties

Fit to Data

$$\phi_{\mathrm{data}}(f) = \|Tf - m\|^2$$

$$\nabla\phi_{\mathrm{data}} = 2T^T Tf - 2T^T m$$

$$\mathcal{H}\phi_{\mathrm{data}} = 2T^T T$$

- typically dense transformation
- use fast matrix-vector products
  - e.g. FFT-based forward/adjoint problem
- linear forward problem gives quadratic objective

Bilateral Filter

$$\hat{f}(x) = \sum_{y \in \mathbb{R} \setminus \{x\}} \underbrace{c(y-x)}_{\text{spatial kernel}} \underbrace{s(f(y) - f(x))}_{\text{range kernel}} f(y)$$

Bilateral Regularizer

$$\phi_{\mathrm{bi}}(f) = \sum_{y \neq x} c(y-x) s(f(y) - f(x))$$

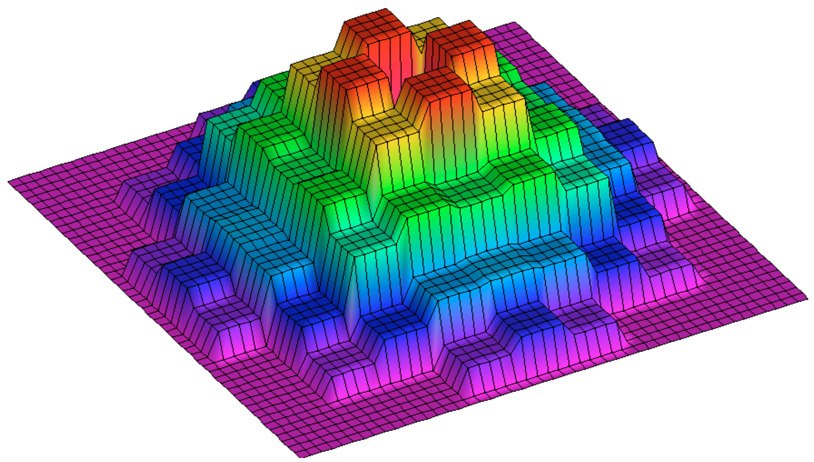
Quadtratic Case

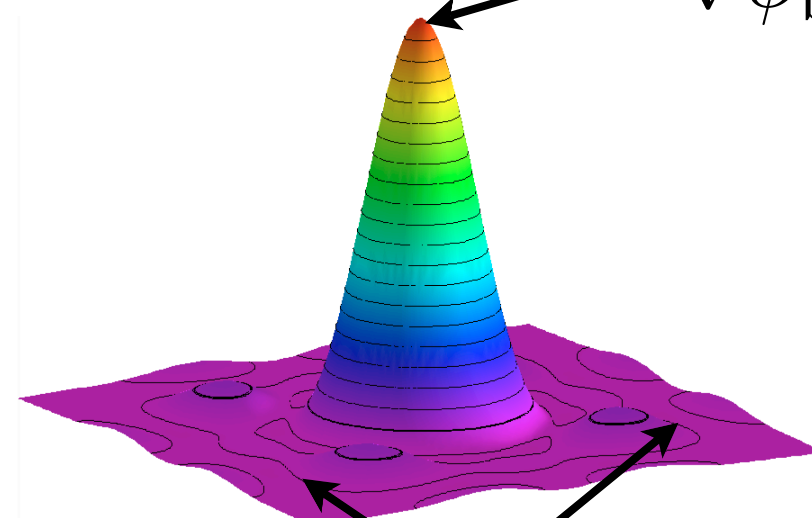
$$\phi_{\mathrm{bi2}} = \sum_{y \neq x} c(y-x) \left\| f(y) - f(x) \right\|^2$$

- optimize direction of gradient
- LP problem like FIR filter design
  - heuristic choice of “stop band”

$$\frac{\partial}{\partial f_i(x)} \phi_{\mathrm{bi2}} = 2 \sum_{y \neq x} (f_i(x) - f_i(y)) c(x-y) \quad = 2 \left(f_i(x) - \sum_{y \neq x} f_i(y) c(y-x) \right)$$

Optimal Spatial Kernel





$$-\nabla\phi_{\mathrm{bi}}(f_{\mathrm{zero}}) = 0$$

$$-\nabla\phi_{\mathrm{bi}}(f_{\mathrm{high}}) = -f_{\mathrm{high}}$$

discrete kernel

not rotational symmetric

exact in
quadratic case

use in
all cases

TV-like

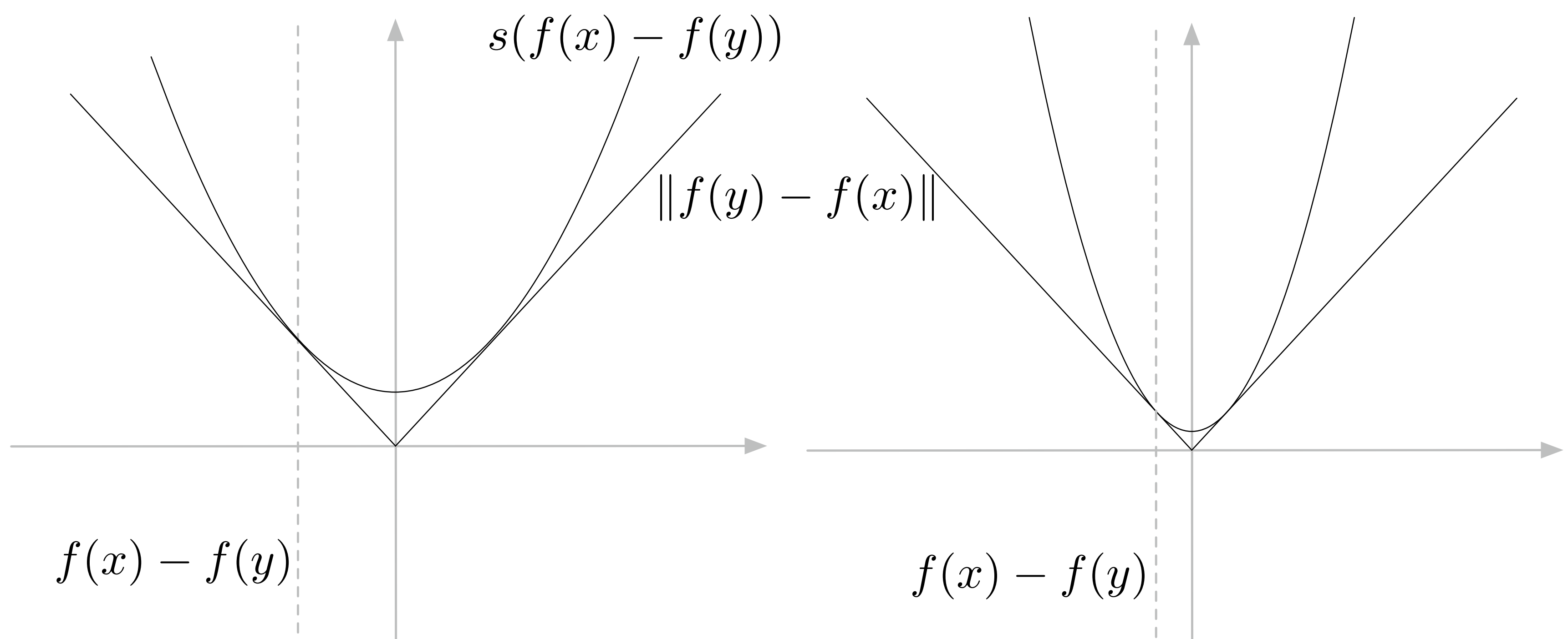
$$\phi_{\mathrm{biTV}} = \sum_{y \neq x} c(y-x)\, \|f(y) - f(x)\|$$

- similar properties to Total Variation
  - won't smooth edges
  - not differentiable

Sequential Quadratic TV-like

$$\phi_{\mathrm{biTv}} = \sum_{y \neq x} \frac{\epsilon c(y-x)}{\sqrt{\left\|\tilde{f}(y) - \tilde{f}(x)\right\|^2 + \epsilon}} \|f(y) - f(x)\|^2$$

- sequential quadratic approximation
- tangent to TV-like



Mask

$$\phi_{\mathrm{mask}} = \sum_{\{x \mid x \text{ is air}\}} \|f(x)\|^2$$

- penalize pixel values outside object

Segmentation

$$\phi_{\text{seg}}(f) = \sum_{\{g \,|\, \text{mean values of components}\}} e^{-\|f-g\|^2/\sigma^2}$$

- probability of observing pixels
  - assumes discrete pixel values
  - equal likelihood
  - equal normal error

Solve: Operator Decomposition

"small"

$$\mathcal{H}(\phi_i) + \alpha I = A + B$$

- block diagonal with sparse banded blocks
- linear-time Cholesky decomposition

$$A = LL^T$$

blocks = image rows

- leads to formal expression

$$\begin{aligned}(A+B)^{-1} &= (LL^T + B)^{-1} \\ &= L^{T^{-1}}(\mathbb{I} + L^{-1}BL^{T^{-1}})^{-1}L^{-1}\end{aligned}$$

Calculate Using Fast Matrix-Vector Ops

(1) calculate $L^{-1}(-\nabla\phi_i)$ by back-solving in $\mathcal{O}(nN)$ operations
(2) calculate $L^{T^{-1}}L^{-1}(-\nabla\phi_i)$ by back-solving in $\mathcal{O}(nN)$ more operations
(3) save result
(4) calculate $BL^{T^{-1}}L^{-1}(-\nabla\phi_i)$ using the fast computation for B
(5) calculate $L^{-1}BL^{T^{-1}}L^{-1}(-\nabla\phi_i)$ by back-solving in $\mathcal{O}(nN)$ more operations
(6) calculate $L^{T^{-1}}L^{-1}BL^{T^{-1}}L^{-1}(-\nabla\phi_i)$ by back-solving in $\mathcal{O}(nN)$ more operations
(7) subtract from result
... continue to the order of truncation.

linear in problem size

plus (poly-order)(fast algorithm)

Row by Row

- each block corresponds to a row
- each block can be calculated in parallel
- (number-rows)-way parallelism

Even Better: Use Minimax Polynomial

$$\min_{p \text{ polynomial}} \max_{x \in \operatorname{spec}(L^{-1} B L^{T\,-1})} \left\| \frac{1}{1+x} - p(x) \right\| = \epsilon$$

- calculate

$$L^{T^{-1}} p(L^{-1} B L^{T^{-1}}) L^{-1} (-\nabla \phi_i)$$

Convergence

next error

$$
\begin{aligned}
\|x+\Delta x - x_0\| &= \left\|x + p(A^{-1}B)A^{-1}(-\mathcal{H}(x-x_0)) - x_0\right\| \\
&\le \left\|p(A^{-1}B)A^{-1}(-\mathcal{H}(x-x_0)) - (\mathcal{H}+\alpha I)^{-1}(-\mathcal{H}(x-x_0))\right\| \\
&\quad + \left\|(\mathcal{H}+\alpha I)^{-1}(-\mathcal{H}(x-x_0)) - (x-x_0)\right\| \\
&\le \left(\epsilon\,\|\mathcal{H}\| + \left\|(\mathcal{H}+\alpha I)^{-1}\right\|\alpha\right)\|x-x_0\|.
\end{aligned}
$$

minimax polynomial error

step shortening

current error

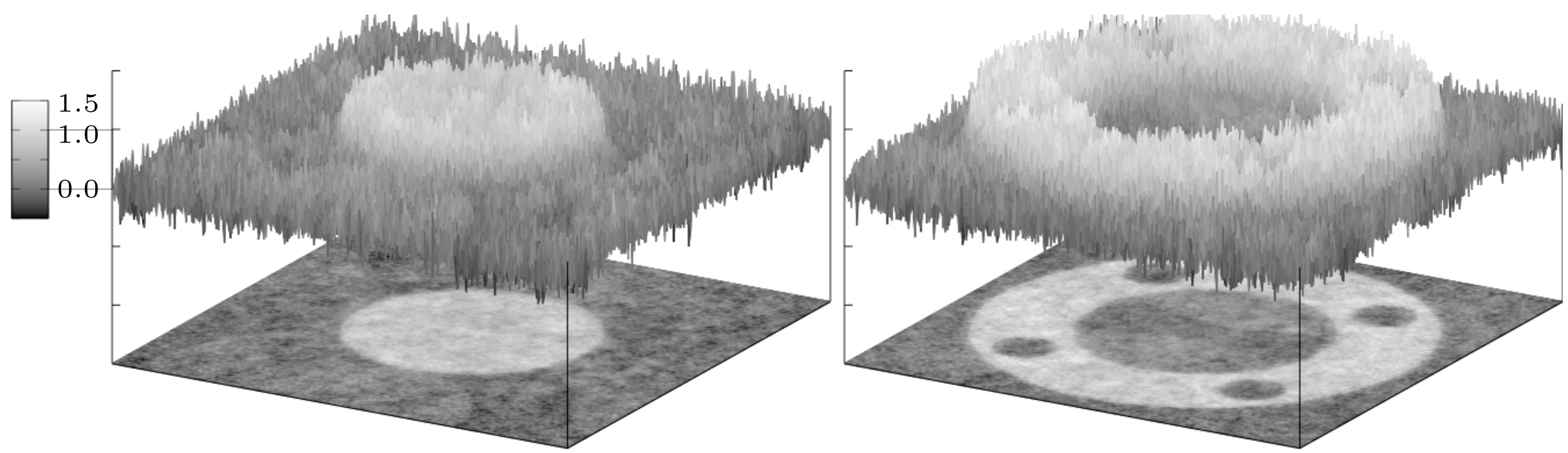
Safe in Single-Precision

- recalculate gradient at each outer iteration
- numerical error only builds up during polynomial evaluation
  - coefficients well-behaved (and in our control)

Numerical Tests

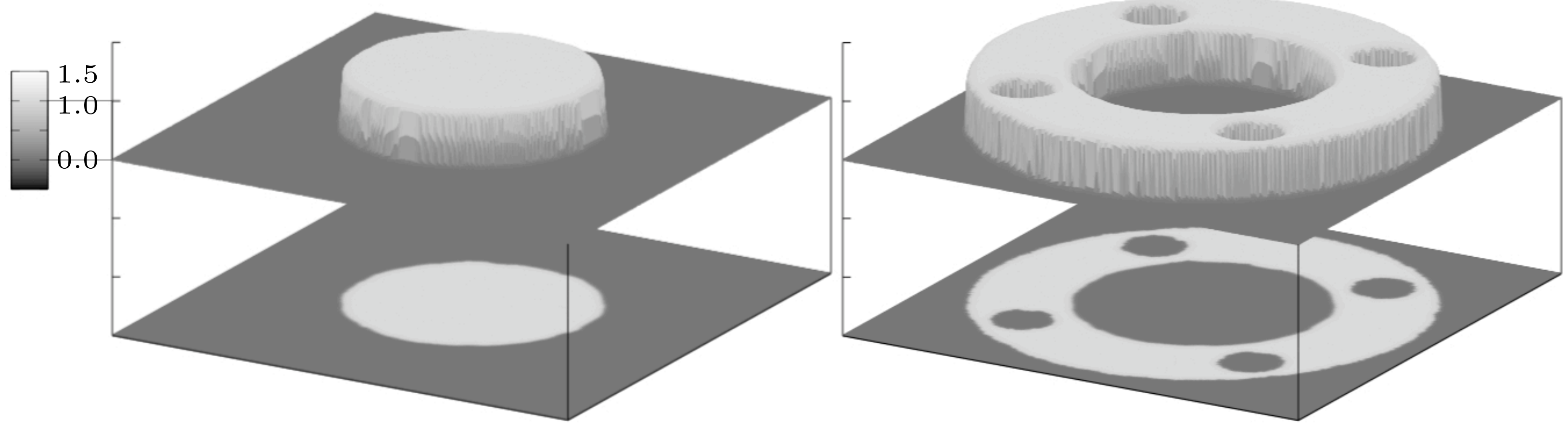
- start with quadratic penalties
- add nonlinear penalties and change weights
- with and without time fixed budget for computation

10 iterations



1. 10 iterations with bi2 regularization

100 iterations

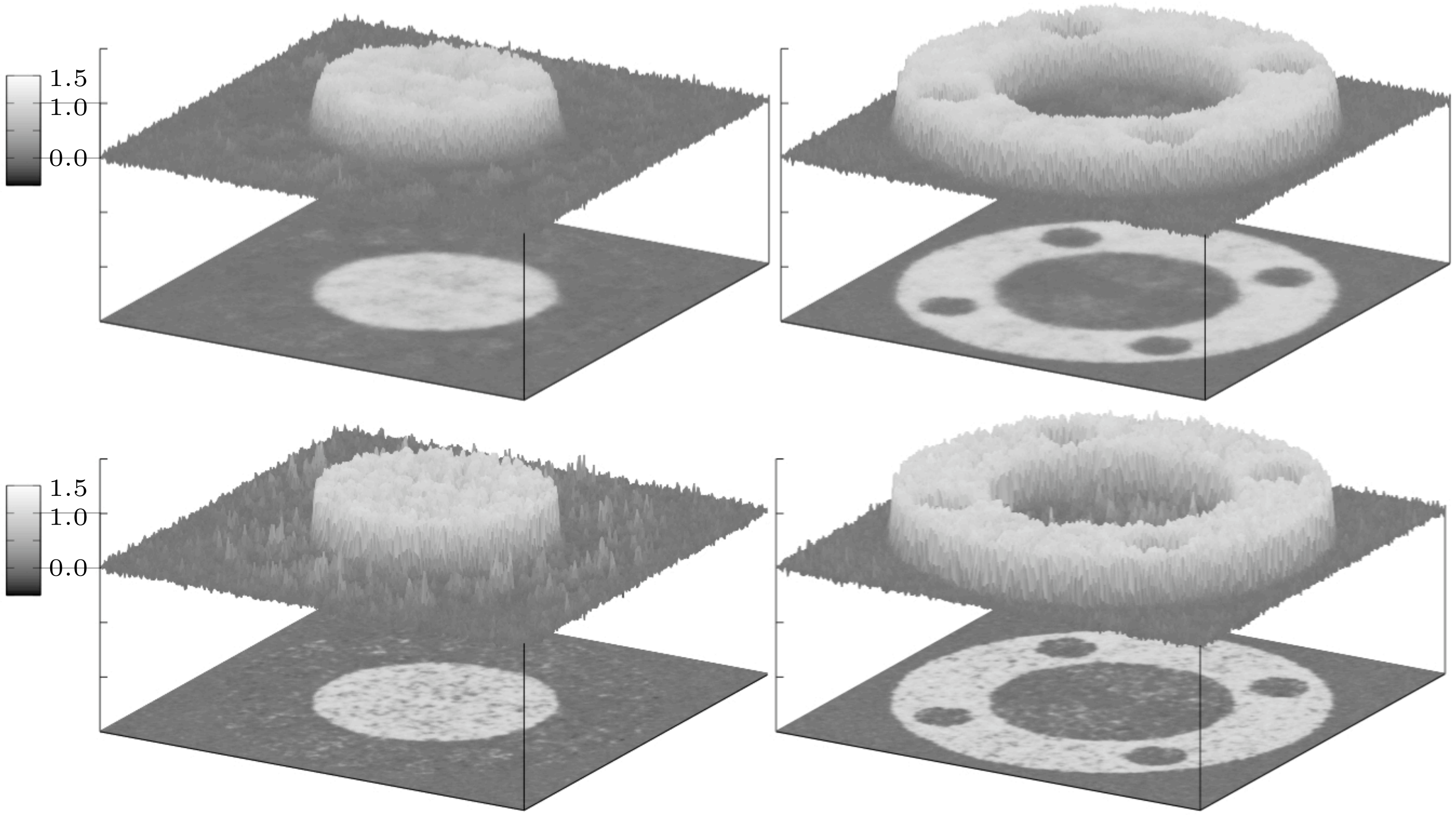


I. 10 iterations with bi2 regularization

II. introduce other penalties

1. masking
2. magnet
3. segmentation

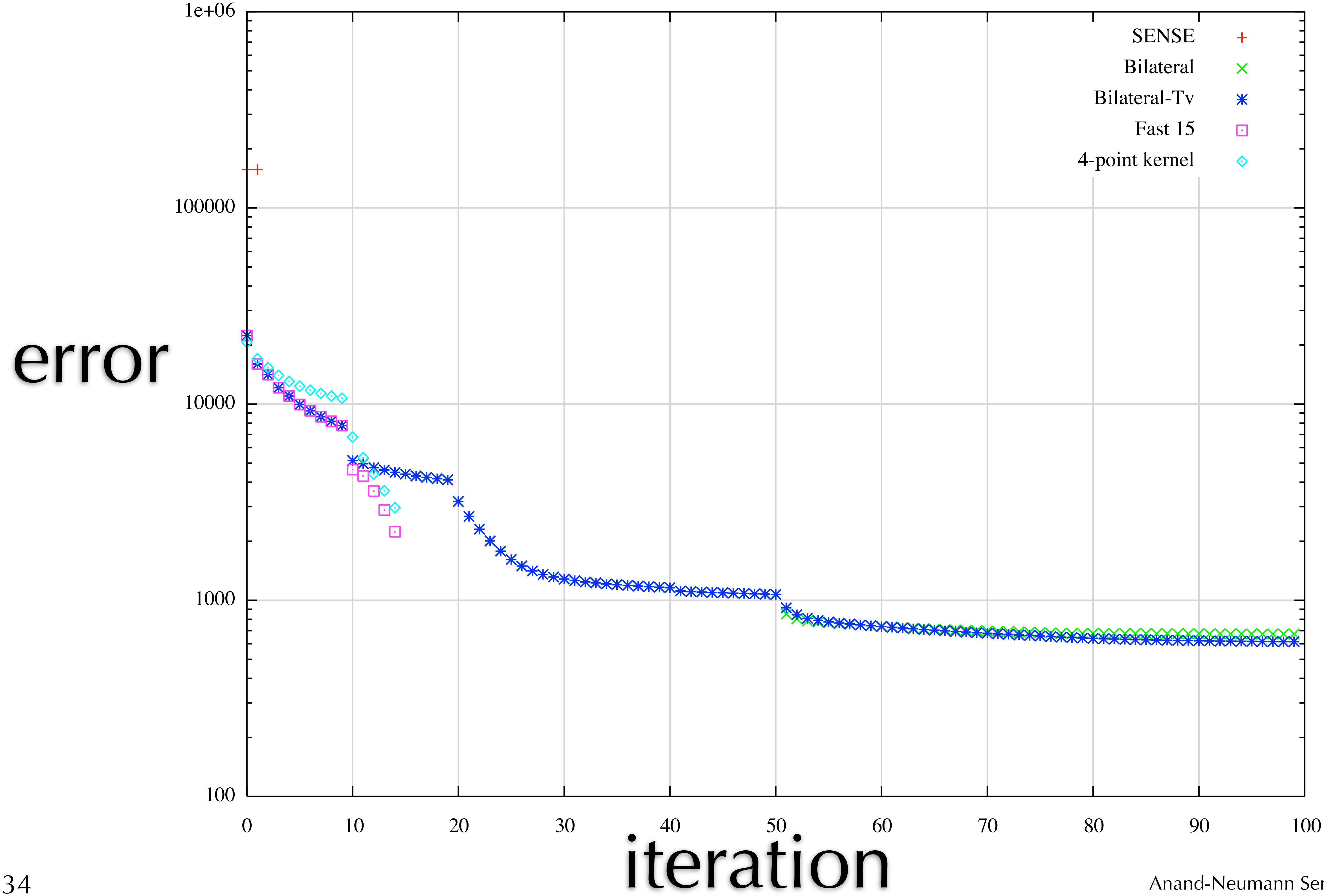
15 iterations



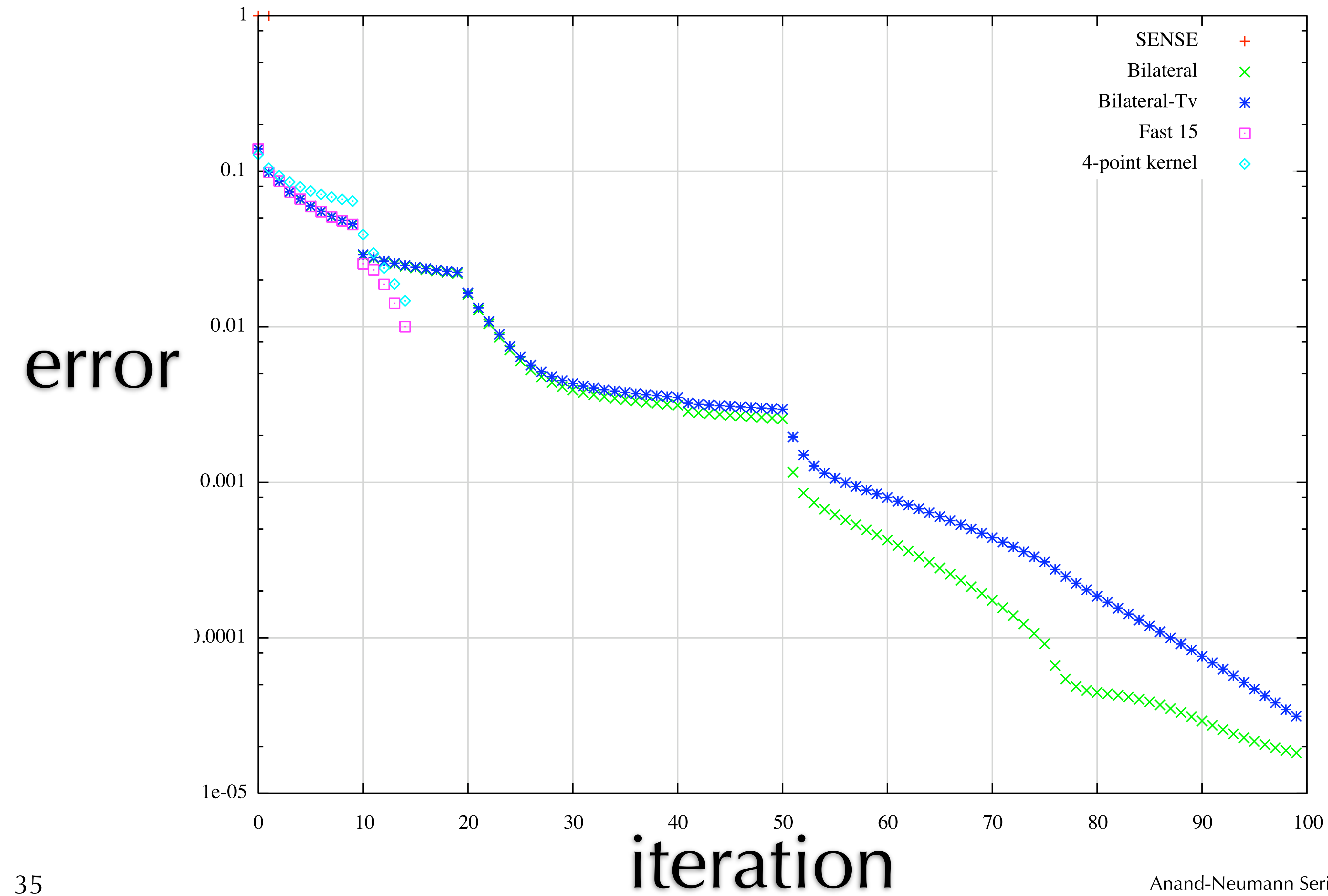
optimized c

simple c

Absolute Error



Relative (to limit) Error



Conclusion

- highly-parallel
- safe in single precision
- robust with respect to noise
- accommodates nonlinear penalties

Thanks to:

students and colleagues in the

Advanced Optimization Laboratory of McMaster University

Inspiring Innovation and Discovery