Numerical Methods for Tomography

William W. Symes

The Rice Inversion Project
Department of Computational and Applied Mathematics
Rice University

Mathematical Geophysics Summer School
Stanford University, August 1999

Prologue: Tomography defined

Fixed and known: m source-receiver pairs $(\mathbf{s}_1, \mathbf{r}_1, \mathbf{s}_2, \mathbf{r}_2, ..., \mathbf{s}_m, \mathbf{r}_m) \in \mathbf{R}^{6m}$

User-selected: "Model Space" = set $\mathcal{M}$ of positive real-valued functions u on closed cube containing all sources and receivers; $\mathcal{M} \subset C^0$

Minimal traveltime $T[u]_i$ from s_i to r_i, corresponding first arrival raypath Ray_i:

$$T[u]_i = \int_{\mathrm{Ray}_i} u\,ds$$

Defines *forward map* $T : \mathcal{M} \to \mathbf{R}^m$

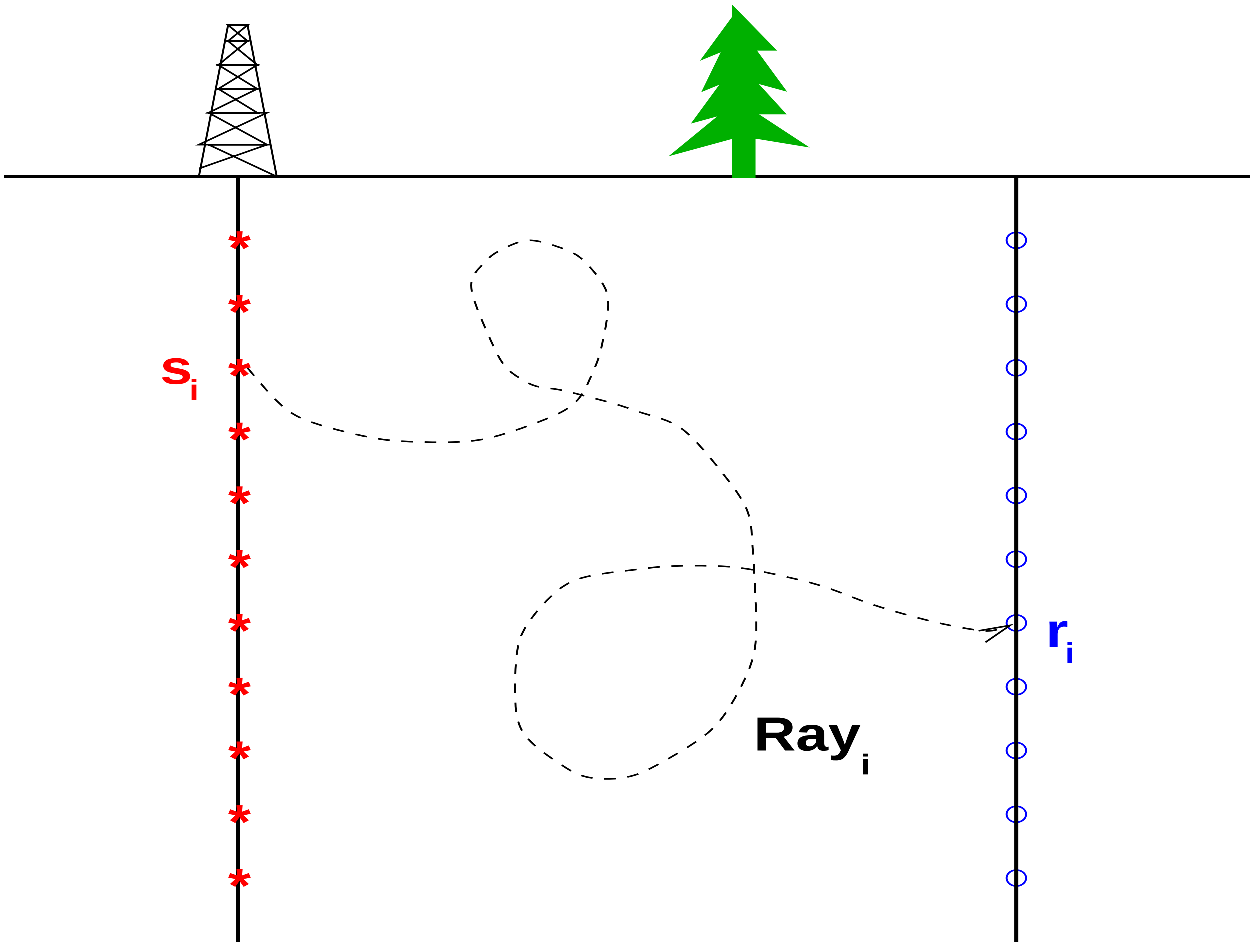

Figure 1: Tomography components: sources s_i, receivers r_i, first arrival ray path Ray_i

Prologue: Tomography defined

First arrival traveltime tomography problem: **Given** $t \in \mathbf{R}^m$, **find** $u \in \mathcal{M}$ so that

$$T[u] \simeq t$$

[precise statement must include meaning of "$\simeq$", definition of $\mathcal{M}$]

F. A. linearized T. T. P.: **Given** $t \in \mathbf{R}^m$, also $u_0 \in \mathcal{M}$, **find** $\delta u \in T_{u_0}\mathcal{M}$ so that

$$DT[u]\delta u \simeq \delta t = t - T[u_0]$$

[must also understand diff. of T, nature of $T\mathcal{M}$]

Outline

1. Linear Tomography
   - 1.1 Discretization
   - 1.2 Linear = Linearized
   - 1.3 Error measures
   - 1.4 Least squares
   - 1.5 Conjugate gradient iteration
   - 1.6 Regularization
2. A (very) few words about (Nonlinear) Tomography

Linear Tomography

Discretization

$$T(\mathbf{x}_s, \mathbf{x}_r) = \int_{\mathrm{Ray}} ds \; u$$

where $u = 1/v$ is the slowness, and the integral is over ray (spatial part) parametrized by arc length s

Basic discretization trick: consider only $u =$ a finite expansion in basis functions: ϕ_i

$$u(\mathbf{x}) = \sum_{i=1}^{n} u_i \phi_i(\mathbf{x})$$

[Expl: $\phi_i =$ characteristic functions of grid cells, cf. Berryman lectures.]

Discretization

Approximate T at m source-receiver pairs $t_j = T(\mathbf{x}_s^j, \mathbf{x}_r^j), j = 1, ..., m$:

$$t_j = \sum_i M_{ji} u_i, \text{ i.e. } t = Mu$$

(slightly abusing notation by using u to denote the vector of coefficients u_i), where M is the *ray path matrix*

$$M_{ji} = \int_{\mathrm{Ray}_j} \phi_i$$

Ray_j being the ray connecting $\mathbf{x}_s^j$ to $\mathbf{x}_r^j$

Will talk about this or equivalent system in these lectures.

Linear = Linearized

After discretization, traveltime tomography problem becomes:

Given a data vector t, find a model vector u so that u predicts t, at least approximately:

$$t \simeq Mu$$

This *looks* like a linear problem...

but it's not, since the rays depend on u and therefore so does M. It's still really

$$t \simeq M[u]u = T[u]$$

Linear = Linearization

Standard dodge: "assume M (i.e. rays) known".

Claim: This is equivalent to *linearization*.

In fact $M = M[u_0]$ for some "reference" slowness u_0; let t_0 be the vector of times predicted by u_0. Then

$$t = t_0 + t - t_0 = Mu_0 + M(u - u_0)$$

so solving $t = Mu$ is equivalent to solving $t - t_0 = M(u - u_0)$. In fact this is the linearization of the slowness $\rightarrow$ traveltime map around u_0.

Linearization of T

$u = u_0 + h\delta u$ for *slowness perturbation* δu, small parameter h.

$$T(\mathbf{x}_s, \mathbf{x}_r) - T_0(\mathbf{x}_s, \mathbf{x}_r) = \int_{\mathrm{Ray}} u - \int_{\mathrm{Ray}_0} u_0$$

$$= \int_{\mathrm{Ray}} u - \int_{\mathrm{Ray}_0} u + h \int_{\mathrm{Ray}_0} \delta u$$

Fermat's principle states that $\left| \int_{\mathrm{Ray}} u - \int_{\mathrm{Ray}_0} u \right|$ is o of the distance between the ray paths.

Linearization of T

If we knew that this distance were $O(h)$, then the first two terms above are $o(h)$ and

$$T(\mathbf{x}_s, \mathbf{x}_r) = T_0(\mathbf{x}_s, \mathbf{x}_r) + h\delta T(\mathbf{x}_s, \mathbf{x}_r) + o(h),$$

$$\delta T(\mathbf{x}_s, \mathbf{x}_r) = \int_{\mathrm{Ray}_0} \delta u$$

Discretizing as before, get "known M" expression:

$$\int_{\mathrm{Ray}_0} \delta u \simeq M[u_0]\delta u$$

Dotting i's and crossing t's

The tomography literature *assumes* that T is differetiable as a function of u.

Good Excercise: Use the method of characteristics to show that, with sufficiently strong norm topology on u's, the map $u \rightarrow$ (T at finitely many points) is indeed differentiable provided that the source and receiver points are close enough for the method of characteristics to succeed (no multiarrivals).

Dotting i's and crossing t's

Good Research Problem: Show global differentiability of first arrival traveltimes, or upset the field with a counterexample.

Even Better Research Problem: Understand how differentiability interacts with convergence of discretization. [Delprat-Jannaud and Lailly: must use a stronger norm than L^2 in domain (models), since otherwise formal linearized map isn't even bounded!]

Error Measures

For brevity, for the rest of this section
$\delta u \rightarrow u, \delta T \rightarrow T, \delta t \rightarrow t$.

Discrete(-ized) linear(-ized) tomography problem: given $t \in \mathbf{R}^m$, find $u \in \mathbf{R}^n$ so that

$$t \simeq Mu$$

"$\simeq$" is recognition that problem is likely *overdetermined*, i.e. $\operatorname{rank} M < m$, and data is *inconsistent* (not in range of M) with probability 1.

Error Measures

Quantifying "$\simeq$": choose *error measure* function $E : \mathbf{R}^m \to \mathbf{R}^+$, solve *optimizaton problem*: find u so that

$$E(t - Mu) = \min_{\tilde{u} \in \mathbf{R}^n} E(t - M\tilde{u})$$

Interesting choices for E: pth powers of l^p norms, i.e.

$$E(t) = \sum_{j=1}^{m} |t_j|^p$$

Of these, *only* the choice $p = 2$ yields a differentiable error function, so will be concerned exclusively with resulting *least squares inverse problem*.

Error Measures

NB: Error measures other than least squares have interesting and useful properties, particularly l^p with $1 \leq p < 2$ ("robust" measures) - see references mentioned in Berryman's notes for this important topic. None of them yield optimization problems equivalent to linear systems, as does least squares.

Another **NB:** Error measures correspond to statistical hypotheses about data error (cf. GP's Wed. lecture) - least squares $\sim$ max likelihood (Gaussian noise with known covariance); $l^1 \sim$ "long-tailed" noise PDF,... Will not pursue this here - see eg. GP, Stark,....

Least Squares

$$E(t) = \sum_{j=1}^{m} t_j^2 = t^T t$$

Recall that

$$\nabla_u E(t - Mu) = M^T(t - Mu)$$

(if this isn't a "recall", it's an *excercise!*) Stationary point condition = "normal equations"

$$M^T M u = M^T t$$

and Hessian matrix $M^T M$ is positive semidefinite, so finding stat pts = mins $\sim$ solving a linear system.

Least Squares

Unfortunate tendency: tomography problem both overdetermined, *underdetermined* (M has nontrivial null space) and even worse (singular values clustering at 0).

SVD analysis:

$M = U\Sigma V^T,\ U^TU = UU^T = I^{m\times m},\ V^TV = VV^T = I^{n\times n}$

Pseudoinverse solution = solution of normal equations of least norm

$$M^\dagger t = V\Sigma^\dagger U^T t = \sum_{\sigma_j \neq 0} \frac{(U_j^T t)V_j}{\sigma_j}$$

$\Rightarrow V_j^T u$ blows up when σ_j small relative to $U_j^T t$

Least Squares - Direct Solution

Variants of *QR algorithm* $\Rightarrow$ SVD, pseudoinverse of M in $O(\max(n,m)^3)$ flops

Matlab command: `pinv`

Sample implementation: `tomodir.m` implements straight ray tomography in 2D (i.e. $u_0 \equiv$ const.) for *crosswell configuration*.

User inputs: source, receiver depth vectors `zs, zg`, cell height, width `dz, dx`, vert., hor. numbers of cells `nz, nx`, traveltime (perturbation) data vector `t`

Uses `pinv` to return a slowness model of size `nz` by `nx`.

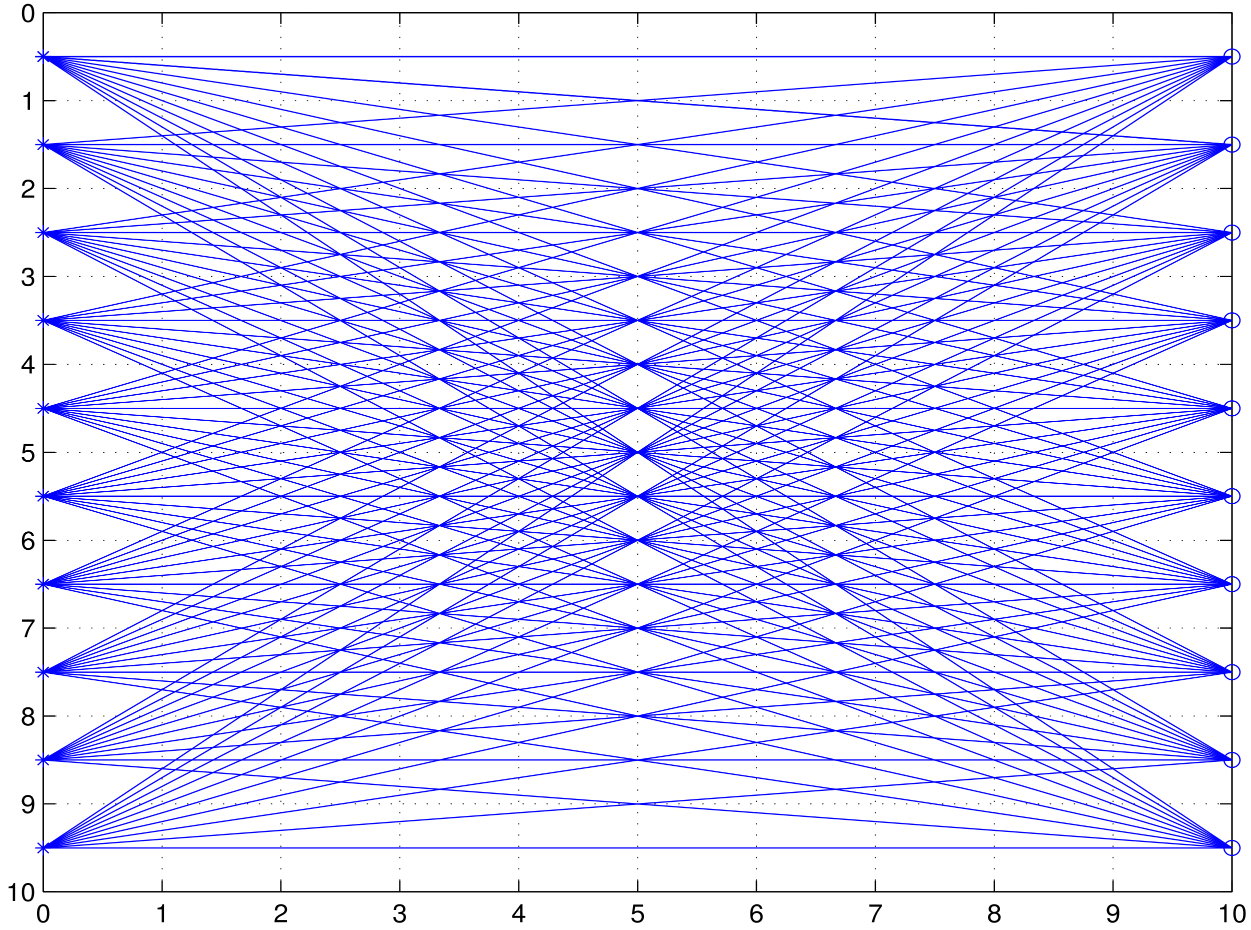

Figure 2: A straight ray "crosswell" tomographic example: 10 sources in left "well", 10 receivers in right "well", 10×10 constant slowness cells

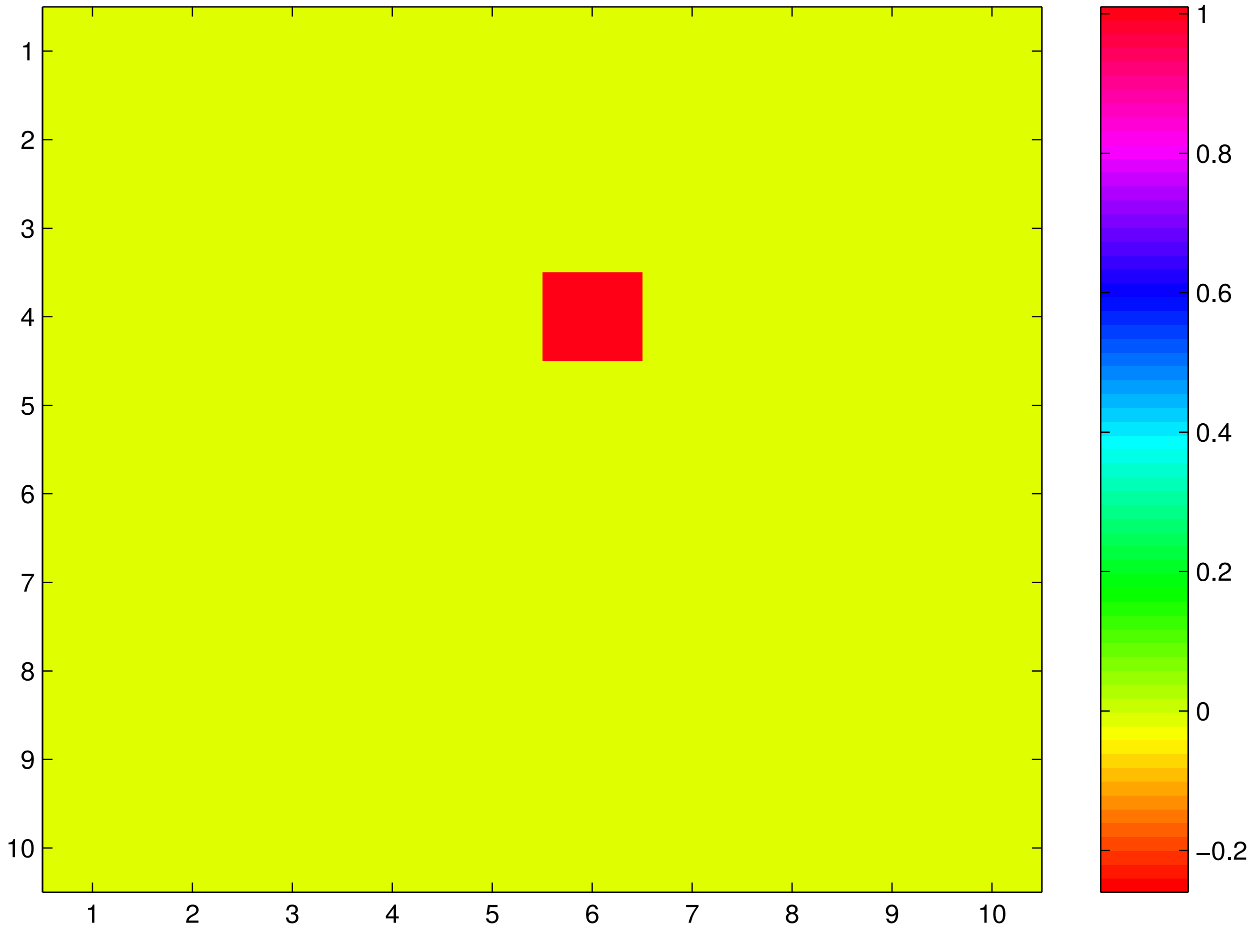

Figure 3: Target slowness perturbation, used to create data

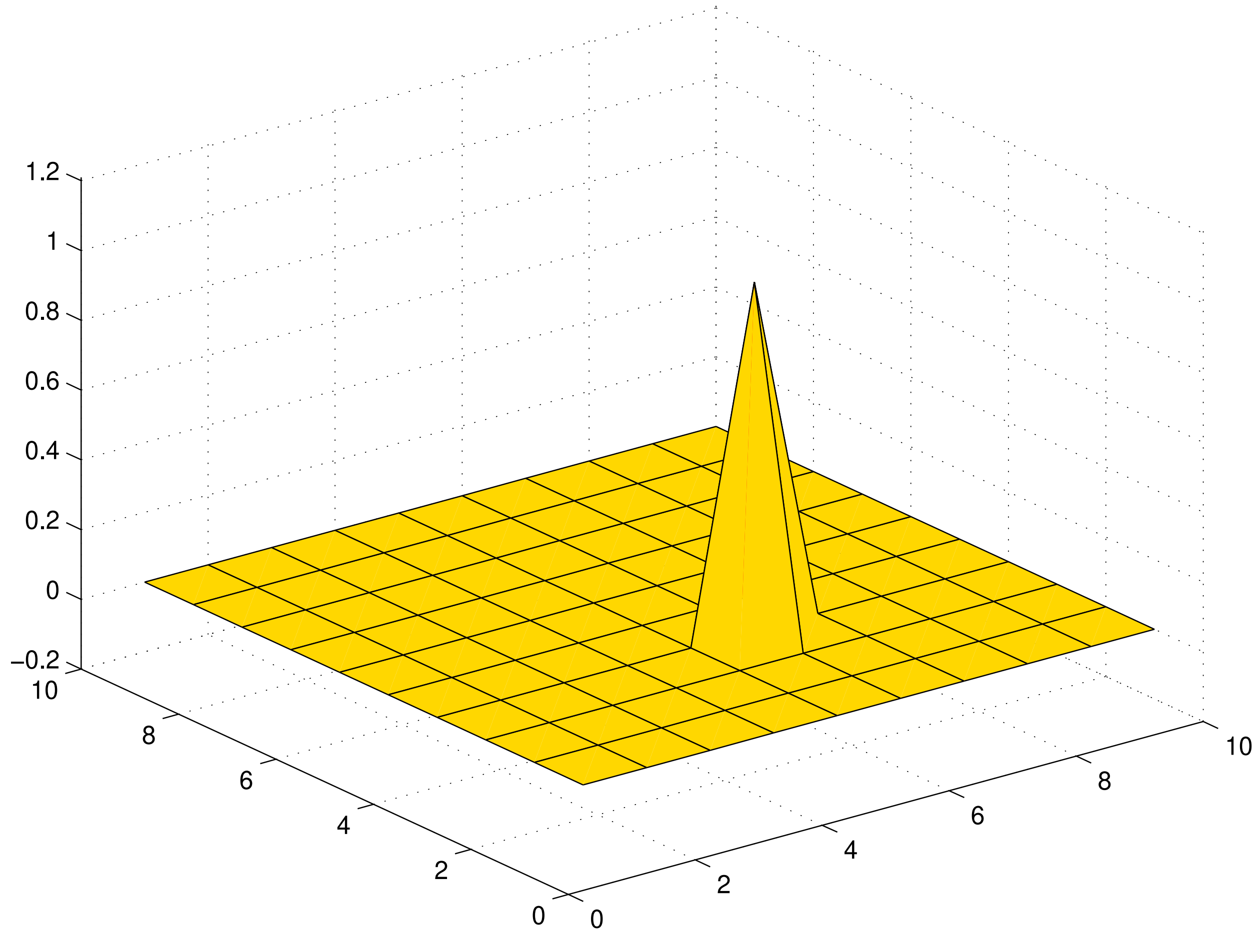

Figure 4: Perspective plot of target perturbation

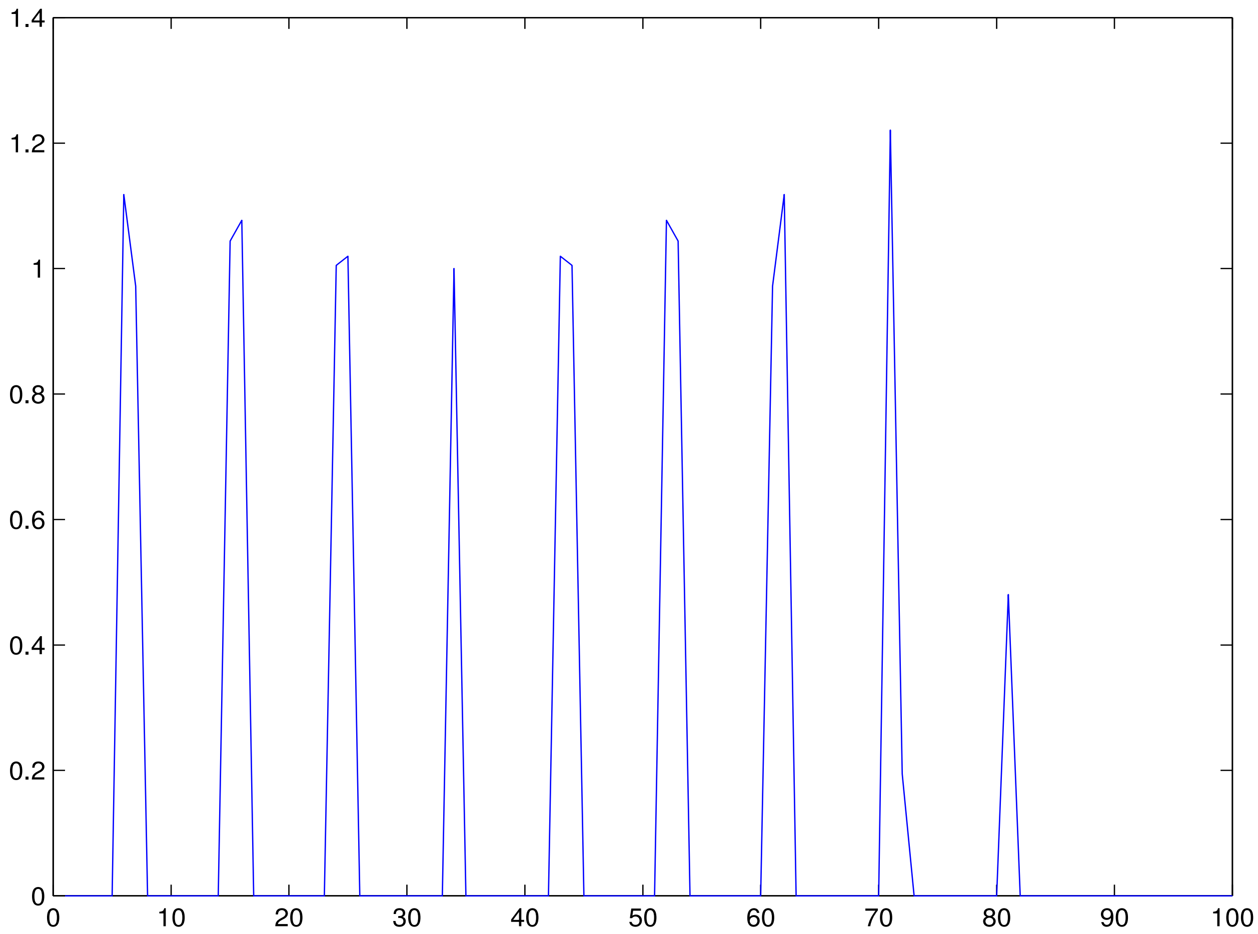

Figure 5: Traveltime perturbation data (noise free)

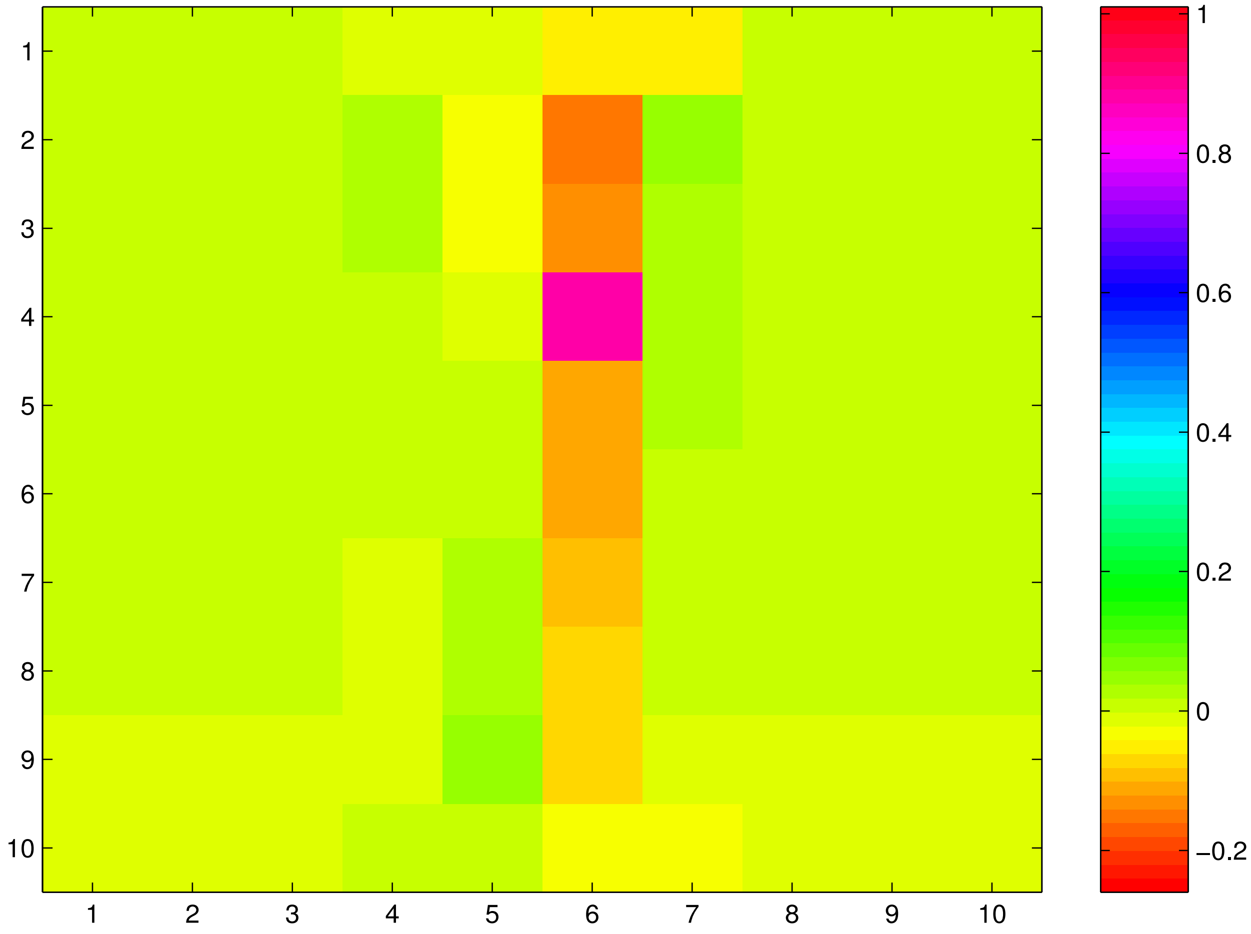

Figure 6: Slowness perturbation estimated *via* pseudoinverse

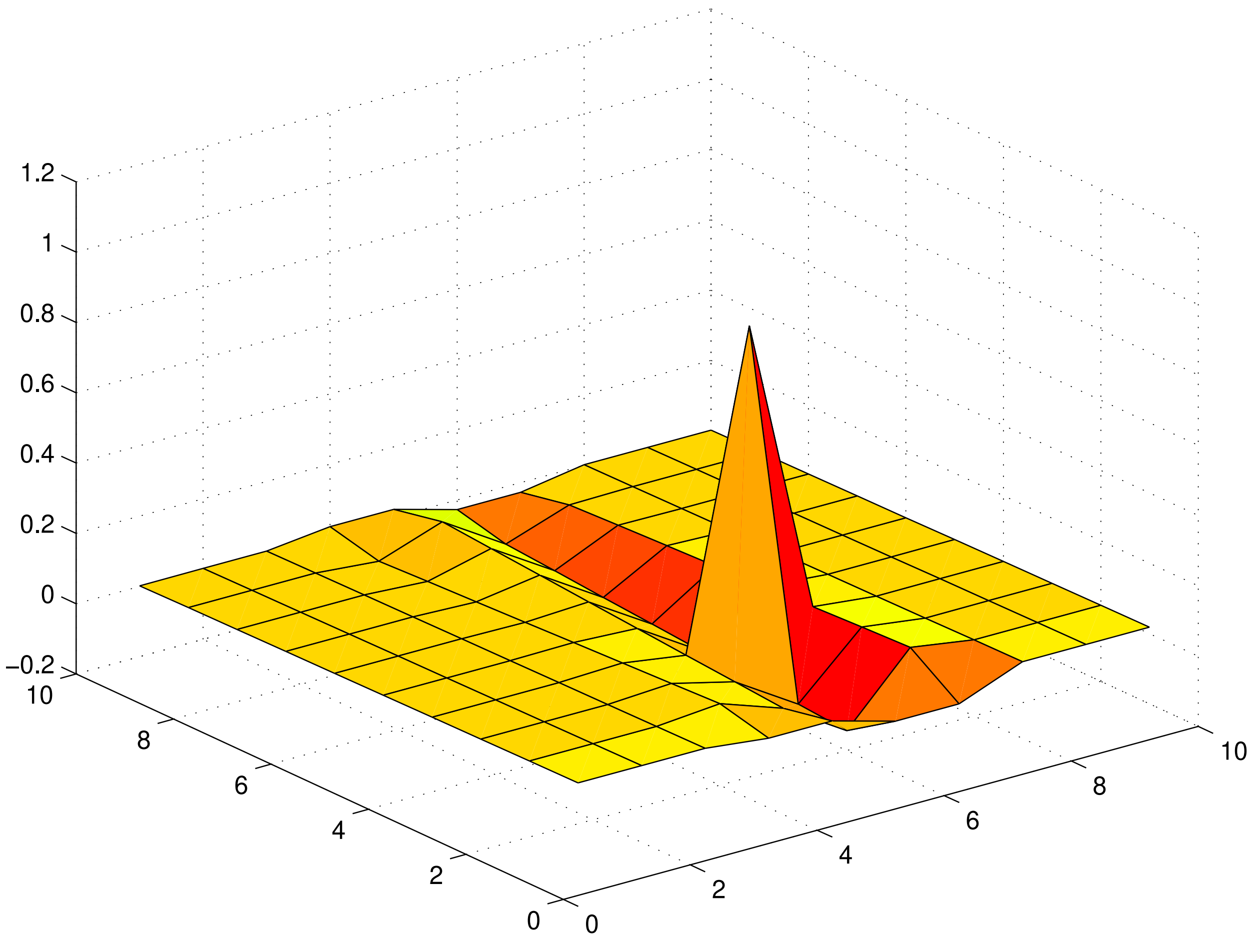

Figure 7: Perspective plot of pseudoinverse estimate

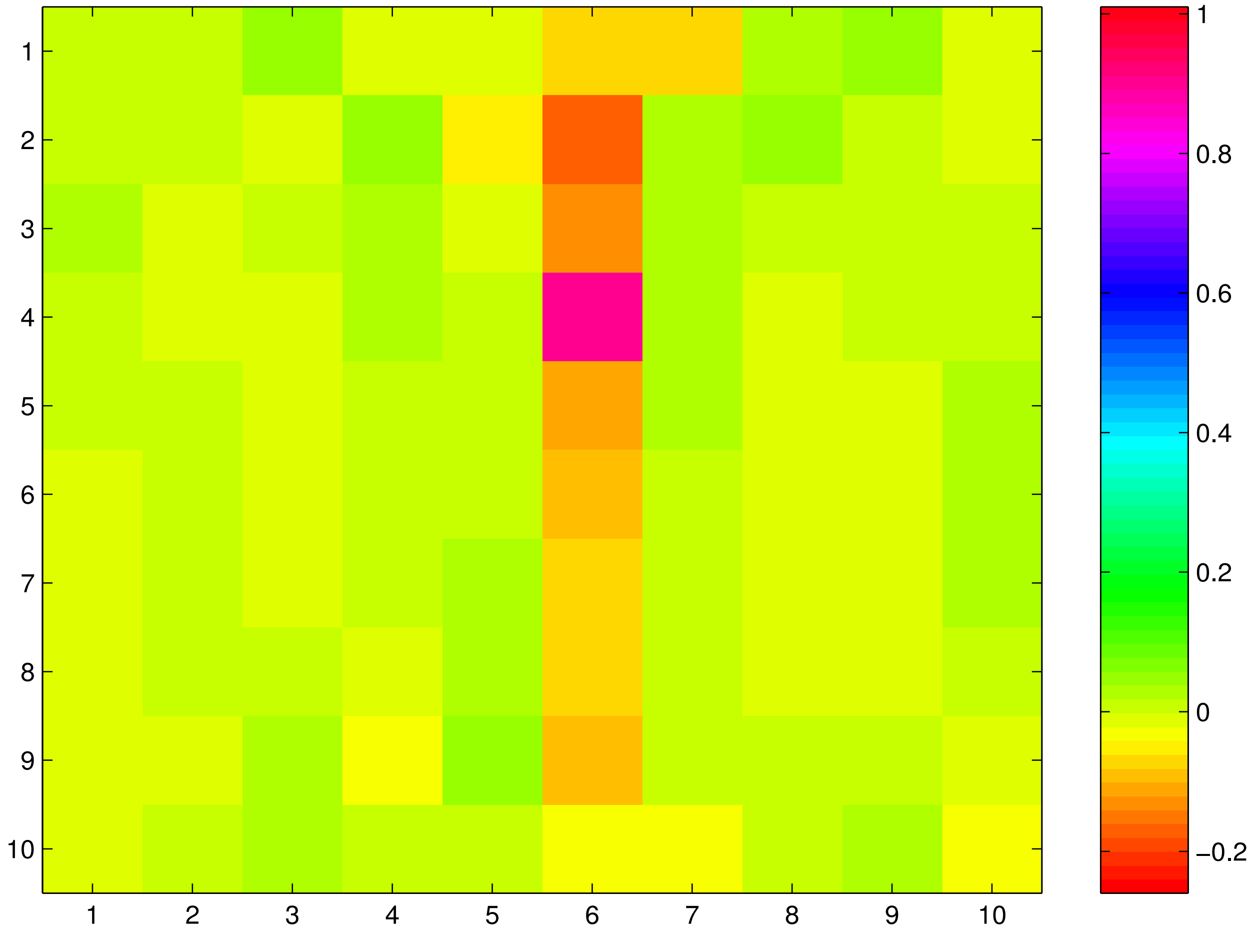

Figure 8: Effect of 1% noise: pseudoinvers slowness estimate

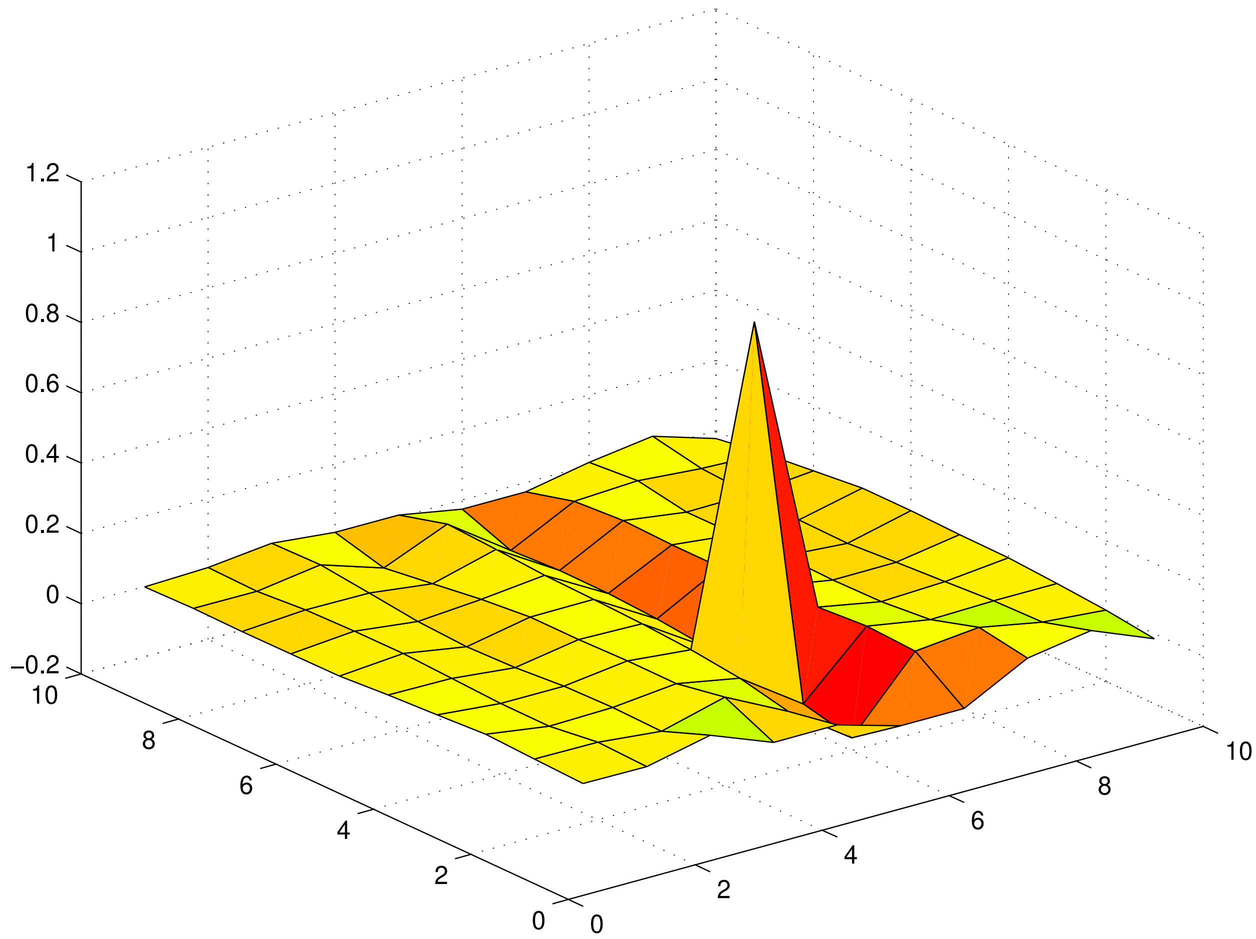

Figure 9: Perspective plot of pseudoinverse estimate, 1% noise

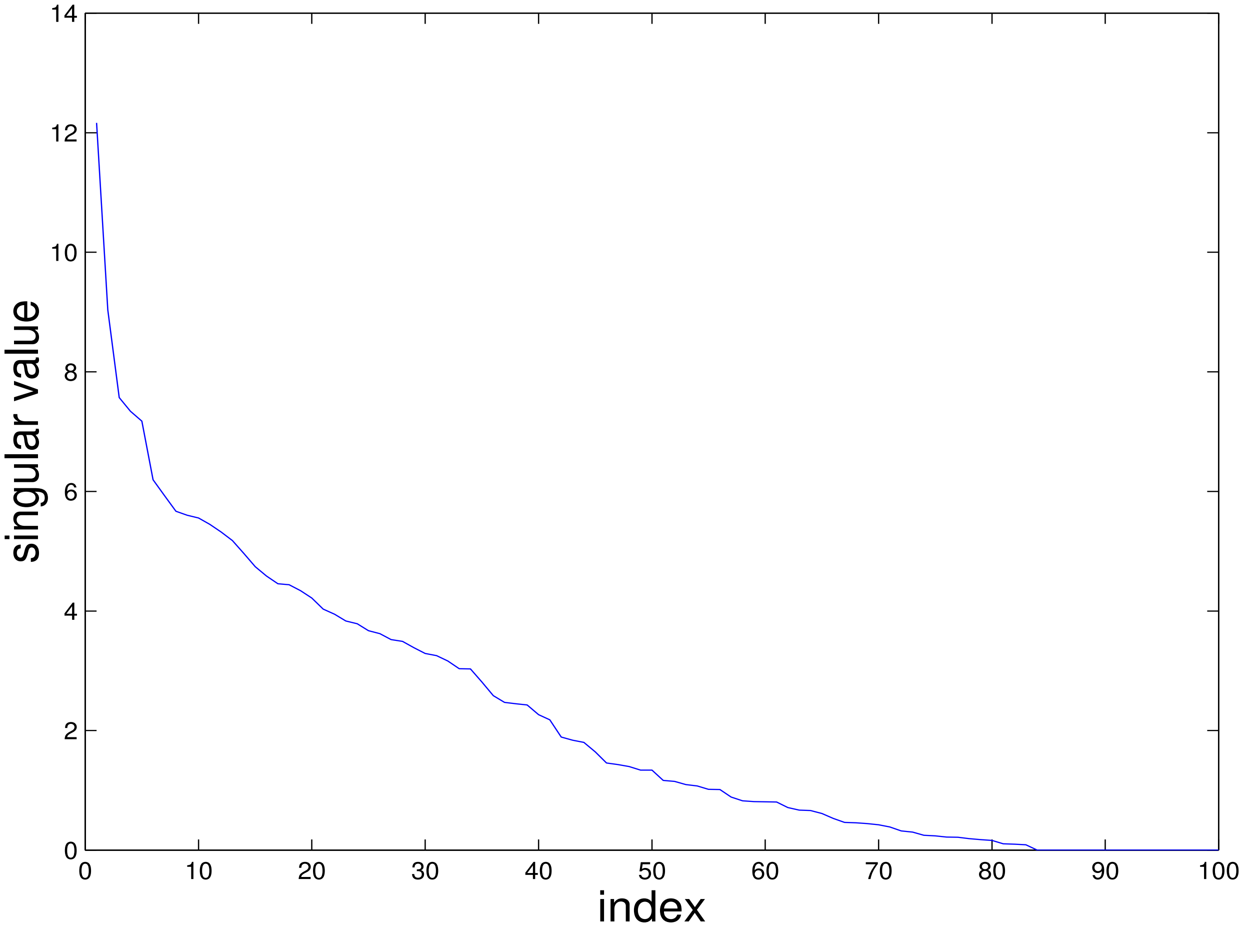

Figure 10: Singular values vs. index, decreasing order

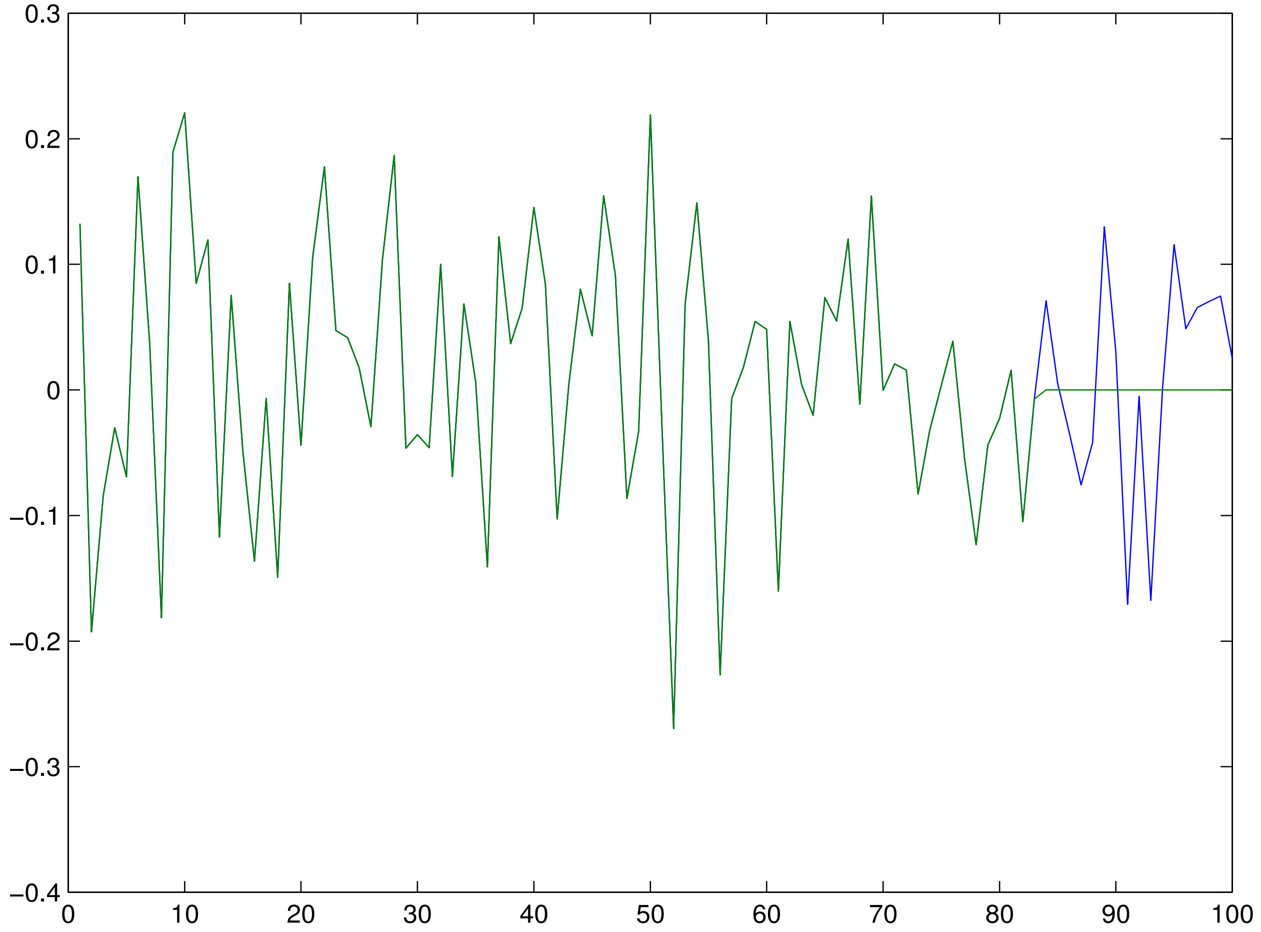

Figure 11: Coefficients of target (blue) and pseudoinverse estimate (green) slowness fields in singular vector expansion

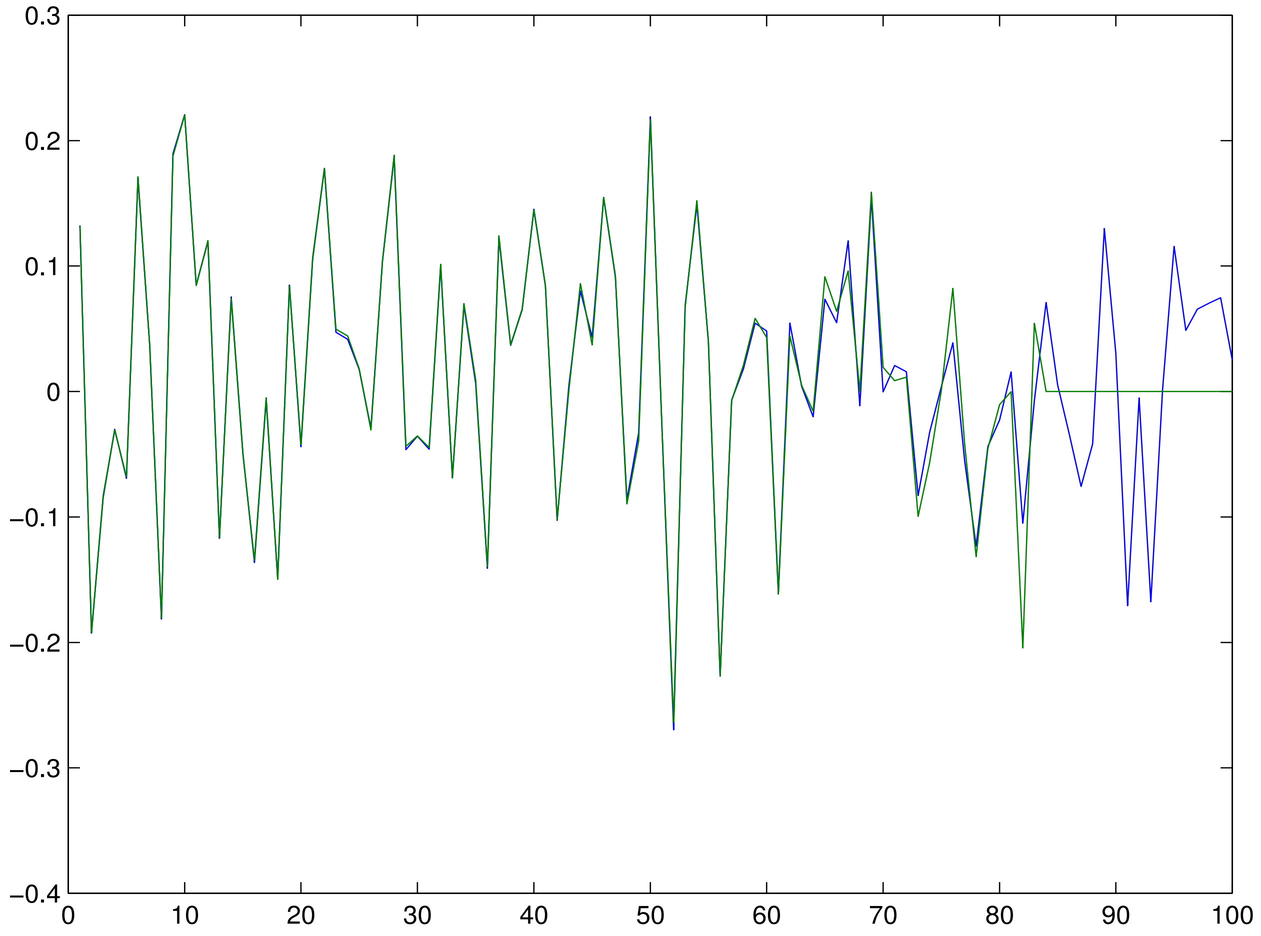

Figure 12: Coefficients of target (blue) and pseudoinverse estimate (green) slowness fields in singular vector expansion, 1% noise

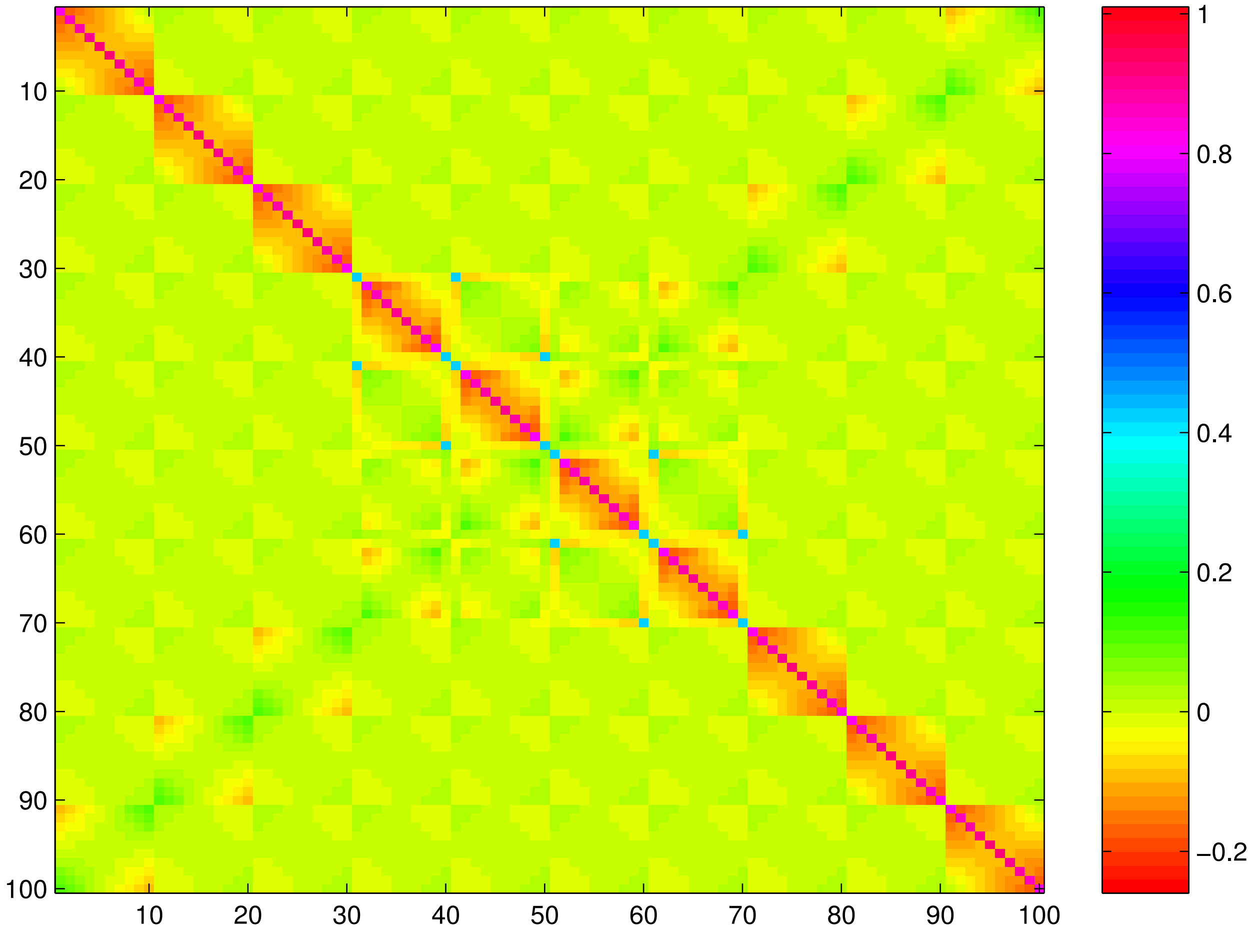

Figure 13: Resolution matrix for pseudoinverse estimator, $M^\dagger M$

Spectral Analysis

Bube and others (see reference list): null space of crosswell tomography linearized forward map M contains

- "stripers" (constant in z, oscillate in x with zero mean)
- "slopers" (linear in z, oscillate in x)
- many other polynomials in z, x

Both large and small singular values associated to high wavenumbers in z

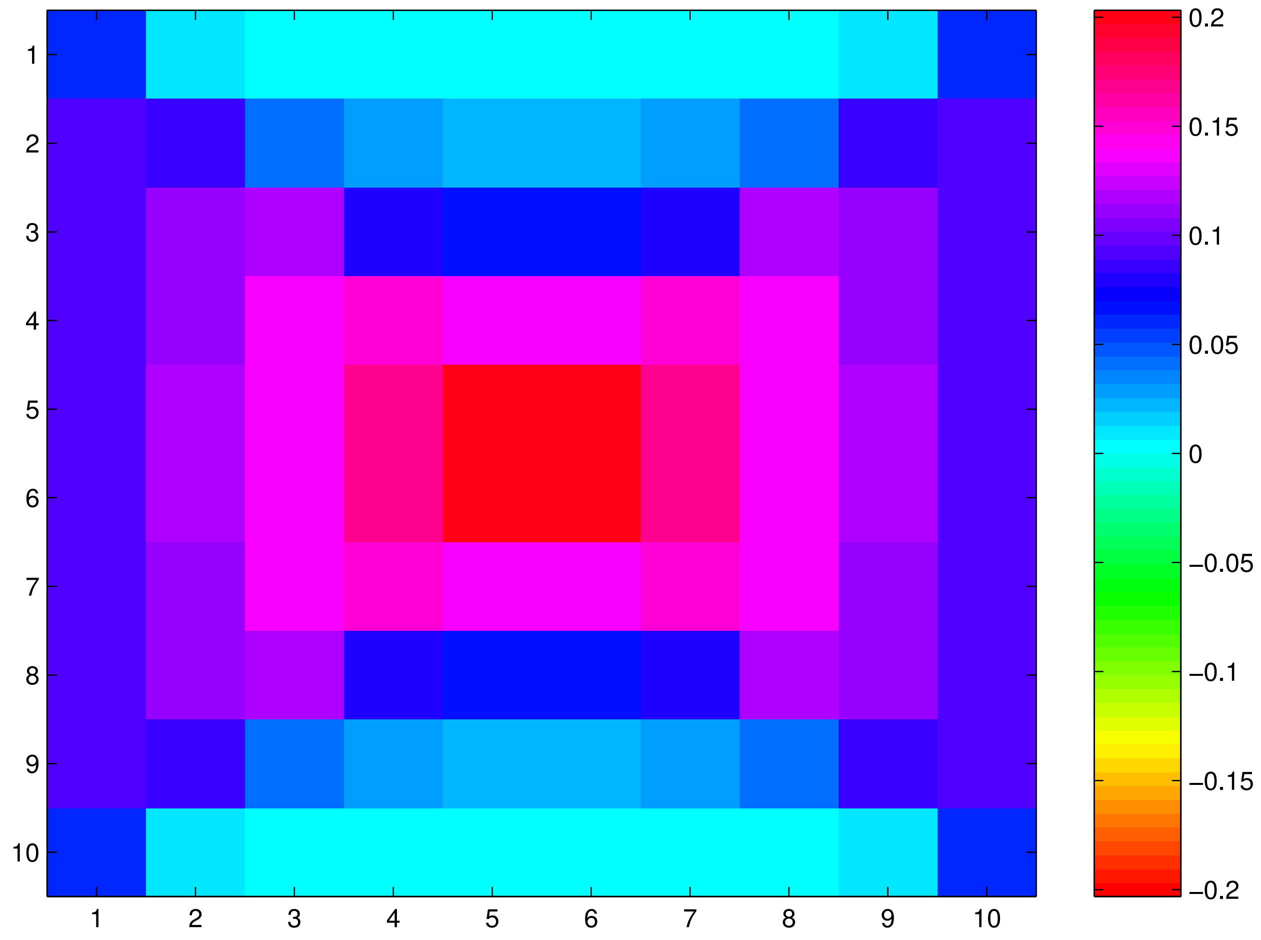

Figure 14: Singular vector 1 (large singular value)

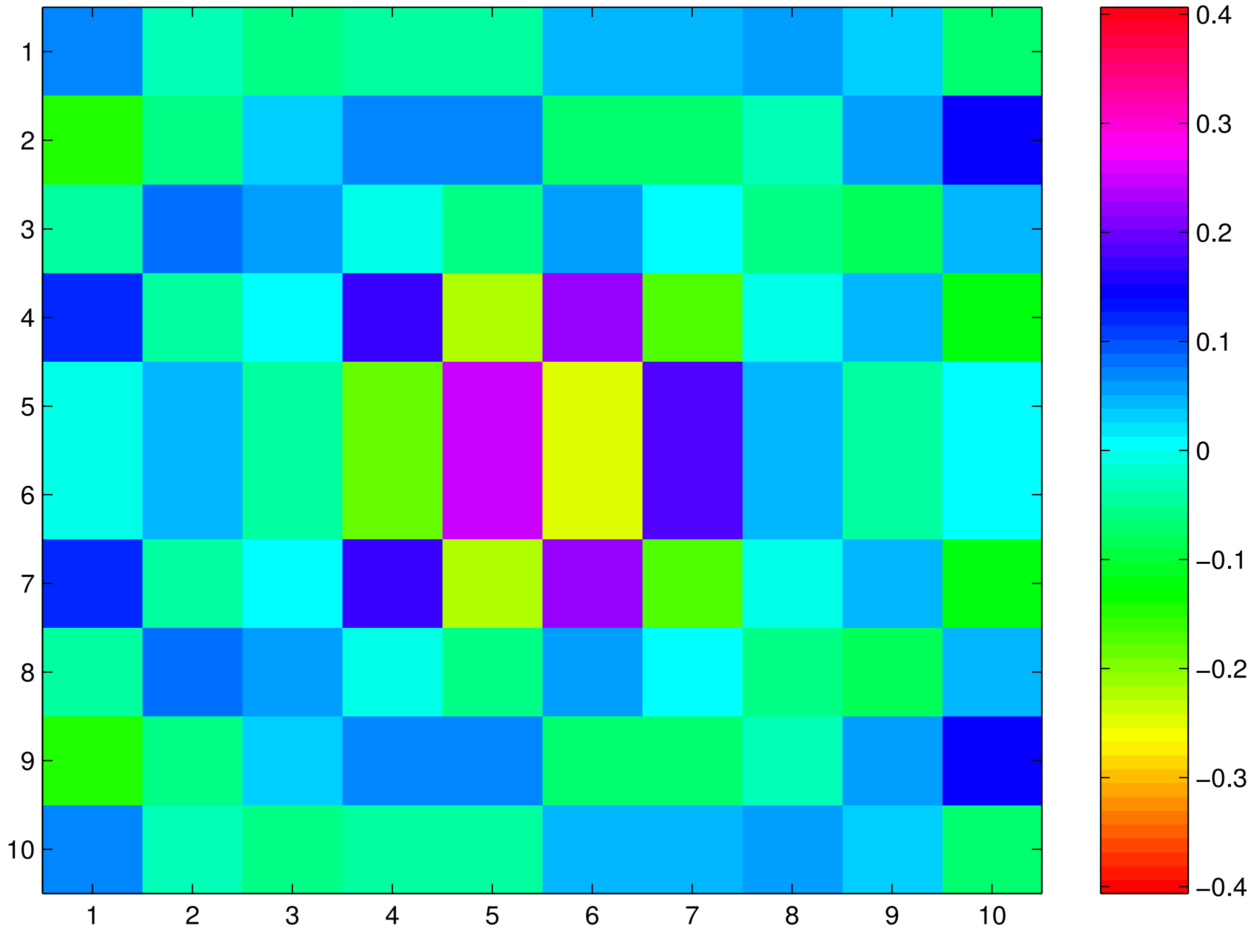

Figure 15: Singular vector 50 (small singular value)

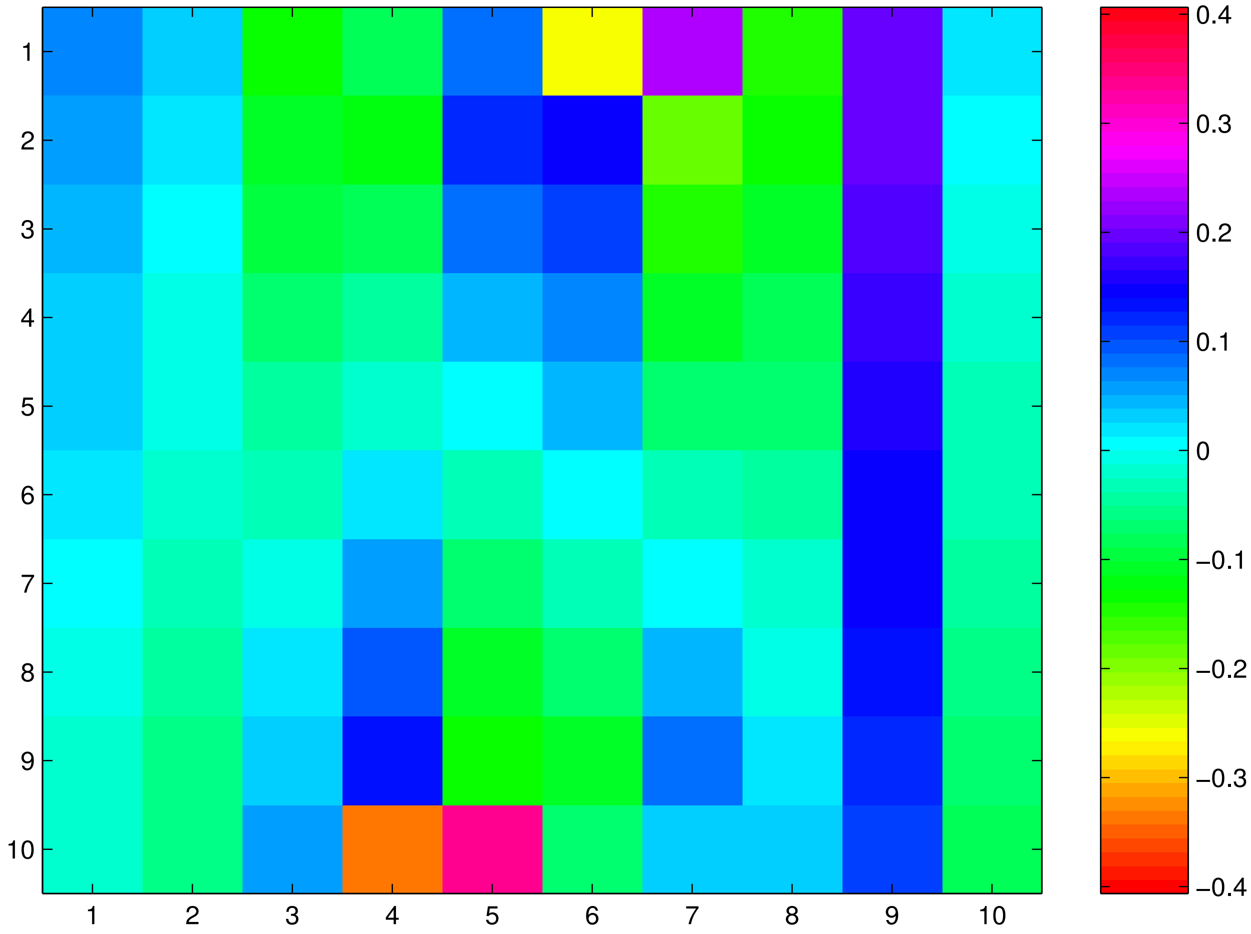

Figure 16: Singular vector 99 (zero singular value - null vector)

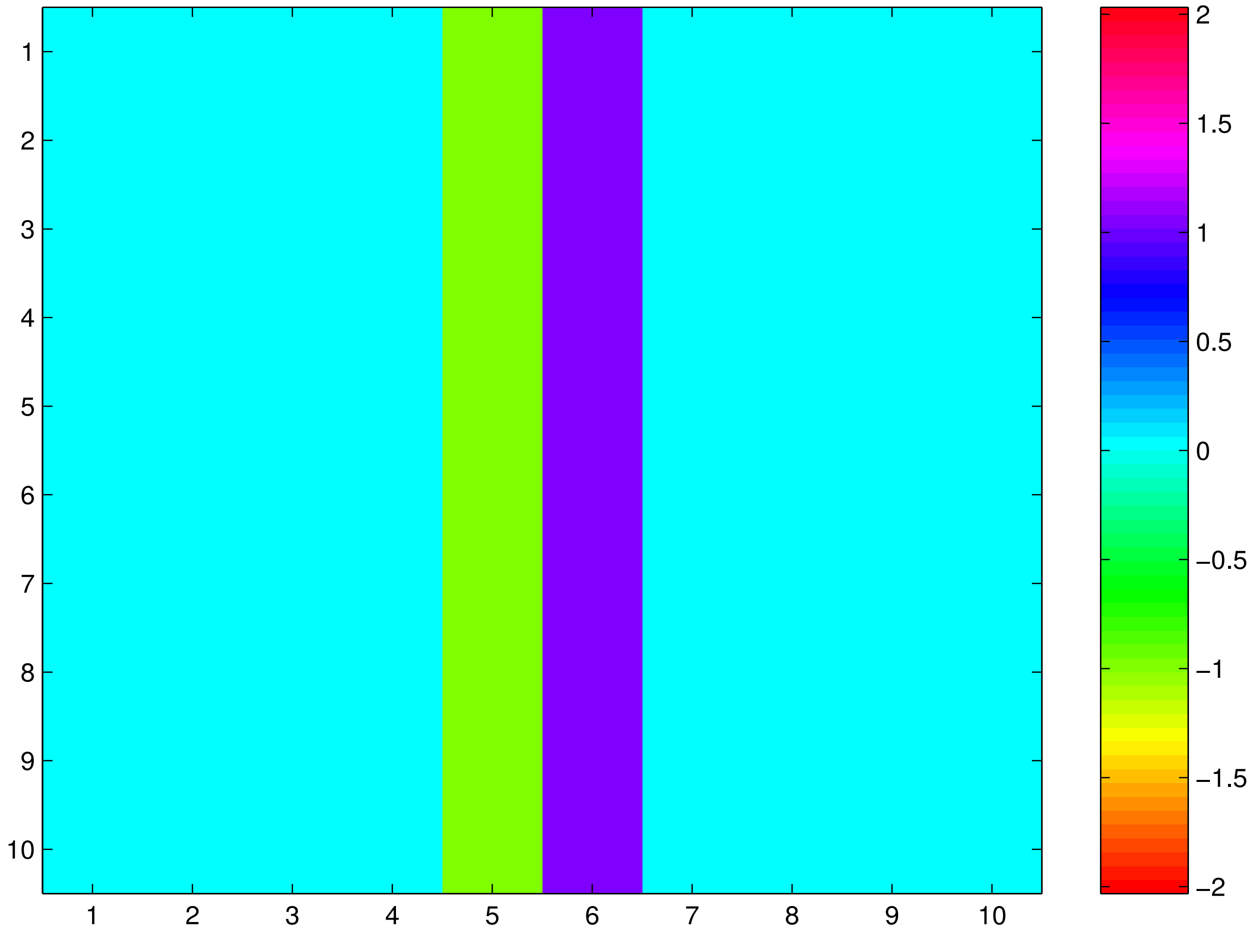

Figure 17: A "striper" in the null space of M; the pseudoinverse estimate of this slowness perturbation is essentially zero

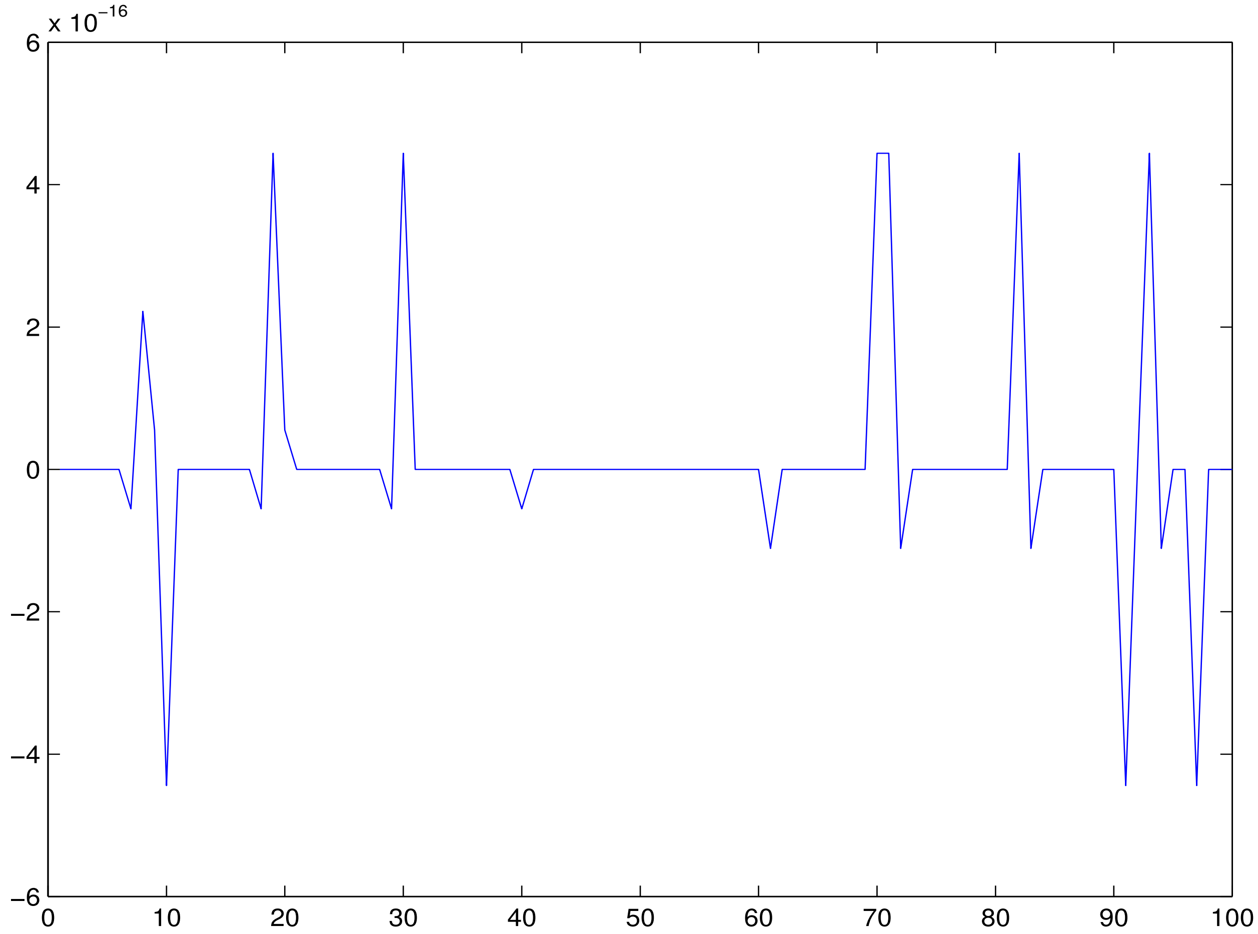

Figure 18: Data generated by the “striper”

But what does it all mean...

All of this discrete stuff is supposed to approximate the continuum.

In order to have *convergent* singular spectrum as grid is refine, must use *stronger norm* in domain. Expl: H^1 norm

$$\|u\|_1^2 = u^T B u$$

where B is discrete *Laplacian*. Relevant singular spectrum: solutions of

$$M^T M v = \lambda B v$$

Iterative Least Squares

Tomography problem may involve $n > 10^5$ and $m > 10^6$, so inconvenient for dense LA tools like QR-SVD. Also M is *very* sparse $\Rightarrow u \mapsto Mu$ is very cheap $\Rightarrow$ either sparse linear algebra or iterative solution.

Iterations:

- stochastic methods - Monte Carlo, simulated annealing, genetic algorithms,...
- row action methods - Kacmarz = ART, SIRT,...
- Krylov space methods - CG, LSQR,...

These lectures: Krylov space methods

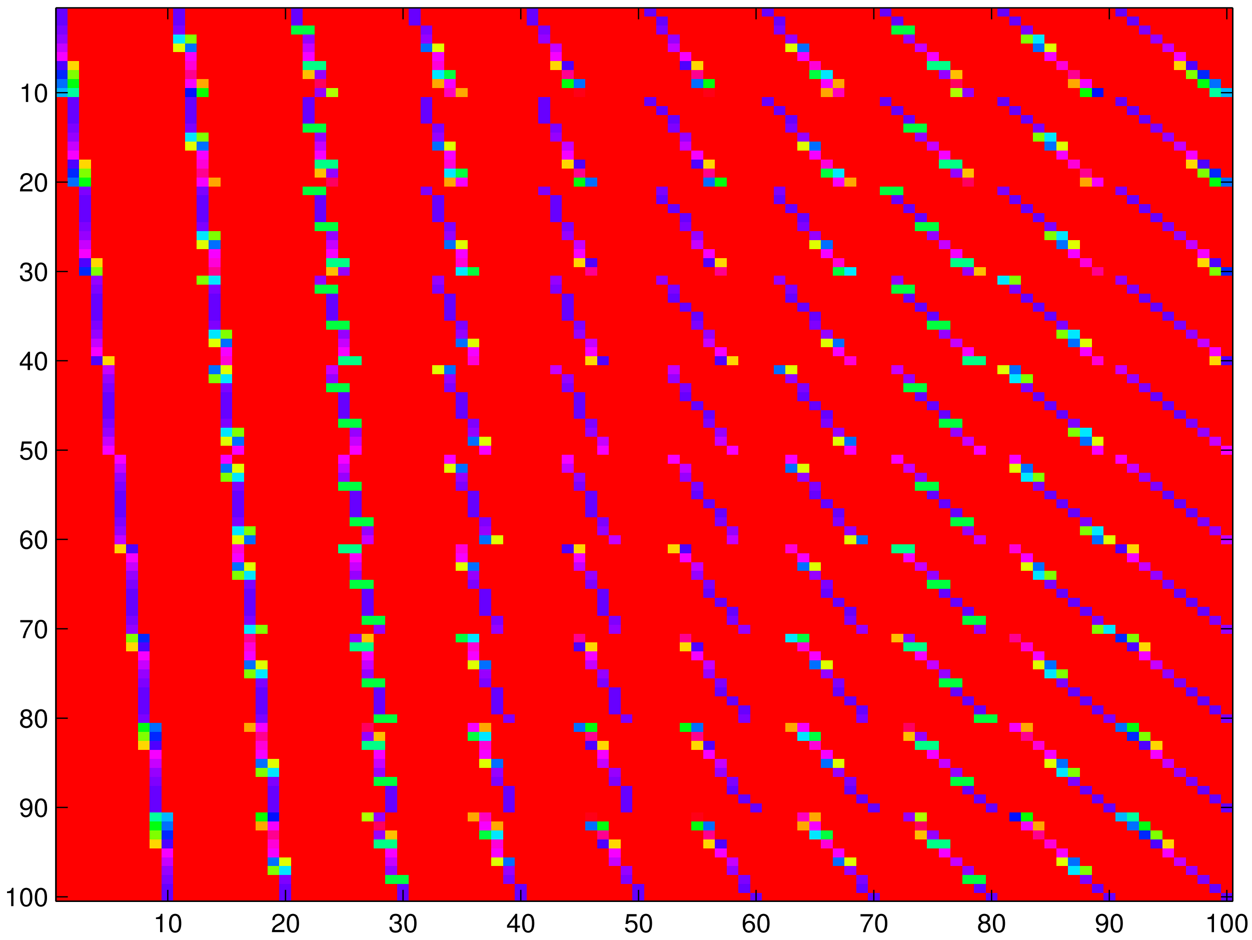

Figure 19: Sparsity pattern of M for 10×10 linear straight ray tomography example.

Krylov Space Methods

$A =$ symmetric positive definite $m \times m$ matrix, $s \in \mathbf{R}^m$

jth Krylov subspace
$\mathcal{K}_j(s, A) = \{a_0 s + a_1 A s + a_2 A^2 s + ... + a_j A^j s\}$

Krylov space methods for solving $Au = b$: given initial estimate u_0 ($\Rightarrow$ initial residual $s_0 = b - Au_0$): find best approximation $u_j \in u_0 + \mathcal{K}_j(s_0, A)$ to solution $u_* = A^{-1} b$ in some norm.

This would be uninteresting, were there not very inexpensive recursions which compute u_j!

CG

Conjugate Gradient approximation: choose $u_j \in u_0 + \mathcal{K}_j(s_0, A)$ to minimize "A norm":

$$u_j = \text{argmin}_{u_0+\mathcal{K}_j(s_0,A)} (u_* - u_j)^T A(u_* - u_j)$$

$$= s_j^T A^{-1} s_j, \;\; s_j = b - Au_j$$

Positive definite quadratic optimization $\Rightarrow$ unique minimizer

$$u_j = u_0 + (a_j^0 I + a_j^1 A + ... + a_j^j A^j) s_0$$

$$= u_0 + P_j(A) s_0$$

CGLS

Specialization to Least Squares: $A = M^T M, b = M^T t$, so $Au = b$ is normal equation $\Rightarrow Mu_* - t = t^\perp \perp \mathrm{Range}(M)$ $\Rightarrow$

$$(u_* - u_j)^T A(u_* - u_j) = (Mu_* - Mu_j)^T (t - Mu_j)$$
$$= \|r_j\|^2 - \|t^\perp\|^2, \; r_j = t - Mu_j$$

That is, u_j *solves the LS problem restricted to the* j*th Krylov subspace*

NB: Officially applies only to not-underdetermined problems, but null space components of u_j just sit still so cause no trouble (in exact arithmetic). Small singular values are a different story...

Spectral Analysis of Convergence

$$s_j = b - Au_j = b - Au_0 - P_j(A)s_0 = (I - AP(A))s_0$$

$\Rightarrow$ alternative char: s_j minimizes A^{-1} norm over $\{Q(A)s_0 : Q \text{ poly of deg } j+1, Q(0) = 1\}$ Introduce $\{v_k\}$ = eigenvectors of A = right singular vectors of M, eigenvalues $\{\lambda_k = \sigma_k^2\}$,

$$s_j = \sum_k s_j^k v_k = \sum_k (1 - \lambda_k P_j(\lambda_k)) s_0^k v_k$$

(sum over S. V. $\neq 0$), whence

$$s_j^T A^{-1} s_j = \sum_k \frac{(s_0^k)^2}{\lambda_k}(1 - \lambda_k P_j(\lambda_k))^2$$

Spectral Analysis of Convergence

Since $(s_0^k)^2/\lambda_k = (r_0^k)^2$, chooses P_j to min.

$$\|r_j\|^2 = \|t^\perp\|^2 + \sum_k |r_0^k|^2 (1 - \lambda_k P_j(\lambda_k))^2$$

Consequences:

- residuals decrease: $\|r_{j+1}\| \leq \|r_j\|$
- famous but generally useless property: since there are at most m distinct eigenvalues of A, $s_{m-1} = 0$ - "convergence in at most m steps"
- much more important property: if most of the "λ^{-1}-weighted energy" in s_0, is clustered on a set Λ, then need only make $(1 - \lambda P(\lambda))$ small on Λ.

Spectral Analysis of Convergence

- extreme example: if all eigenvalues are near λ_*, then get very good approx in *1 iteration!*
- more generally, if eigenvalues cluster near J limit points, then get very good approximation in J iterations
- hint at bad consequences of *small eigenvalues with large residual components* $(r_0^k)^2$: since $Q(0) = 1$, must use high degree to fit $\Rightarrow$ many iterations.

Spectral Analysis of Convergence

- another famous but useless estimate:

$$\|r_j\| = \|t^\perp\| + O\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)$$

  where κ = condition number of $A = \lambda_{\max}/\lambda_{\min}$ (over nonzero eigenvalues) = extremely large

- however can rescue to some extent: if $\sum_{\lambda_k \leq \lambda_{\mathrm{effmin}}} |r_0^k|^2$ is small, iteration behaves as if $\kappa = \lambda_{\max}/\lambda_{\mathrm{effmin}}$ at least until $\|r_j\|^2 - \|t^\perp\|^2$ roughly that small.

- residual components $\sim$ large S. V.s reduced *first*

Iteration revealed

The famous iterative construction of u_j:

Initialize $s_0 = b - Ax_0, p_0 = s_0$; do until convergence

$$\alpha_j = s_j^T s_j / p_j^T A p_j$$

$$u_{j+1} = u_j + \alpha_j p_j$$

$$s_{j+1} = s_j - \alpha_j A p_j$$

$$\beta_j = s_{j+1}^T s_{j+1} / s_j^T s_j$$

$$p_{j+1} = s_{j+1} + \beta_j p_j$$

Implementation for LS: `cgls.m`

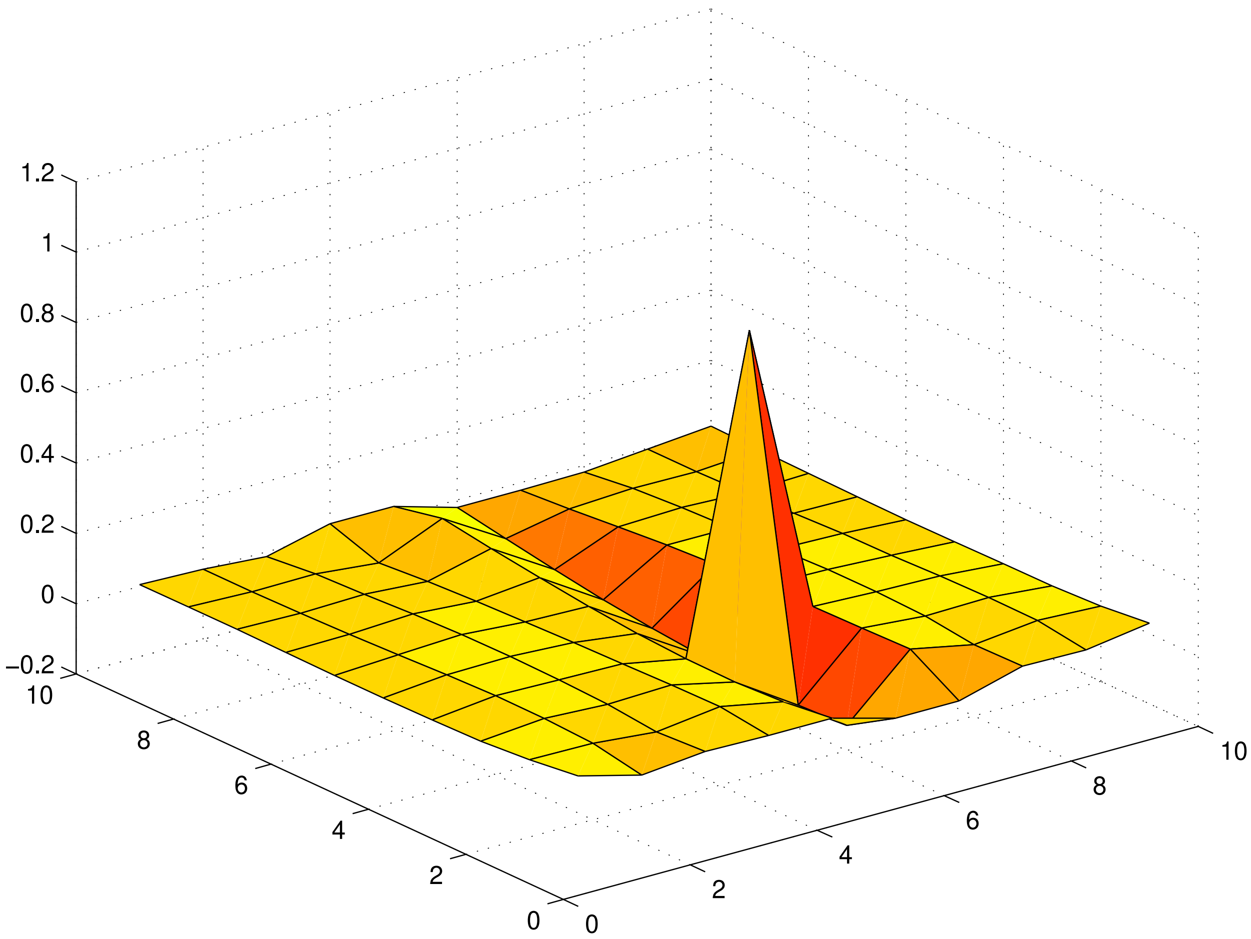

Figure 20: Estimate resulting from 50 conjugate gradient iterations

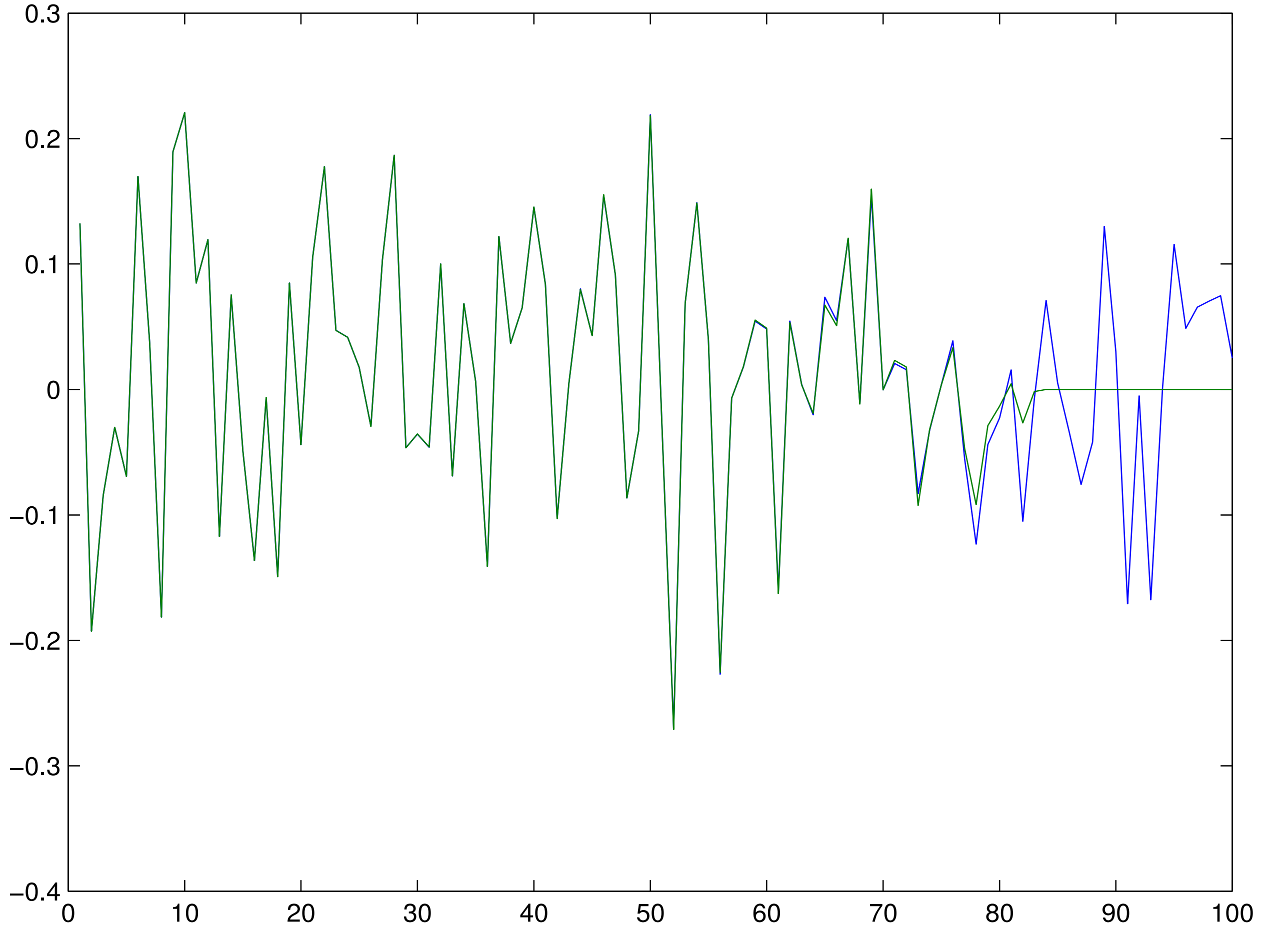

Figure 21: Coefficients of target (blue) and CG estimate, 50 iterations (green) slowness fields in singular vector expansion

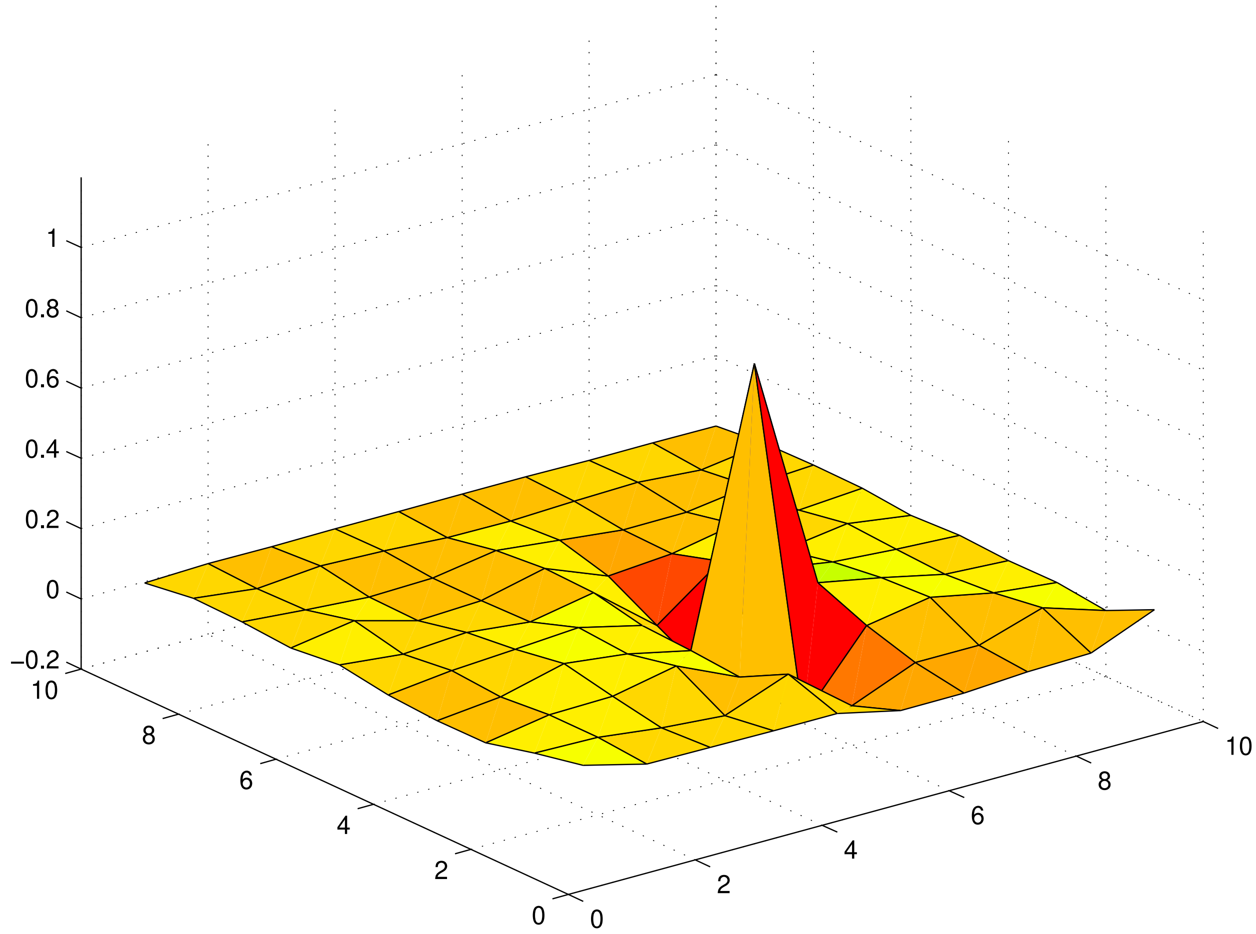

Figure 22: Estimate resulting from 50 conjugate gradient iterations

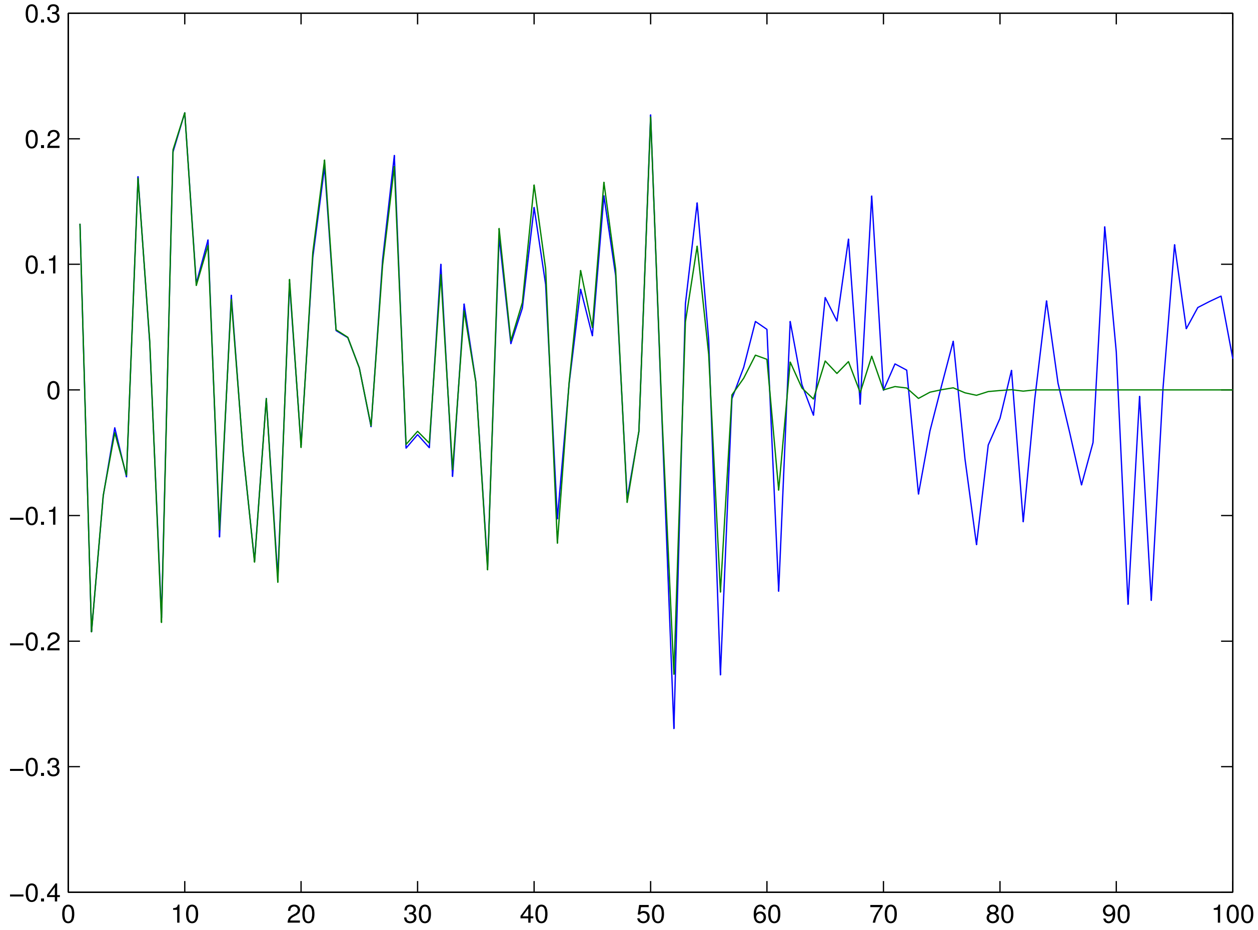

Figure 23: Coefficients of target (blue) and CG estimate, 10 iterations (green) slowness fields in singular vector expansion

Regularization

Essential matter, *aprés* Jackson: data are (almost always)

- *inconsistent*: no model predicts them precisely
- *inexact and insufficient*: not to be taken that seriously anyway ("noise"), and in any case grossly unable to pin down the model

An Applied Example: coastwise piloting

Regularization

= a means of choosing a representative model from amongst those that fit the data within experimental tolerance

- Jackson's "recalcitrant viewpoint": extremize something subject to fitting the data withing a (deterministic) error $\Rightarrow$ Tikhonov regularization, damped Least Squares, Levenburg-Marquardt,...*must know something about noise*

Regularization

- statistical cross-validation: randomly withhold data subsets, estimate model, predict withheld data, adjust constraints until prediction is as good as fit. Involves sampling a density,...*must know something about noise*
- families of data, rates of noise decay in various norms lead to optimal choice of damping parameters,... *must know something about noise*

In all cases: add information to remove ambiguity

Nonlinear Tomography

Generalities

Most of literature $\rightarrow$ local solutions to nonlinear least squares problems

$$\min_{u \in \mathcal{U}} J[u;t], \;\; J[u;t] \equiv \frac{1}{2}\|T[u]-t\|^2, \;\; T[u] \equiv M[u]u$$

Appeal to (?) smoothness, use some variant of Newton's method. Basic computations: gradient, Hessian

$$g[u;t] \equiv \nabla_u J[u;t] = DT[u]^T(T[u]-t)$$

$$H[u;t] \equiv \nabla\nabla J[u;t] = DT[u]^T DT[u] + D^2T[u](T[u]-t,\cdot)$$

Newton

Newton search direction p satisfies:

$$H[u;t]p = -g[u;t]$$

No reason *a priori* for this equation to have a solution.

Gauss-Newton approximation: drop second term in Hessian expression,

$$H_{\mathrm{GN}}[u;t]p_{\mathrm{GN}} = DT[u]^T DT[u]p_{\mathrm{GN}} = -g[u;t]$$

Always solvable, guaranteed to make progress:

$$J[u + \alpha p_{\mathrm{GN}}[u;t], t] \leq J[u;t]$$

for α sufficiently small, unless u is a stationary point.

Why tomographers love Gauss-Newton

Recall (!) Fermat $\Rightarrow DT[u] = M[u]$, so G-N condition is

$$M[u]^T M[u] p = M[u] r, \; r = t - T[u]$$

i.e. Linear Tomography all over again!

Generic Algorithm (see n papers, $n >> 1$): initialize u, r, then repeat unto exhaustion

- choose α
- update $u \leftarrow u + \alpha p$
- compute $r, M[u]$

One from column A...

- step (α) selection: simple damping, sophisticated linesearch with backtracking, trust region
- computation: $r, M[u]$
- other possibilities for H: generalized secant ("BFGS", NL2SOL,...)
- other gradient-based iterations: nonlinear CG
- something else altogether: simulated annealing, genetic algorithms,...
- constraints: must impose somehow