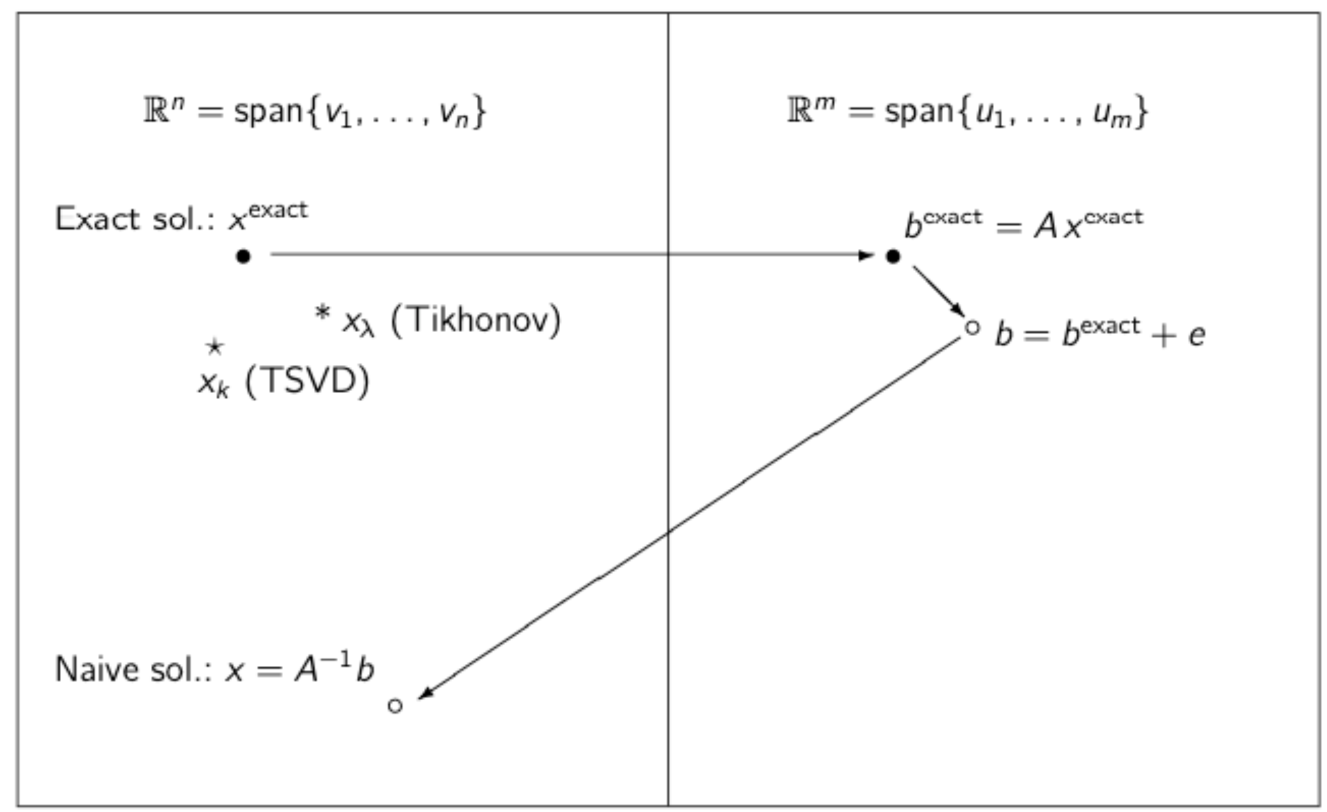
Lecture 4

From theory to practice

Topics in Inverse Problems
Fall 2021

Last time

1. Picard's theorem - solvability of $Ax = y$.
   - Least squares solution: $x = A^\dagger y$ (pseudo-inverse) is **unstable**
2. Regularisation: restoring stability in practice
3. Convergence and optimality



Plan for today

- Discretization: constructing finite $Ax = y$
- Tikhonov regularisation
- Choosing the regularisation parameter
- Iterative reconstruction schemes

Literature

1. J. L. Mueller and S. Siltanen, *Linear and Nonlinear Inverse Problems with Practical Applications*. SIAM, 2012, **chapters 1-5**
2. Per Christian Hansen, *Discrete Inverse Problems: Insight and Algorithms*, SIAM, 2010
3. A. Kirsch, *An introduction to the mathematical theory of inverse problems*, vol. 120. Springer Science & Business Media, 2011, **chapters 1-3**
4. R. Kress, V. Maz'ya, and V. Kozlov, *Linear integral equations*, vol. 82. Springer, 1989, **chapters 15-17**
5. Heinz Werner Engl, Martin Hanke, A. Neubauer, *Regularization of Inverse Problems,* 1996

1,2: mostly discrete setting, easier to read
3,4,5: heavier math, rigorous proofs
4: focused on integral equations

Discretisation

$$Af = g$$

- $Af = \int K(x, y)f(y)dy$ **(1st kind)**
- $(A^*A + \alpha^2 I)f = A^*g$ **(2nd kind)**

"Direct" approximation

Continuous $$g(t) = (Af)(t) = \int_a^b K(t,s)f(s)ds, \quad t \in [c,d]$$

Semi-discrete $$g(t_i) = (Af)(t_i) = \int_a^b K(t_i,s)f(s)ds, \quad t_i \in [c,d]$$

- **Choose** $s_1, \ldots, s_n \in [a,b]$ **and** $t_1, \ldots, t_n \in [c,d]$
- **Look for an approximate solution** $\tilde{f}_j \approx f(s_j)$
- **Use a quadrature rule** $\int_a^b \varphi(x)dx = \sum_{j=1}^n \omega_j \varphi(s_j) + E_n$

Simple example: midpoint quadrature rule

$$s_j = \frac{j - \frac{1}{2}}{n}, \quad \omega_j = \frac{1}{n}, \quad j = 1,2,\ldots,n\,.$$

- **Solve the linear system**

Ill-conditioned system

$$\sum_{j=1}^n \omega_j K\left(t_i, s_j\right)\tilde{f}_j = g\left(t_i\right), \quad i = 1,\ldots,n$$

A few words about quadrature

General idea: interpolate, then integrate the interpolant exactly

- <u>Equally spaced points</u>: Newton-Cotes (Trapezoidal, Simpson)
- <u>Gauss</u>: weights, endpoint singularities etc. Hard to compute but very accurate.
- <u>Clenshaw-Curtis</u>: Tchebyshev polynomials (can use FFT) - **Chebfun**
- …

Best choice: depends on properties of $K(t, s)f(s)$, integration domain

Projection methods

$$Af = g, \qquad A : X \to Y$$ (general operator equation)

Replace with $Q_n A f_n = Q_n g, \qquad Q_n A : X_n \to Y_n$

1. $X_n = \text{span}\{\hat{x}_1, \ldots, \hat{x}_n\} \subset X$, $Y_n = \text{span}\{\hat{y}_1, \ldots, \hat{y}_n\} \subset Y$ **(bases)**
2. $Q_n : Y \to Y_n$ **a projection operator (**$Q_n Q_n = Q_n$**).**
3. **Set** $Q_n g = \sum_{j=1}^{n} g_{j,n} \hat{y}_j$
4. **Set** $f_n = \sum_{j=1}^{n} \zeta_{j,n} \hat{x}_j$

$$Q_n A \hat{x}_j = \sum_{i=1}^{n} a_{i,j} \hat{y}_i$$

$$\sum_{j=1}^{n} a_{i,j} \zeta_{j,n} = g_{i,n}, \quad i = 1, \ldots, n$$

$$\tilde{A}_n \zeta_n = \mathbf{g}_n$$

Kirsch Ch. 3.1

Kress Ch. 13, 17.1

Special case 1: collocation

$$(Af)(t) = \int K(t,s)f(s)ds = g(t) \qquad A : X \to C[a,b]$$

- Force $(Af_n)(t_i) = g(t_i)$ for *collocation points* $\{t_1, \ldots, t_n\}$.
- Define $Y_n = \mathcal{S}_n(t_1, \ldots, t_n)$ to be the space of linear splines (can be any linear space unisolvent w.r.t. $\{t_i\}$)
- Let $Q_n : Y \to Y_n$ be the corresponding **interpolation operator** on $\{t_i\}$
- Then $Q_n A f_n = Q_n g$ is equivalent to $(Af_n)(t_i) = g(t_i)$, $i = 1, \ldots, n$.
- In this case, the linear system becomes

$$\sum_{j=1}^{n} a_{i,j}\zeta_{j,n} = g(t_i), \quad a_{i,j} = (A\hat{x}_j)(t_i) = \int K(t_i, s)\hat{x}_j(s)ds$$

Special case 2: Galerkin method

- Key idea: enforce orthogonality of the residual to Y_n
- $Q_n A f_n = Q_n g$ and $Q_n : Y \to Y_n$ is an orthogonal projection
- $\left\langle A f_n - g, \hat{y}_i \right\rangle = 0$ for all $i = 1, \ldots, n$. ***(weak formulation)***

$$\sum_{j=1}^{n} a_{i,j} \zeta_{j,n} = \langle \hat{y}_i, g \rangle, \quad a_{i,j} = \langle \hat{y}_i, A \hat{x}_j \rangle = \iint K(s,t) \hat{y}_i(t) \hat{x}_j(s) ds dt$$

Note: collocation is <u>computationally</u> equivalent to Galerkin with $\hat{y}_j = \delta(t - t_j)$

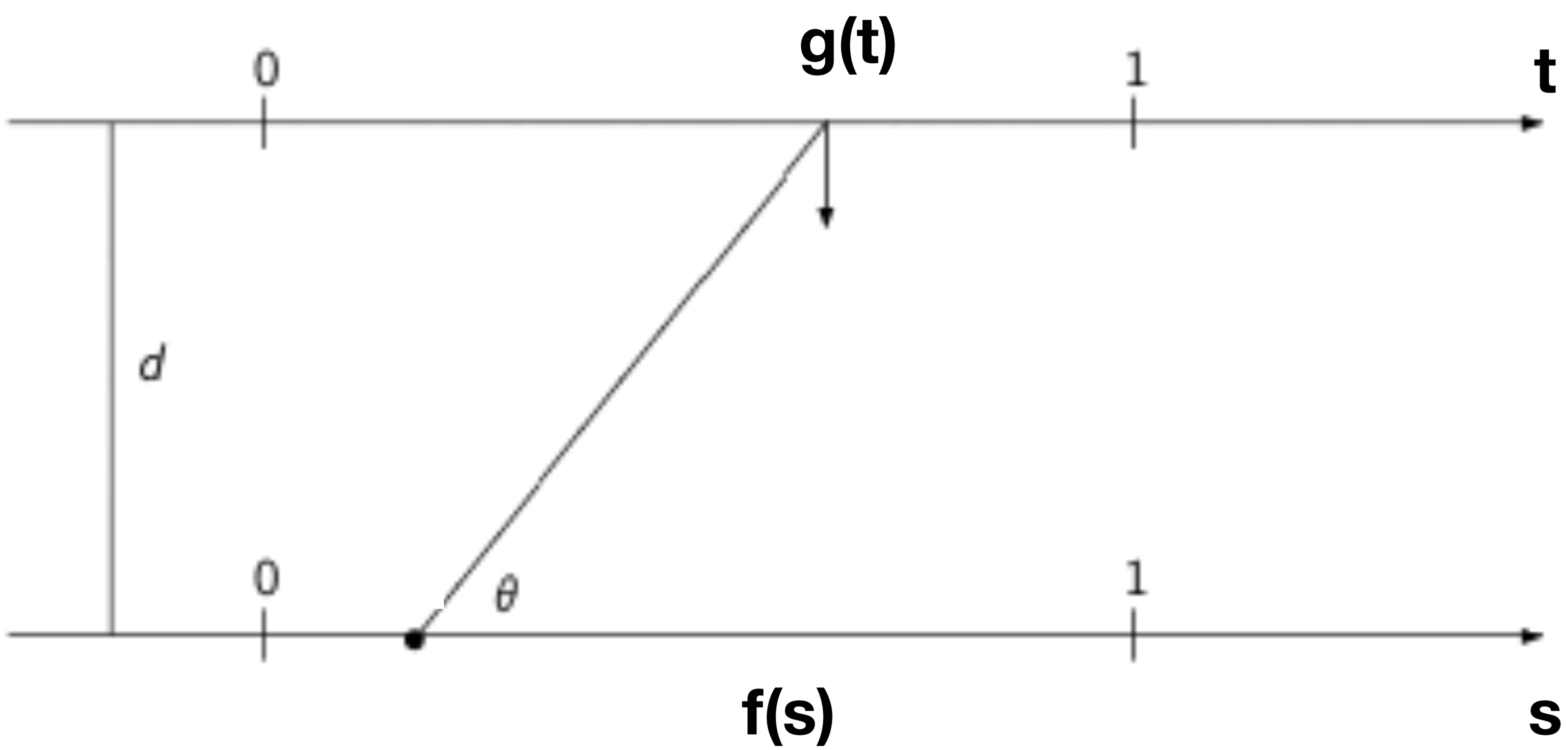
How to choose?

	Quadrature	Collocation	Galerkin
	$\sum_{j=1}^{n} \omega_j K\left(t_i, s_j\right) \tilde{f}_j = g\left(t_i\right)$	$a_{i,j} = (A\hat{x}_j)(t_i) = \int K(t_i, s)\hat{x}_j(s)ds$ $g_{i,n} = g(t_i)$	$a_{i,j} = \langle \hat{y}_i, A\hat{x}_j \rangle = \iint K(s,t)\hat{y}_i(t)\hat{x}_j(s)dsdt$ $g_{i,n} = \langle \hat{y}_i, g \rangle = \int g(t)\hat{y}_j(t)dt$
For			
Against			

How to choose?

	Quadrature	Collocation	Galerkin
	$\sum_{j=1}^{n} \omega_j K\left(t_i, s_j\right) \tilde{f}_j = g\left(t_i\right)$	$a_{i,j} = (A\hat{x}_j)(t_i) = \int K(t_i, s)\hat{x}_j(s)ds$ $g_{i,n} = g(t_i)$	$a_{i,j} = \langle \hat{y}_i, A\hat{x}_j \rangle = \iint K(s,t)\hat{y}_i(t)\hat{x}_j(s)dsdt$ $g_{i,n} = \langle \hat{y}_i, g \rangle = \int g(t)\hat{y}_j(t)dt$
For	++Implementation	+Implementation +Data availability	+Accuracy
Against	-Analysis (convergence etc.)	- Implementation	- - Implementation

Example



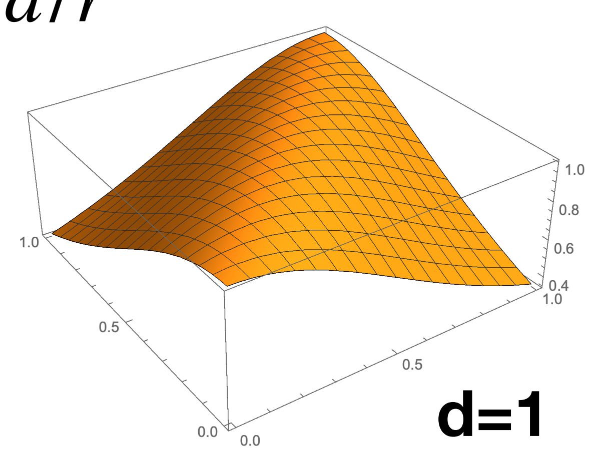
- $f(s)$ - unknown mass distribution
- $g(t)$ - measured gravitational force

$$dg = f(t)\sin(\theta)/r^2 dt$$

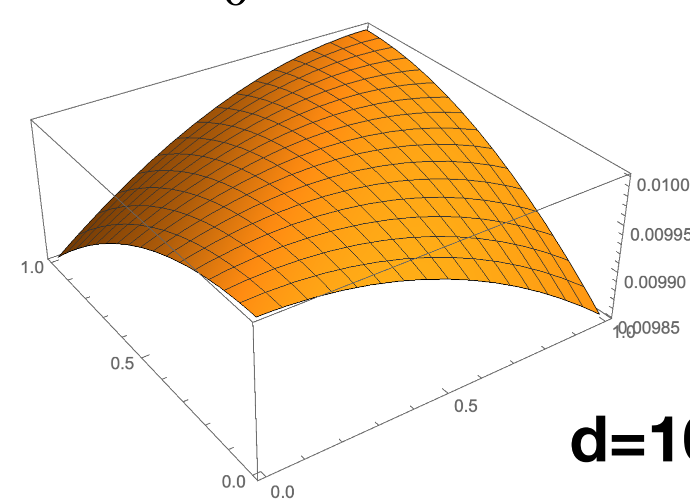
$$r^2 = d^2 + (s-t)^2$$

$$\sin(\theta) = d/r$$

$$\longrightarrow \quad g(t) = \int_0^1 \frac{d}{(d^2 + (t-s)^2)^{3/2}} f(s) \mathrm{d}s$$



d=1

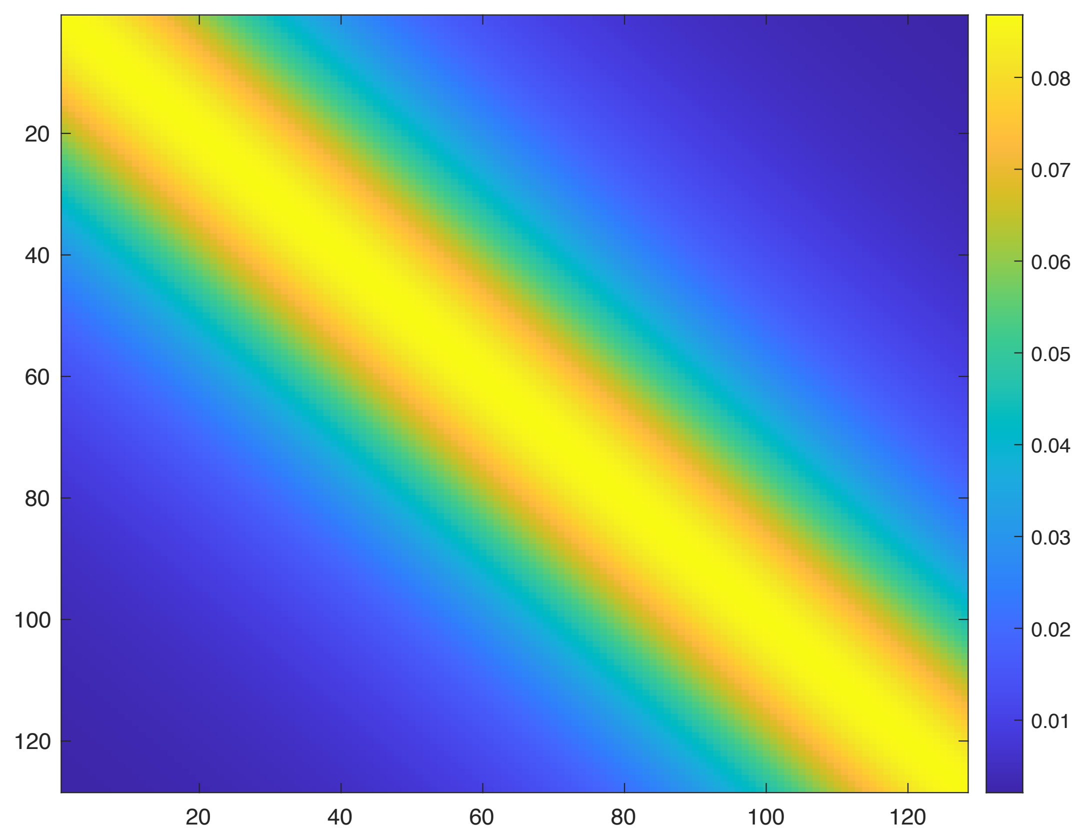


d=10

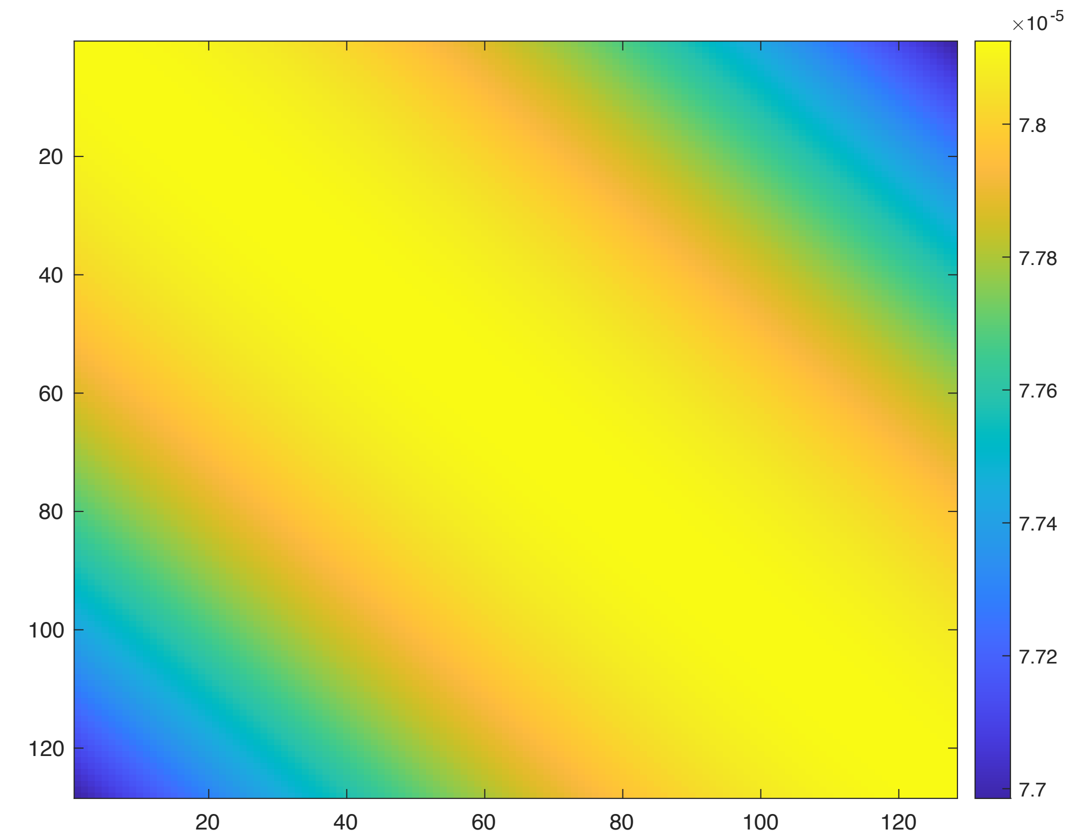
Quadrature

$$g(t) = \int_0^1 \frac{d}{(d^2 + (t-s)^2)^{3/2}} f(s)\mathrm{d}s$$

$$t_j = s_j = \frac{j - \frac{1}{2}}{n}, \quad \omega_j = \frac{1}{n}, \quad j = 1,2,\ldots,n. \longrightarrow \quad a_{i,j} = \frac{d}{n}\left(d^2 + \left(\frac{i-j}{n}\right)^2\right)^{-3/2}$$



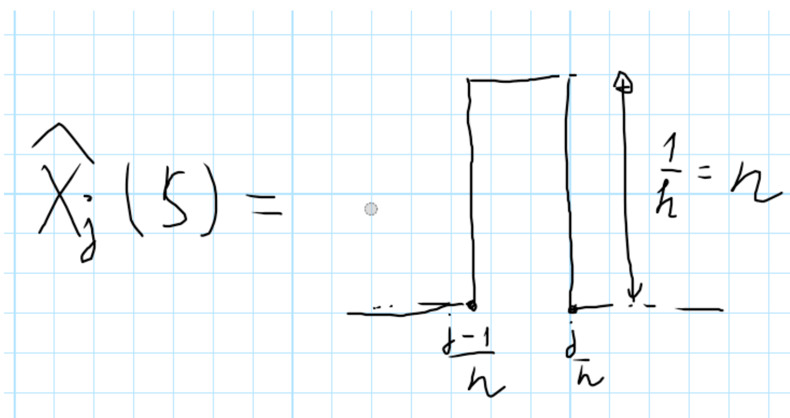
d=0.3



d=10

Galerkin/collocation

$$g(t) = \int_0^1 \frac{d}{(d^2 + (t-s)^2)^{3/2}} f(s)\mathrm{d}s$$



$$\hat{y}_i(t) = \delta(t - t_i), \quad t_i = \frac{i - 1/2}{n}$$

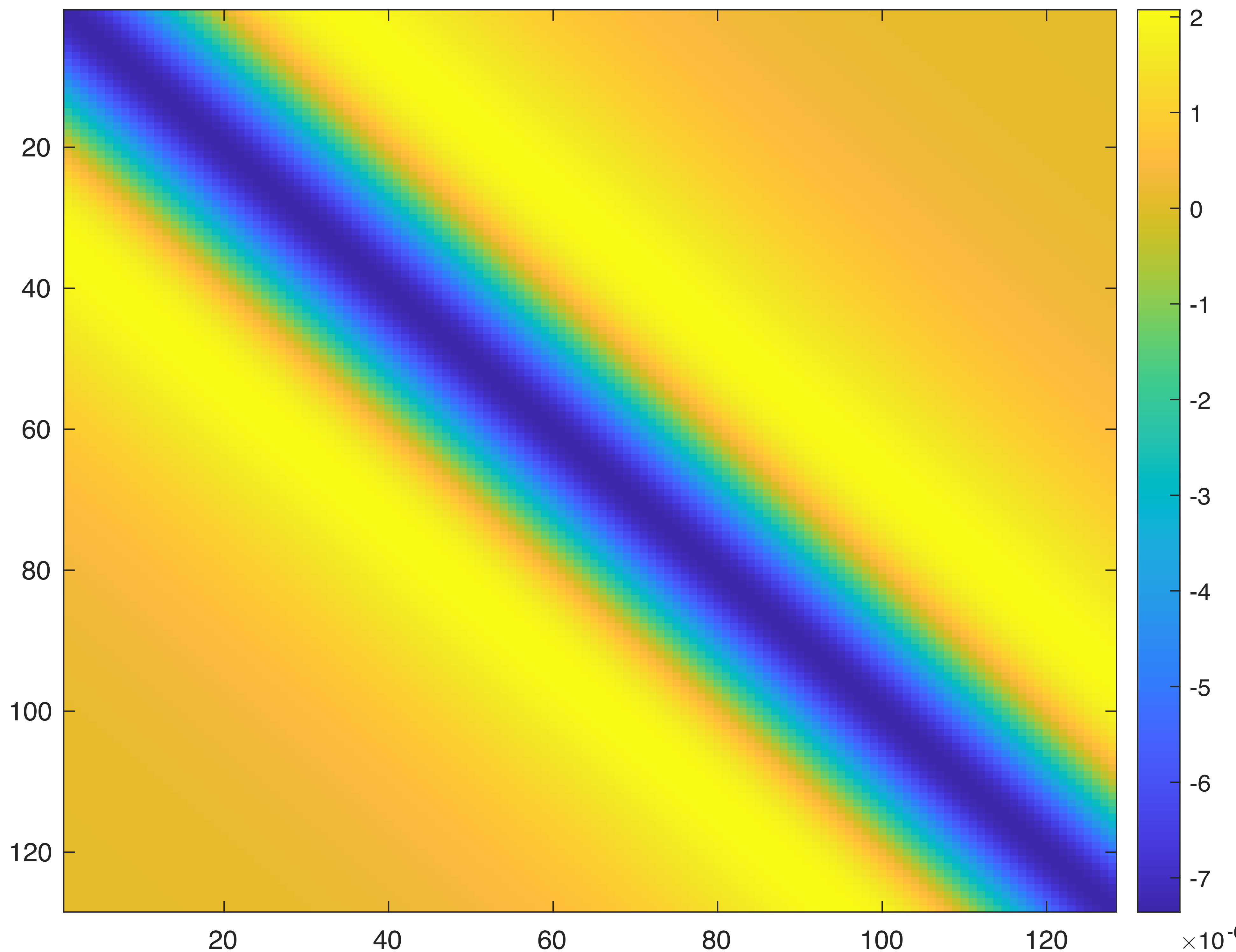
$$a_{i,j} = \int_0^1 K(t_i, s)\hat{x}_j(s)ds$$

$$= \int_{\frac{j-1}{n}}^{\frac{j}{n}} K(t_i, s)ds$$

$$\dots$$

$$= \frac{\frac{-2i+2j+1}{\sqrt{4d^2 + \frac{(-2i+2j+1)^2}{n^2}}} + \frac{2i-2j+1}{\sqrt{4d^2 + \frac{(2i-2j+1)^2}{n^2}}}}{dn}$$

Poll:
should the matrices be equal?



Difference matrix for d=0.3

Galerkin/collocation

$$g(t) = \int_0^1 \frac{d}{(d^2 + (t-s)^2)^{3/2}} f(s)\mathrm{d}s$$

$\hat{x}_i(t) = P_i(t)$ **(Legendre)**

$\hat{y}_i(t) = \delta(t - t_i), \quad t_i = \dfrac{i - 1/2}{n}$

$$a_{i,j} = \int_0^1 K(t_i, s)\hat{x}_j(s)ds = ?$$

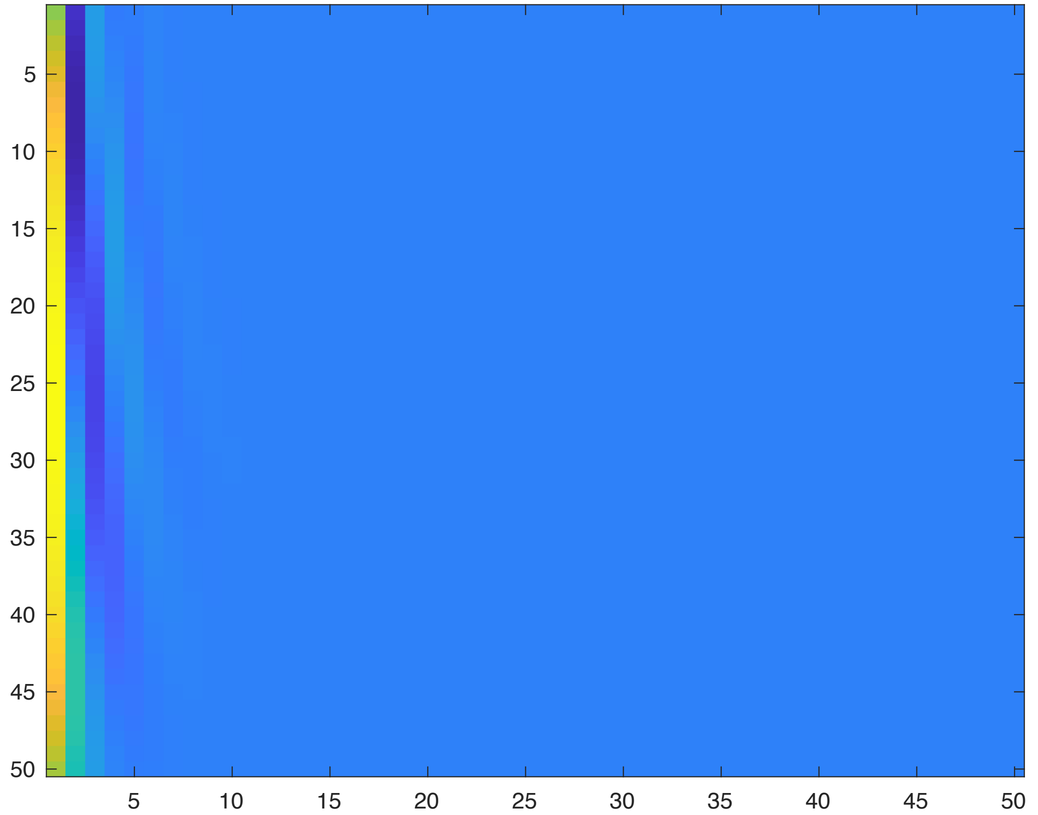
```
n = 50;
d = 0.3;

k = 1:n;

K = @(s,t) d./((d^2+(t-s).^2).^1.5);

A_leg = zeros(n,n);

for i=1:n
    fprintf('i=%d\n',i);
    ti = (i-1)/n;
    for j=0:n-1
        fprintf('\t j=%d\n',j+1);
        fj = @(x) K(ti,x).*legendreP(j,2*x-1);
        A_leg(i,j+1) = quadgk(fj,0,1);
    end
end
```

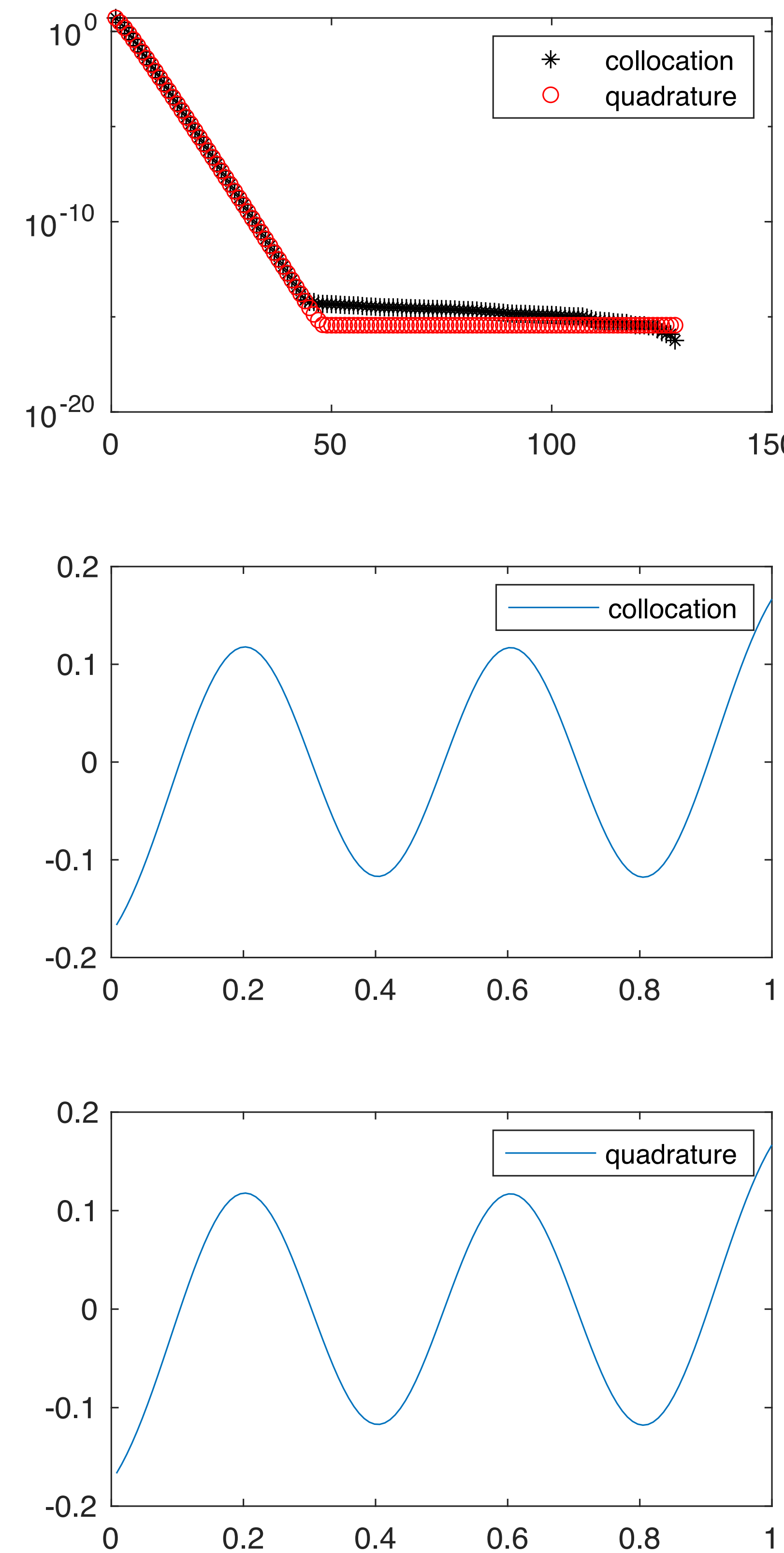


Approximating SVD

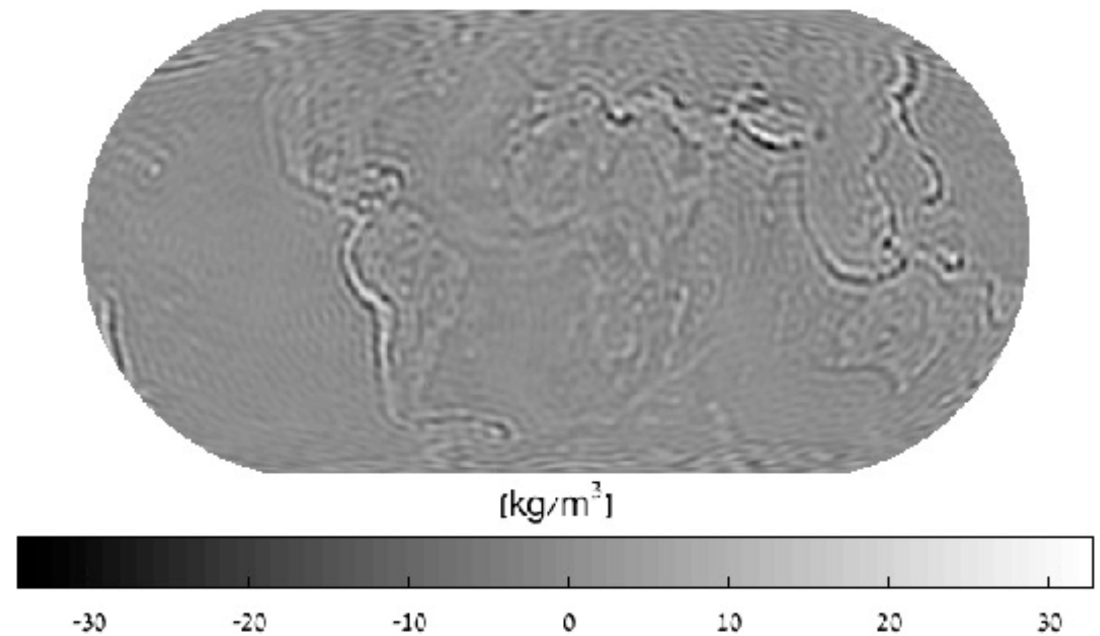
- For most operators A, the SVD cannot be computed analytically

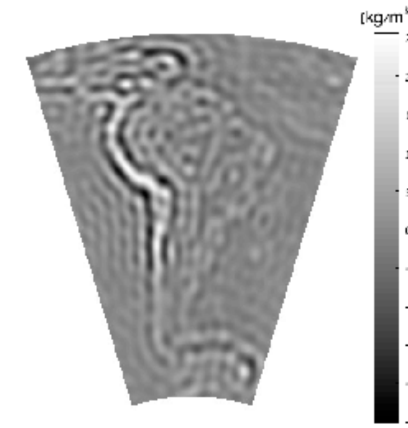
- We can use Galerkin with $n \to \infty$ and **orthonormal basis** $\{\hat{x}_j\}, \{\hat{y}_i\}$ to get accurate estimates

P. C. Hansen, “Computation of the singular value expansion,” *Computing*, vol. 40, no. 3, pp. 185–199, Sep. 1988, doi: 10/btjhsm.



More complicated example





Measurements: gravitational potential

$$(T\rho)(\vec{y}) = V(\vec{y}) = -G\int_{\oplus} \frac{\rho(\vec{x})}{|\vec{x}-\vec{y}|}\mathrm{d}^3x,$$

Basis for Y $\quad \mathcal{F}_k(\rho) = -G\int_B \frac{\rho(\vec{y})}{|\vec{y}-\vec{x}_k|}\mathrm{d}^3y, \quad \vec{x}_k \in \mathbb{R}^3\backslash\bar{B}. \qquad \hat{y}_j = \delta(t-t_j)$

Basis for X $\quad \mathcal{F}_k K_{\mathcal{H}_J}(\cdot,\vec{y}) \qquad K_{\mathcal{H}_J}(\vec{x},\vec{y}) = \sum_{n=0}^{\infty}\sum_{m=-n}^{n} \varphi_J^2(n) H_{n,m}^B(\vec{x}) H_{n,m}^B(\vec{y})$

Matrix elements

$$= \sum_{n=0}^{\infty} \varphi_J^2(n) \frac{4\pi\beta^3 G^2}{(2n+1)(2n+3)} \frac{1}{|\vec{x}_i||\vec{x}_j|}\left(\frac{\beta^2}{|\vec{x}_i||\vec{x}_j|}\right)^n P_n\left(\frac{\vec{x}_i\cdot\vec{x}_j}{|\vec{x}_i||\vec{x}_j|}\right).$$

- V. Michel and K. Wolf, Numerical aspects of a spline-based multiresolution recovery of the harmonic mass density out of gravity functionals, Geophysical Journal International 173, 1 (2008).
- M. Fengler, D. Michel, and V. Michel, Harmonic spline–wavelets on the 3–dimensional ball and their application to the reconstruction of the earths density distribution from gravitational data at arbitrarily shaped satellite orbits, ZAMM 86, 856 (2006).

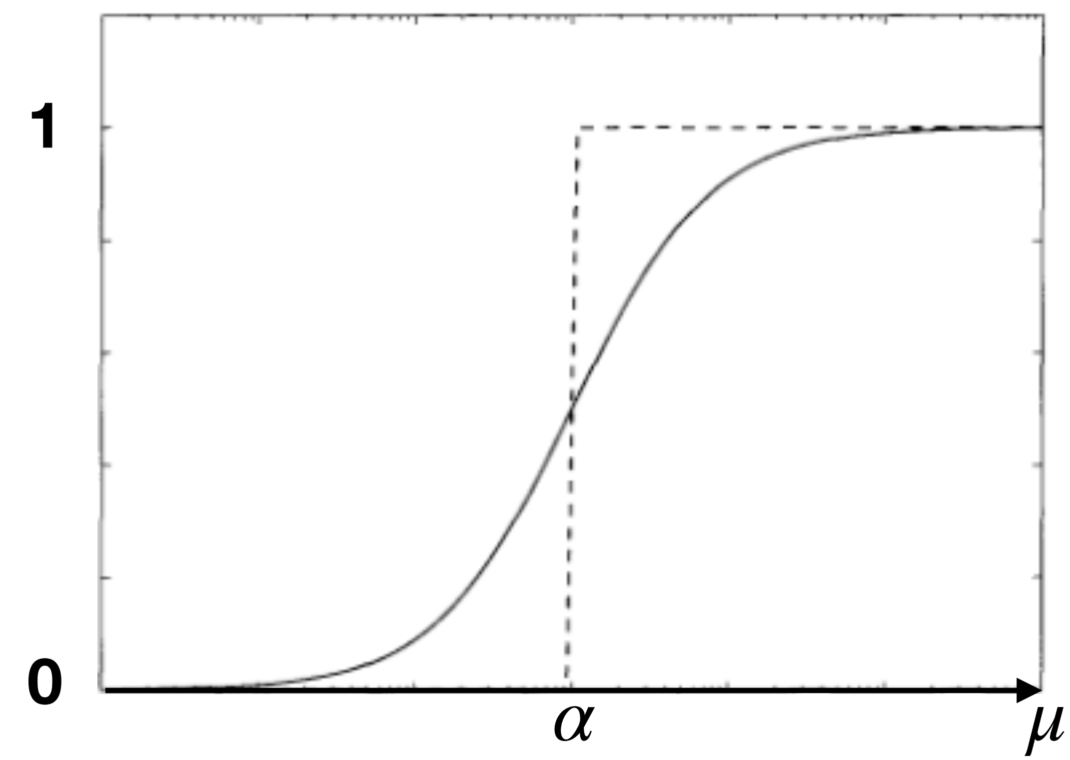
A project done in the course

Tikhonov regularisation

Three points of view

$$x^{\delta,\alpha} = \sum_{i=1}^{\infty} \frac{q\left(\alpha, \mu_i\right)}{\mu_i} \langle y^\delta, \psi_i \rangle \varphi_i =: R_\alpha y^\delta$$

$$q(\alpha, \mu) = \frac{\mu^2}{\mu^2 + \alpha^2}$$



Replace ill-posed problem $Ax^\delta = y^\delta$ **whose solution is** $x^\delta = (A^*A)^{-1}A^*y^\delta$

with $x^{\delta,\alpha} = (A^*A + \alpha^2 I)^{-1}A^*y^\delta$

$$x^{\delta,\alpha} = \arg\min_x \|Ax - y^\delta\|^2 + \alpha^2\|x\|^2$$

(penalize large x**)**

Convergence & optimality

$$x^{\delta,\alpha} = (A^*A + \alpha^2 I)^{-1} A^* y^\delta$$

Can be shown that:

Convergence: any $\alpha(\delta) \to 0$ **such that** $\dfrac{\delta}{\alpha(\delta)} \to 0$ **as** $\delta \to 0.$

For example, $\alpha(\delta) = \delta^p, \quad 0 < p < 1.$

Optimality under range conditions:

$\|(A^*)^{-1}x\| \leq E$ $\qquad$ $\alpha(\delta) = c\sqrt{\dfrac{\delta}{E}}$ $\qquad$ $\|\delta x\| \leq (c + 1/c)\sqrt{\delta E}\,.$

Tikhonov regularisation

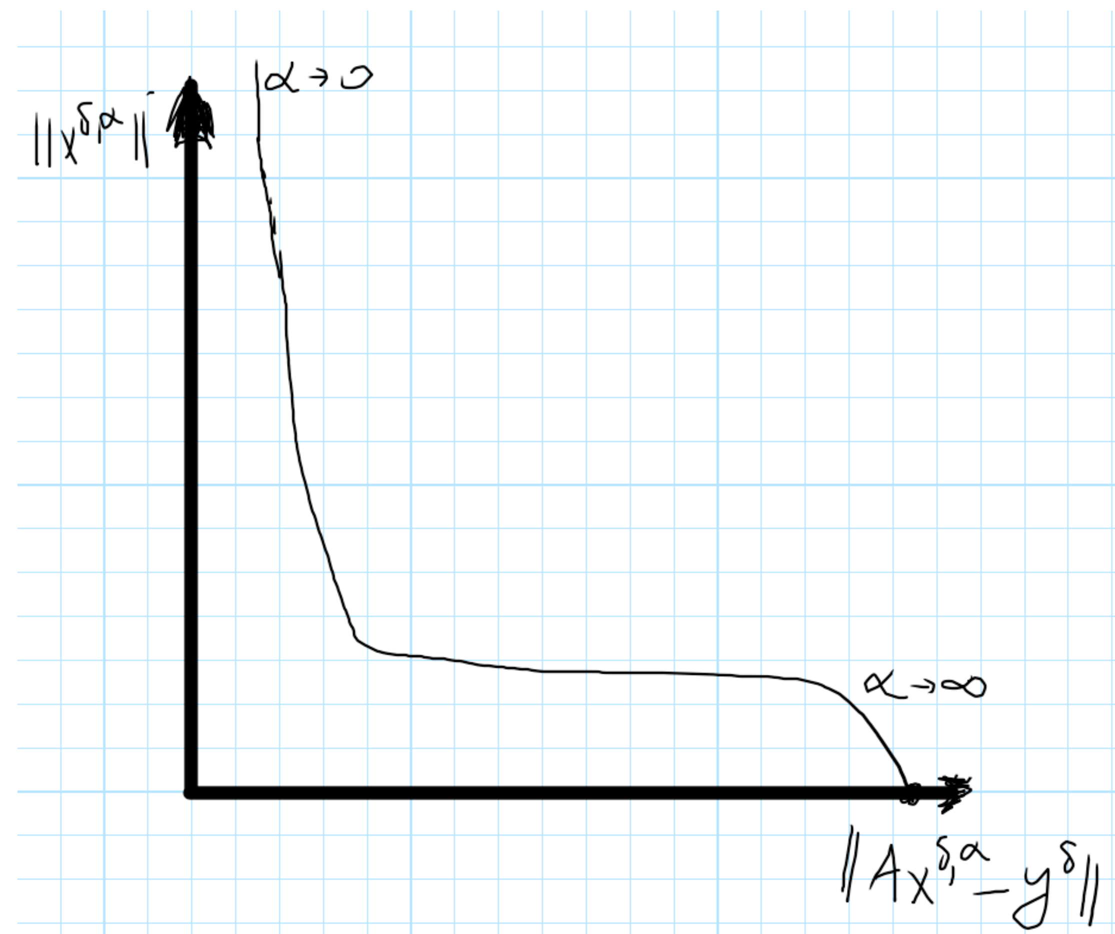
$$x^{\delta,\alpha} = \arg\min_x \|Ax - y^\delta\|^2 + \alpha^2\|x\|^2$$

- Normal equations: $(A^TA + \alpha^2 I)x^{\delta,\alpha} = A^T y^\delta$
- "Trick": write this as $\arg\min_x \left\| \begin{pmatrix} A \\ \alpha I \end{pmatrix} x - \begin{pmatrix} y^\delta \\ 0 \end{pmatrix} \right\|_2$
- We saw how to choose α optimally if $\|(A^*)^{-1}x\| \leq E$, but this is a-priori choice
- Want a-posteriori choices (decide based on the actual computed solution(s))

L-curve

$$x^{\delta,\alpha} = \arg\min_x \|Ax - y^\delta\|^2 + \alpha^2\|x\|^2$$

- **We don't know x^0, but can compute**
    1. **Residual norm: $\|Ax^{\delta,\alpha} - y^\delta\|^2$**
    2. **Current solution norm: $\|x^\delta\|^2$**
- **As α goes from 0 to ∞, can show (using SVD) that**
    3. **$\|x^\delta\|^2$ monotonically decreases from $\|A^\dagger y^\delta\|^2$ to 0**
    4. **$\|Ax^\delta - y^\delta\|^2$ monotonically increases from 0 to $\|y^\delta\|^2$**
- **L-curve: plot one against the other (in log-log scale)**
- **Typically, it has two distinct regions**
    - **An almost vertical, for small α**
    - **An almost horizontal, for large α**
- **A heuristic rule: choose the "corner". <u>Not always works!</u>**

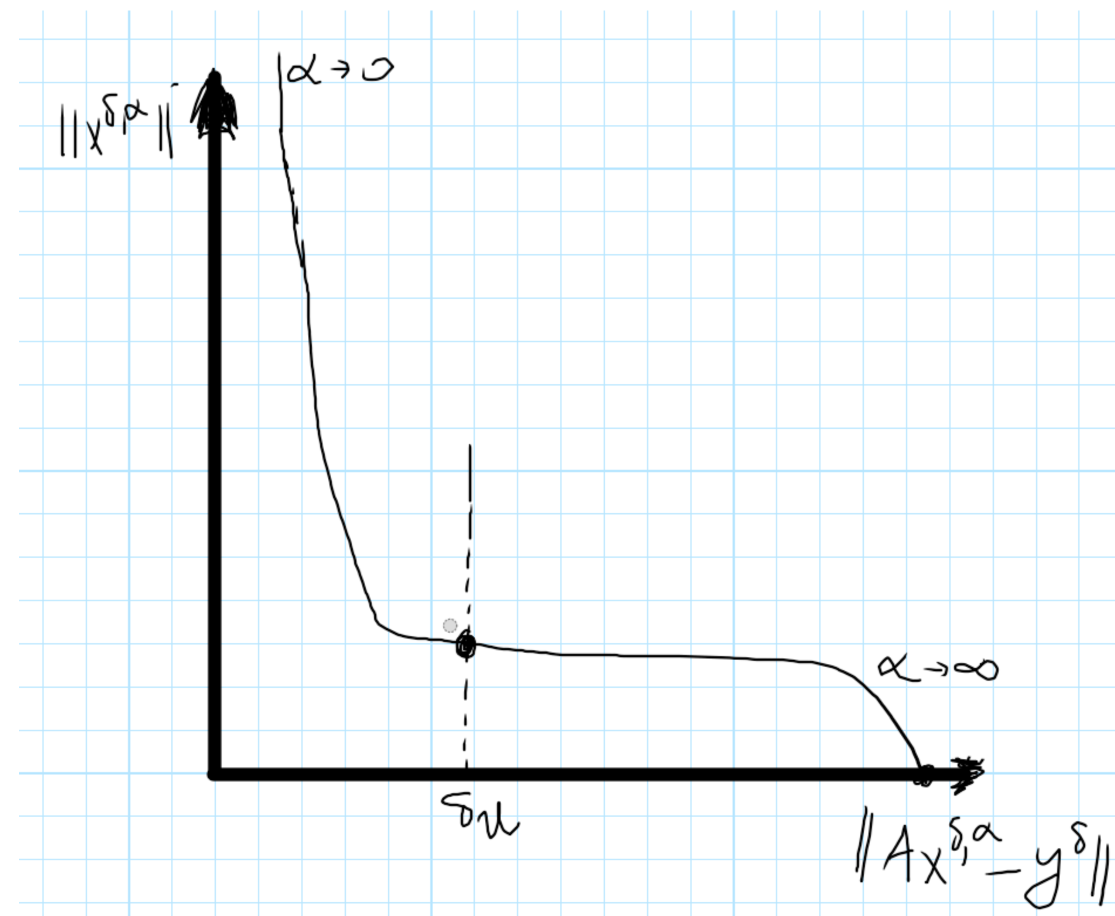


Discrepancy principle

- Idea: make sure that $\|Ax^{\delta,\alpha} - y^{\delta}\| = \delta_u$, where δ_u is an upper bound on the *discrepancy of the true solution.*

- Turns out that we can solve this nonlinear equation easily

- Can show convergence and optimality as $\delta \to 0$



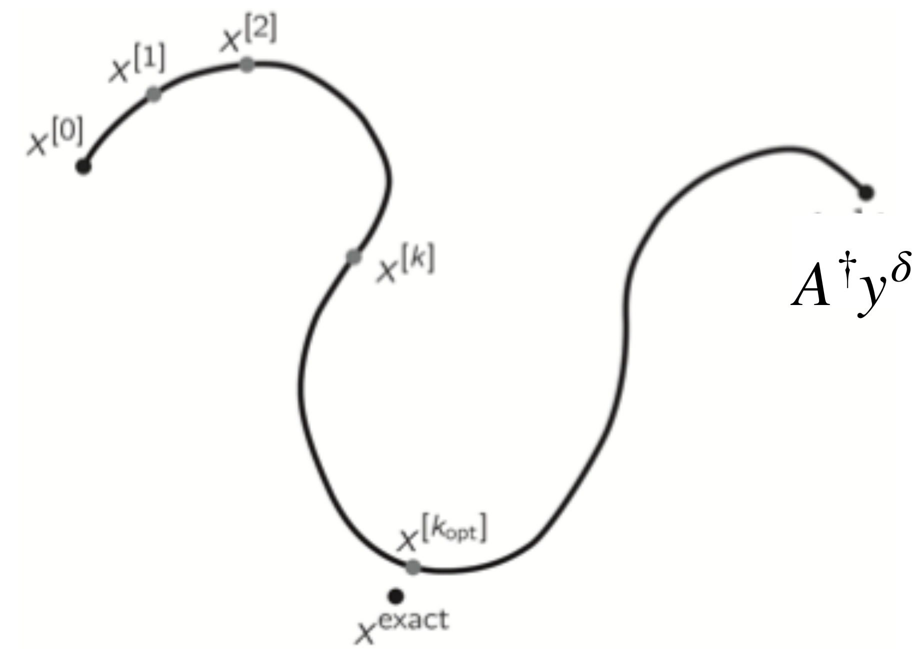
Iterative methods

Motivation

- For large-scale problems, cannot construct A and/or compute the SVD
- Do not want to solve for many α's
- Would like to use "black-box" functions computing multiplication by A or A^T

Iterative methods

$$Ax = y^{\delta}$$



- Start with some x^0 (usually $x^0 = 0$ is OK)
- Update $x^{k+1} = F(x^k)$ according to some rule
- In principle, algorithm should converge to $A^{\dagger}y^{\delta}$
- We stop at some $k = k_{opt}$ - this plays the role of reg. parameter

Gradient descent?

$$x^{k+1} = x^k - w\,\nabla f(x^k)$$, w**=step size**

- What is the gradient step for $f(x) = \frac{1}{2}\|Ax - y^\delta\|^2$?
- Using some matrix calculus, easy to show that
$$\nabla f(x) = A^T(Ax - y^\delta)$$
- **Landweber iteration:** $x^{k+1} = x^k + wA^T(y^\delta - Ax^k)$
- Need to choose $0 < w < \frac{2}{\|A^TA\|} = \frac{2}{\sigma_1^2}$ for convergence

Analysis

$$x^{k+1} = x^k + wA^T(y^\delta - Ax^k),\ x^0 = 0$$

1. **Show by induction that** $x^{k+1} = w\left(\sum_{j=0}^{k} S^j\right)A^T y^\delta$**, where**

$$Sx = (I - wA^TA)x = \sum_{i=0}^{\infty}(1 - w\mu_i^2)\left\langle x, \varphi_i\right\rangle \varphi_i$$

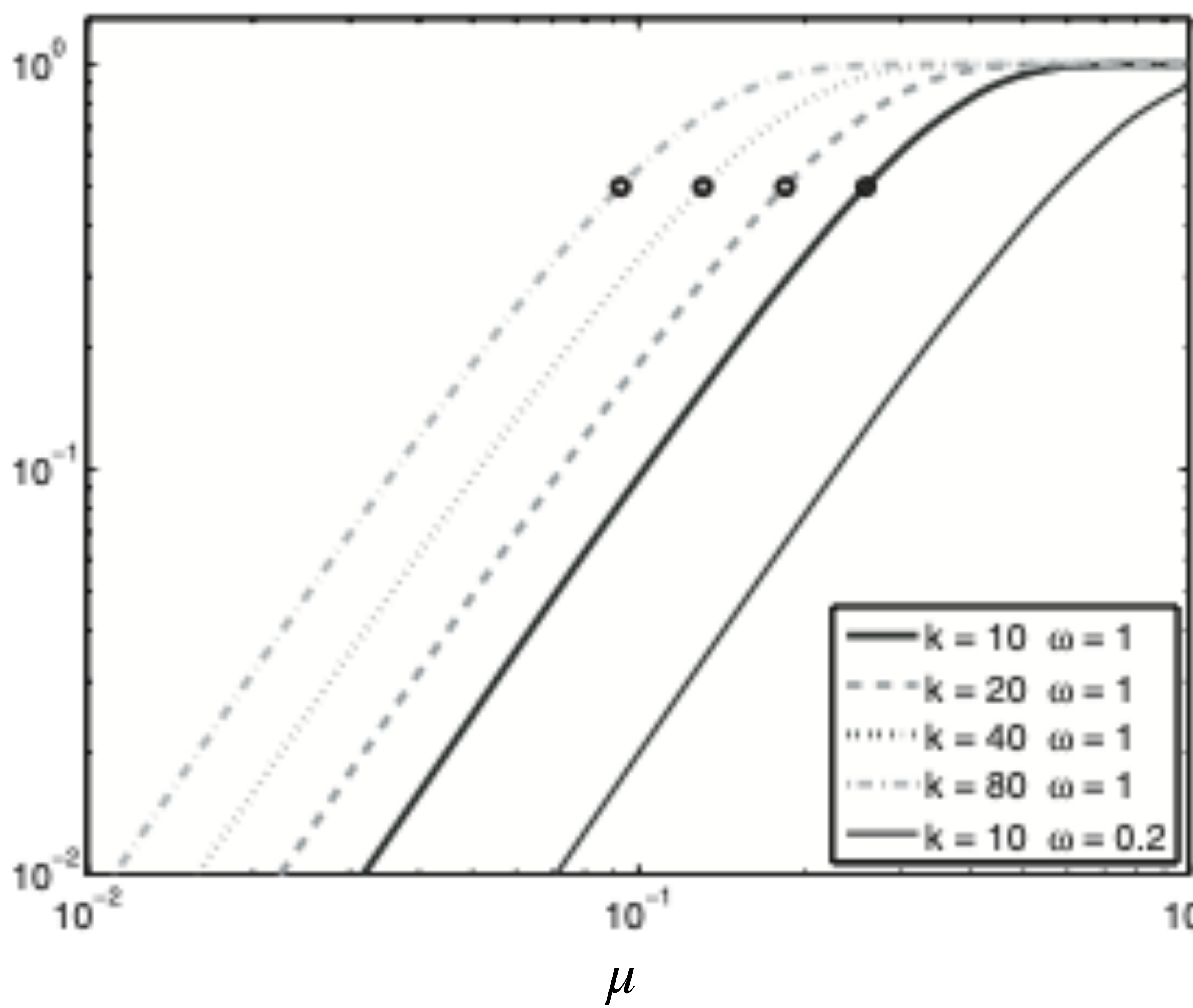
2. **Then, with a little more work, can show that**

$$x^{k+1} = \sum_{i=0}^{\infty}\frac{q_w(k,\mu_i)}{\mu_i}\left\langle y^\delta, \psi_i\right\rangle \varphi_i$$**, where** $q_w(k,\mu) = 1 - (1 - w\mu^2)^{k+1}$

3. **So this is a spectral filter, changing with each iteration**

Landweber filters

$$q_w(k, \mu) = 1 - (1 - w\mu^2)^{k+1}$$



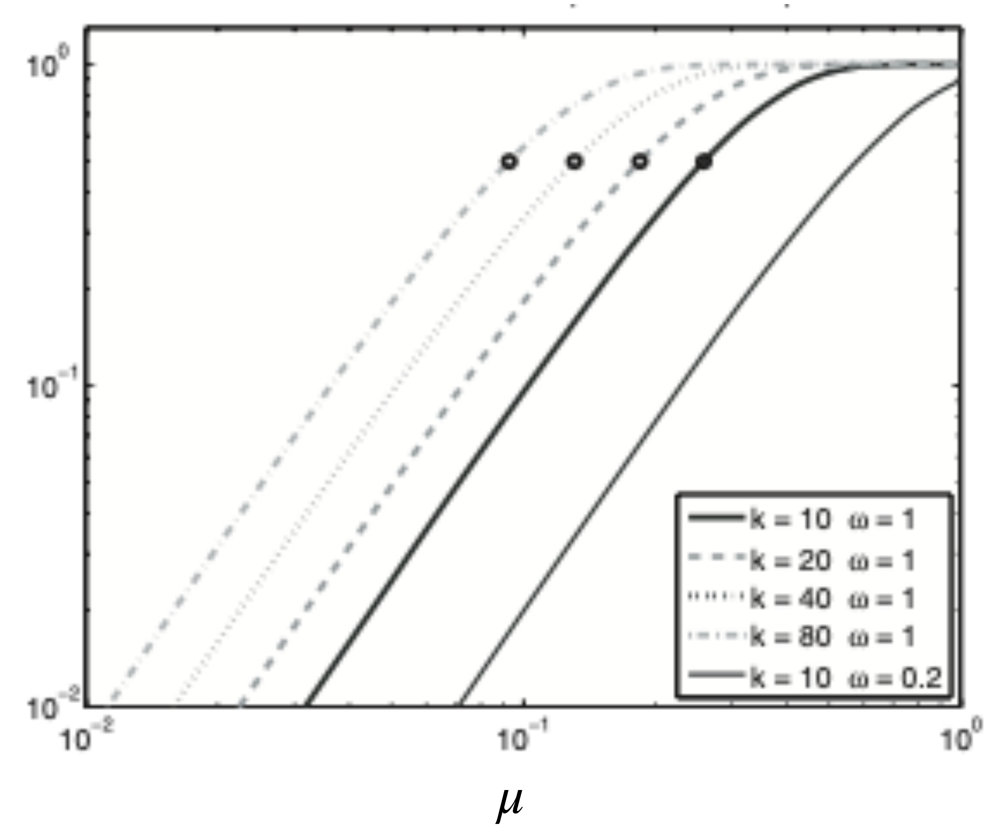
Landweber iteration: summary

$$x^{k+1} = x^k + wA^T(y^\delta - Ax^k),\ x^0 = 0$$

$$x^{k+1} = \sum_{i=0}^{\infty} \frac{q_w(k, \mu_i)}{\mu_i} \left\langle y, \psi_i \right\rangle \varphi_i$$

$$q_w(k, \mu) = 1 - (1 - w\mu^2)^{k+1}$$

- Slow convergence
- Optimal strategy with stopping rule (e.g. discrepancy principle)
- Can augment with a **projection** $x^{k+1} = \mathscr{P}\{x^k + wA^T(y^\delta - Ax^k)\}$



Conjugate gradients

- Descent direction at step k:
  $d_k = A^T r^k = A^T(y^\delta - Ax^k)$
- Problem: descent directions can repeat themselves
- Idea: choose d_k which is linearly independent from $\{d_1, \dots, d_{k-1}\}$
- Solution in at most n steps
- Tends to choose large singular components first, so can stop early