Iterative methods for Image Processing

Lothar Reichel

Como, May 2018.

Lecture 2: Tikhonov regularization and truncated SVD for large-scale problems.

Outline of Lecture 2:

- Small to moderately-sized problems
  - Tikhonov regularization in standard form
  - Tikhonov regularization in general form
  - The generalized SVD
- Large-scale problems
  - Tikhonov regularization based on Krylov subspace methods
  - Truncated SVD for large-scale problems

Tikhonov regularization

Solve the minimization problem

$$\min_x \{\|Ax - b\|_2^2 + \mu \|Lx\|_2^2\},$$

where $\mu > 0$ is a regularization parameter (to be determined) and $L \in \mathbf{R}^{p \times n}$ is a regularization matrix chosen so that

$$\mathcal{N}(A) \cap \mathcal{N}(L) = \{0\}.$$

Then the minimization problem has a unique solution for any $\mu > 0$.

Common choices of L: identity, discretizations of differential operator.

In our applications A is a smoothing operator. Therefore, the Tikhonov minimization problem generally has a unique solution when L is a discrete differential operator.

We would like L be such that important features of x_{exact} are not damped. This is the case when they are in $\mathcal{N}(L)$.

The normal equations associated with the Tikhonov minimization problem

$$(A^T A + \mu L^T L)x = A^T b$$

have the unique solution

$$x_\mu := (A^T A + \mu L^T L)^{-1} A^T b$$

for any $\mu > 0$. Generally,

$$\lim_{\mu \searrow 0} x_\mu = A^\dagger b, \qquad \lim_{\mu \to \infty} x_\mu = 0.$$

Neither x_0 nor x_∞ are useful approximations of x_{exact}. A proper choice of the value of μ is important. It involves computing x_μ repeatedly for different μ-values. May be expensive.

The discrepancy principle

Assume that a fairly accurate estimate for

$$\delta := \|b - b_{\text{exact}}\|_2$$

is known. The discrepancy principle prescribes that $\mu > 0$ be chosen so that

$$\|Ax_\mu - b\|_2 = \eta\delta$$

for some constant $\eta > 1$ independent of δ.

The computation of such a μ-value requires solution of the Tikhonov minimization problem for several values of μ.

Methods for repeated Tikhonov minimization

Assume that $A \in \mathbf{R}^{m\times n}$ is small and let $L = I$. Compute the SVD of A,

$$A = U\Sigma V^T,$$

where $U \in \mathbf{R}^{m\times m}$ and $V \in \mathbf{R}^{n\times n}$ are orthogonal, and

$$\Sigma = \mathrm{diag}[\sigma_1, \sigma_2, \ldots, \sigma_n] \in \mathbf{R}^{m\times n}$$

with $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0$. The Tikhonov solution is given by

$$x_\mu = V(\Sigma^T\Sigma + \mu I)^{-1}\Sigma^T U^T b.$$

The evaluation of $\|Ax_\mu - b\|_2$ requires only $\mathcal{O}(m)$ flops for every μ-value (without forming Ax_μ).

The Generalized SVD (GSVD)

Assume that $A \in \mathbf{R}^{m \times n}$ and $L \in \mathbf{R}^{p \times n}$ are small. (Here $L \neq I$). The GSVD of the matrix pair $\{A, L\}$ are the factorizations

$$A = U \Sigma X^T, \qquad L = V M X^T,$$

where $U \in \mathbf{R}^{m \times m}$ and $V \in \mathbf{R}^{p \times p}$ are orthogonal, $X \in \mathbf{R}^{n \times n}$ is nonsingular, and Σ and M are diagonal.

When $m \geq n \geq p$,

$$\begin{aligned} \Sigma &= \text{diag}[\sigma_1, \sigma_2, \ldots, \sigma_p, 1, 1, \ldots, 1] \in \mathbf{R}^{m \times n}, \\ M &= [\text{diag}[\mu_1, \mu_2, \ldots, \mu_p], 0, 0, \ldots, 0] \in \mathbf{R}^{p \times n}, \end{aligned}$$

$$\begin{aligned} &0 \leq \sigma_1 \leq \sigma_2 \leq \ldots \leq \sigma_p \leq 1, \\ &1 \geq \mu_1 \geq \mu_2 \geq \ldots \geq \mu_p \geq 0, \\ &\sigma_j^2 + \mu_j^2 = 1, \quad 1 \leq j \leq p. \end{aligned}$$

The Tikhonov solution is given by

$$x_\mu = X^{-T}(\Sigma^T \Sigma + \mu M^T M)^{-1} \Sigma^T U^T b.$$

The evaluation of $\|Ax_\mu - b\|_2$ requires only $\mathcal{O}(m)$ flops for every μ-value (without evaluating Ax_μ).

When the matrices A and L are large, the computation of the SVD of A or GSVD of the matrix pair $\{A, L\}$ is expensive.

When $A, L \in \mathbf{R}^{n \times n}$ then, roughly,

- the computation of the SVD of A requires about $10n^3$ flops, and
- the computation of the GSVD of $\{A, L\}$ requires about $25n^3$ flops.

Therefore, the evaluation of these decompositions is impractical for large-scale problems.

Methods for large-scale problems

Zha described an iterative method for determining a few vectors of the GSVD of a pair of large matrices $\{A, L\}$. Kilmer, Hansen, and Español apply this method to Tikhonov regularization. Some properties:

- It is an inner-outer iterative method. Generalized singular vectors are computed in the inner iteration.
- Zha's method may require fairly many iterations.

We are interested in developing methods that require only few matrix-vector product evaluations with A.

Application of standard Krylov subspace methods

The Arnoldi process:

Application of k steps to $A \in \mathbf{R}^{n\times n}$ with initial vector b gives the Arnoldi decomposition

$$AV_k = V_{k+1}H_{k+1,k},$$

where the orthonormal columns of $V_k \in \mathbf{R}^{n\times k}$ span the Krylov subspace

$$\mathbf{K}_k(A,b) = \text{span}\{b, Ab, A^2b, \ldots, A^{k-1}b\}$$

with $V_k e_1 = b/\|b\|_2$ and $H_{k+1,k} \in \mathbf{R}^{(k+1)\times k}$ upper Hessenberg.

We solve

$$\min_{x \in \mathbf{K}_k(A,b)} \{ \|Ax - b\|_2^2 + \mu \|Lx\|_2^2 \}$$

by using the QR factorization

$$LV_k = Q_k R_k,$$

where $Q_k \in \mathbf{R}^{n \times k}$ has orthonormal columns and $R_k \in \mathbf{R}^{k \times k}$ is upper triangular. Let $x = V_k y$. Then

$$\min_{y \in \mathbf{R}^k} \{ \|H_{k+1,k} y - e_1 \|b\|_2 \|_2^2 + \mu \|R_k y\|_2^2 \}.$$

This reduced problem can be solved by using the GSVD of $\{H_{k+1,k}, R_k\}$.

Some remarks:

- The Arnoldi process can be replaced by a range restricted Arnoldi process that generates an orthonormal basis for the solution subspace

  $$\mathbf{K}_k(A, A^j b) = \text{span}\{A^j b, A^{j+1} b, A^{j+2} b, \ldots, A^{j+k-1} b\}.$$

  Typically, $j = 1$ or $j = 2$.

- The Arnoldi process can be replaced by some other Krylov subspace method for reducing A, such as Golub–Kahan bidiagonalization.

- The solution subspace is independent of L. For some problems this is a disadvantage.

Reduction methods for matrix pairs $\{A, L\}$

Reduction method by Li and Ye:

Generalizes the Arnoldi process to matrix pairs:

$$\begin{aligned} AV_k &= V_{2k} H^{(A)}_{2k,k}, \\ LV_k &= V_{2k+1} H^{(L)}_{2k+1,k}, \end{aligned}$$

where V_{2k+1} has orthonormal columns with $V_{2k+1} e_1 = b/\|b\|$. The matrices $H^{(A)}_{2k,k}$ and $H^{(L)}_{2k+1,k}$ are upper "super Hessenberg".

Example: Matrices for $k = 4$.

$$H_{8,4}^{(A)} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ & * & * & * \\ & * & * & * \\ & & * & * \\ & & * & * \\ & & & * \\ & & & * \end{bmatrix}, \qquad H_{9,4}^{(L)} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ & * & * & * \\ & * & * & * \\ & & * & * \\ & & * & * \\ & & & * \\ & & & * \end{bmatrix}.$$

Solution subspace $\mathcal{R}(V_k)$ generated by the Li–Ye method with initial vector b is of the form

$$\mathbf{K}_k(A, L, b) = \text{span}\{b, Ab, Lb, A^2b, LAb, ALb, L^2b,$$
$$A^3b, LA^2b, ALAb, A^2Lb, LALb, AL^2b, L^3b, \dots \}$$

The method alternatingly evaluates a matrix-vector product with A and a matrix-vector product with L.

Generalized Arnoldi process for matrix pairs $\{A, L\}$:

1. **Given** q_1 **with** $\|q_1\| = 1$**;**
2. $N := 1$;
3. **for** $j = 1, 2, \ldots, k$ **do**
4. **if** $j > N$ **then exit;**
5. $\hat{q} := Aq_j$;
6. **for** $i = 1, 2, \ldots N$ **do**
7. $h_{A;i,j} := q_i^T \hat{q}$; $\hat{q} := \hat{q} - q_i h_{A;i,j}$;
8. **end for**
9. $h_{A;N+1,j} := \|\hat{q}\|$;
10. **if** $h_{A;N+1,j} > 0$ **then**
11. $N := N + 1$; $q_N := \hat{q}/h_{A;N,j}$;
12. **end if**

13. $\hat{q} := Lq_j$;
14. **for** $i = 1, 2, \ldots N$ **do**
15. $h_{L;i,j} := q_i^T \hat{q}$; $\hat{q} := \hat{q} - q_i h_{L;i,j}$;
16. **end for**
17. $h_{L;N+1,j} := \|\hat{q}\|$;
18. **If** $h_{L;N+1,j} > 0$ **then**
19. $N := N + 1$; $q_N := \hat{q}/h_{A;N,j}$;
20. **end if**
21. **end for**

The scalar N in the algorithm tracks the number of vectors q_i generated so far during the computations. Let α_k and β_k denote the values of N at the end of Lines 12 and 20, respectively, when $j = k$.

$$\begin{aligned} AQ_{(:,1:k)} &= Q_{(:,1:\alpha_k)} H_{A\,(1:\alpha_k,1:k)}, \\ LQ_{(:,1:k)} &= Q_{(:,1:\beta_k)} H_{L\,(1:\beta_k,1:k)}; \end{aligned}$$

We solve

$$\min_{x \in \mathbf{K}_k(A,L,b)} \{ \|Ax - b\|_2^2 + \mu \|Lx\|_2^2 \}$$

by using the generalized Arnoldi decompositions. Let $x = V_k y$. Then we obtain the reduced problem

$$\min_{y \in \mathbf{R}^k} \{ \|H_{2k,k}^{(A)} y - e_1 \|b\|_2 \|_2^2 + \mu \|H_{2k+1,k}^{(L)} y\|_2^2 \}.$$

It can be solved by the GSVD.

Example: We would like to determine the unavailable noise-free image represented by 412×412 pixels.

The entries of the vector $b \in \mathbf{R}^{412^2}$ store the pixel values, ordered column-wise, of the available blur- and noise-contaminated image.

The blurring matrix $A \in \mathbf{R}^{412^2 \times 412^2}$ represents severe Gaussian blur. The image also has been contaminated by 30% Gaussian noise. We apply the Li–Ye method to solve

$$\min_{x \in \mathbf{K}_k l(A,L,b)} \{\|Ax - b\|_2^2 + \mu \|Lx\|_2^2\}$$

for two different regularization matrices L:

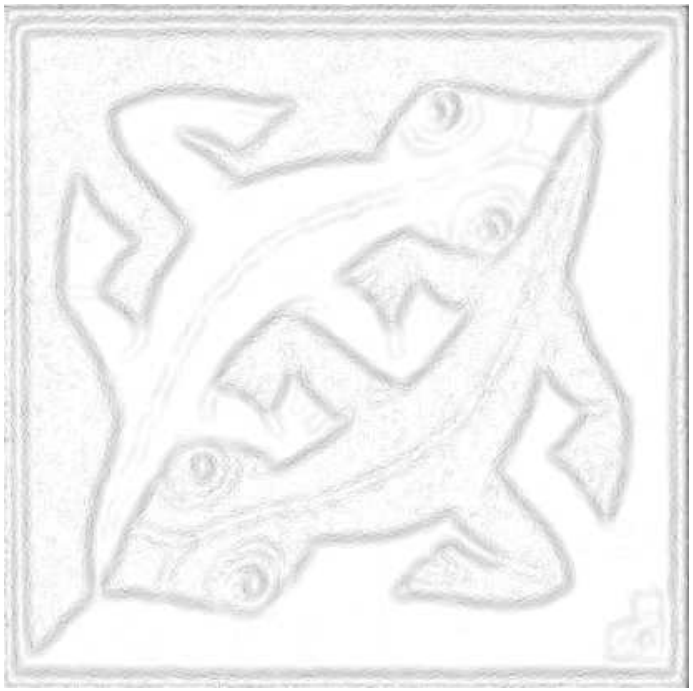
- $L = \Delta$, the standard discrete Laplace operator based on the five-point stencil.
- L is a discretized and linearized Perona–Malik operator:

$$\mathcal{L}(x) = \operatorname{div}(g(|\nabla x|^2)\nabla x), \quad g(s) = \frac{1}{1 + \frac{s}{\rho}}, \quad \rho = 10^{-4}.$$

Restored image using $L = \Delta$. 6 generalized Arnoldi steps.

Restored image with L determined by the Perona–Malik operator. Two step of GMRES give an approximate restoration with which $\mathcal{L}$ is defined.

Edge map for restoration with Perona–Malik operator.



Some remarks:

- To work well with the discrepancy principle, e_1 should be replaced by $P_{\mathcal{R}(H_{2\ell,\ell}^{(A)})}e_1$, i.e.,

$$\begin{aligned} \|Ax_\mu - b\|_2 &= \|H_{2k,k}^{(A)} y_\mu - e_1\|b\|_2\|_2 \\ &\geq \|H_{2k,k}^{(A)} y_\mu - P_{\mathcal{R}(H_{2k,k}^{(A)})}e_1\|b\|_2\|_2. \end{aligned}$$

  The discrepancy principle is applied to the right-hand side.

- The method requires the generation of about twice as many orthonormal vectors as the dimension of the solution subspace.

Reduction method based on the flexible Arnoldi process:

Let $A \in \mathbf{R}^{n \times n}$. Apply k steps of the flexible Arnoldi process (due to Saad) to A with initial vector b. This gives a decomposition

$$AV_k = U_{k+1} H_{k+1,k},$$

where U_{k+1} has orthonormal columns with
$U_{k+1} e_1 = b/\|b\|$.

Columns of V_k arbitrary. We use the QR factorization

$$LV_k = Q_k R_k.$$

The flexible Arnoldi algorithm

0. Input $A \in \mathbf{R}^{n\times n}$, $\{v_j\}_{j=1}^{k} \subset \mathbf{R}^n$, $b \in \mathbf{R}^n$;
1. Let $u_1 = b/\|b\|_2$;
2. `for` $j = 1, \ldots, k$ `do`
   - 2.1. $w = Av_j$;
   - 2.3. `for` $i = 1, \ldots, j$ `do`

     $$h_{i,j} = w^T u_i;\ w = w - h_{i,j}u_i;$$

   - 2.4. `end for`
   - 2.5. $h_{j+1,j} = \|w\|_2$;
   - 2.6. $u_{j+1} = w/h_{j+1,j}$;
3. `end for`

Then

$$\min_{x \in \mathcal{R}(V_k)} \{ \|Ax - b\|_2^2 + \mu \|Lx\|_2^2 \}$$

simplifies to the small problem

$$\min_{y \in \mathbf{R}^k} \{ \|H_{k+1,k} y - e_1 \|b\|_2 \|_2^2 + \mu \|R_k y\|_2^2 \}$$

which we can solve with the GSVD.

We determine the column v_{j+1} of V_ℓ by evaluating

$$w = Av_j \quad \text{or} \quad w = Lv_j$$

and then orthogonalizing w against the columns of V_j.

Example: Alternate between $w = Av_j$ and $w = Lv_j$.
Then

$$\mathcal{R}(V_k) = \text{span}\{b, Ab, Lb, A^2b, LAb, ALb, L^2b, A^3b, \ldots\ \}.$$

If the use of 4 vectors $w = Lv_j$ is followed by one vector $w = Lv_j$ in a cyclic fashion, then

$$\mathcal{R}(V_k) = \text{span}\{b, Lb, L^2b, L^3b, L^4b, Ab, L^5b, \ldots\ \}.$$

The latter space often gives better results than the former when L is a difference operator.

Example: Consider the inverse Laplace transform

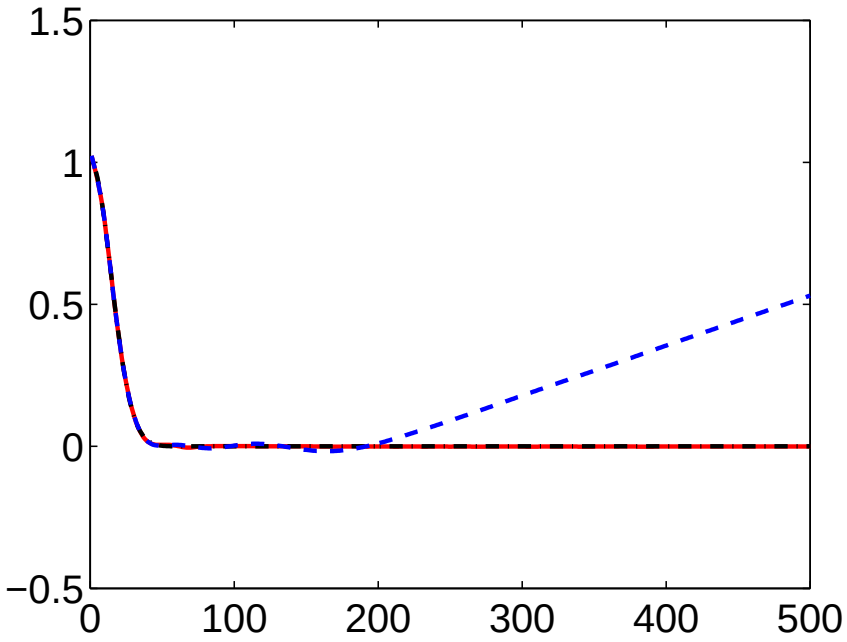
$$\int_0^\infty \exp(-st)x(t)dt = \frac{1}{s+1/2}, \qquad 0 \le s < \infty,$$

whose solution is $x(t) = \exp(-t/2)$. Discretize by MATLAB function i_laplace from Regularization Tools. Gives $A \in \mathbf{R}^{500\times 500}$ and discretized scaled solution $\widehat{x} \in \mathbf{R}^{500}$. The data vector b has 0.1% Gaussian noise.

The regularization matrix is tridiagonal and zero padded:

$$L = \begin{bmatrix} 0 & 0 & \dots & & & 0 \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \\ 0 & \dots & & & 0 & 0 \end{bmatrix} \in \mathbf{R}^{500\times 500}.$$

Use $w = Av_j$ every 50th step. Figure shows computed solution x_μ after 92 steps (red solid curve), GSVD solution (blue dashed curve), and desired solution $\widehat{x}$ (blue dash-dotted curve).



Some remarks:

- The method allows much flexibility in the choice of solution subspace.
- The method requires A and L to be square.
- The flexible Arnoldi process can be applied without Tikhonov regularization.

The flexible Arnoldi process and truncated iteration:

Flexible Arnoldi gives sequence of decompositions

$$AV_k = U_{k+1} H_{k+1,k}, \quad k = 1, 2, 3, \ldots ,$$

where U_{k+1} has orthonormal columns, $U_{k+1} e_1 = b/\|b\|$. We let V_k have orthonormal columns. Then

$$\min_{x \in \mathcal{R}(V_k)} \|Ax - b\|_2 = \min_{y \in \mathbf{R}^k} \|H_{k+1,k} y - e_1 \|b\|_2 \|_2.$$

Denote solution by y_k. Terminate the iterations as soon as

$$\|H_{k+1,k} y_k - e_1 \|b\|_2 \|_2 \leq \delta. \qquad \text{(discrepancy principle)}$$

Gives similar results as flexible Arnoldi with Tikhonov regularization for $L = I$.

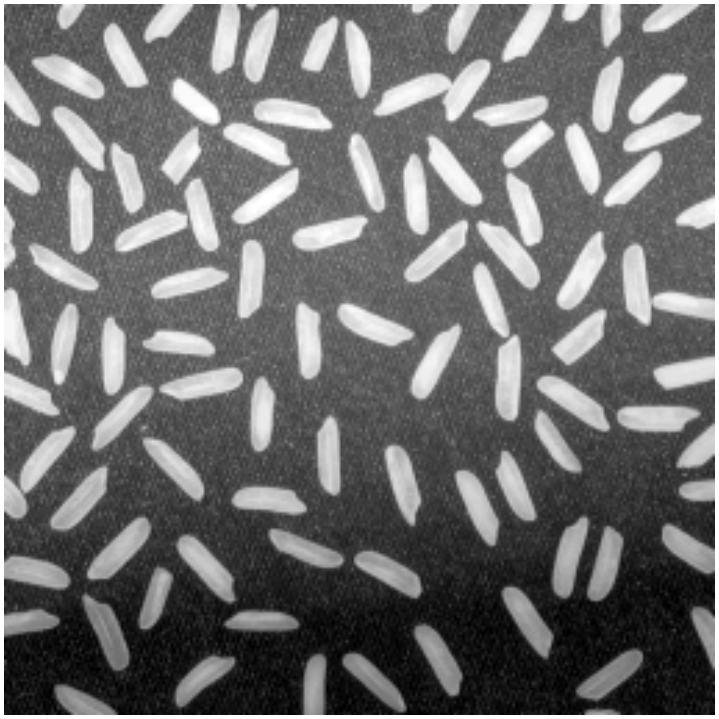
A simple extension of the flexible Arnoldi-based method:

We determine the last column v_{j+1} of V_{j+1} by evaluating

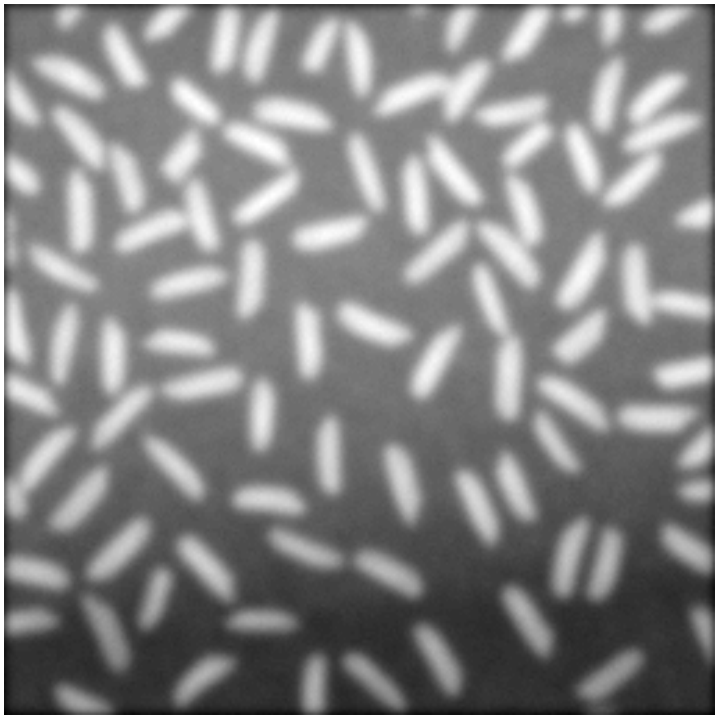
$$w = Av_j \quad \text{or} \quad w = L^*Lv_j$$

and then orthogonalizing w against the columns of V_j.
This allows L to be rectangular.

Example: We would like to determine the unavailable noise-free image represented by 256×256 pixels.



The entries of the vector $b \in \mathbf{R}^{256^2}$ store the pixel values, ordered column-wise, of the available image contaminted by Gaussian blur and 1% Gaussian noise.

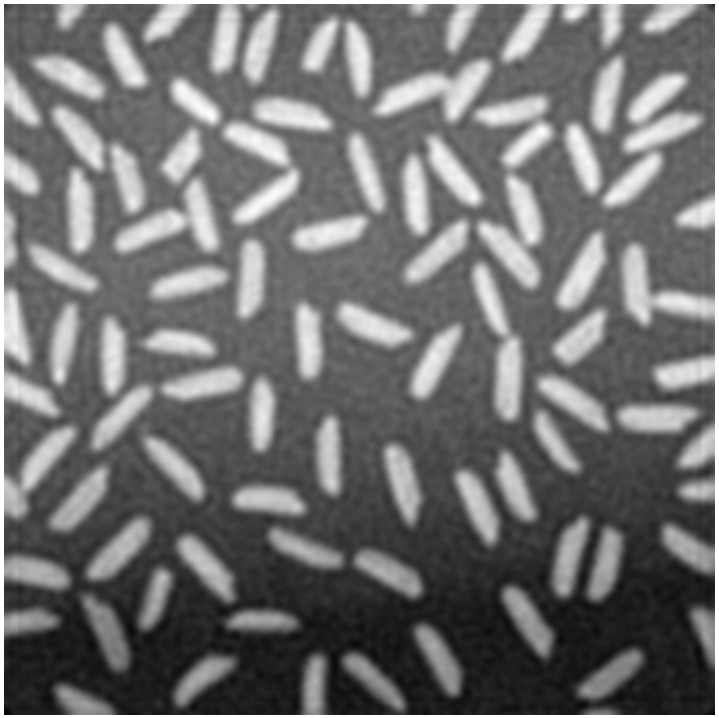


The regularization matrix is given by

$$L = \begin{bmatrix} I & \otimes & L_1 \\ L_1 & \otimes & I \end{bmatrix}, \quad L_1 = \frac{1}{2}\begin{bmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & 1 & -1 \end{bmatrix}$$

with $I \in \mathbf{R}^{256\times256}$, $L_1 \in \mathbf{R}^{255\times256}$, and $L \in \mathbf{R}^{130560\times65536}$.

Restored image after 22 steps with one vector $w = Av_j$ for every 10 vectors $w = L^*Lv_j$ for constructing the solution subspace.



Some remarks:

- The method allows a lot of flexibility in the choice of solution subspace and regularization matrix.
- The method requires A to be square.

A generalized Golub–Kahan-type reduction method for matrix pairs.

Matrix-vector products are evaluated with the matrices A, L, A^T, and L^T in a periodic fashion. With initial vector b, we have after k steps

$$\begin{aligned} AV_k &= U_{k+1}H_{k+1,k}, & LV_k &= W_kK_{k,k}, \\ A^TU_k &= V_{2k-2}\,H_{k,2k-2}^T, & L^TW_k &= V_{2k+1}\,K_{k,2k+1}^T, \end{aligned}$$

where U_{k+1}, V_{2k+1}, and W_k have orthonormal columns with $U_{k+1}e_1 = b/\|b\|$.

The matrix H has the structure

$$H = \begin{bmatrix} \times & & & & & & & \\ \times & \times & & & & & & \\ & \times & \times & \times & & & & \\ & & \times & \times & \times & \times & & \\ & & & \times & \times & \times & \times & \times \end{bmatrix}$$

and

the structure of K is given by

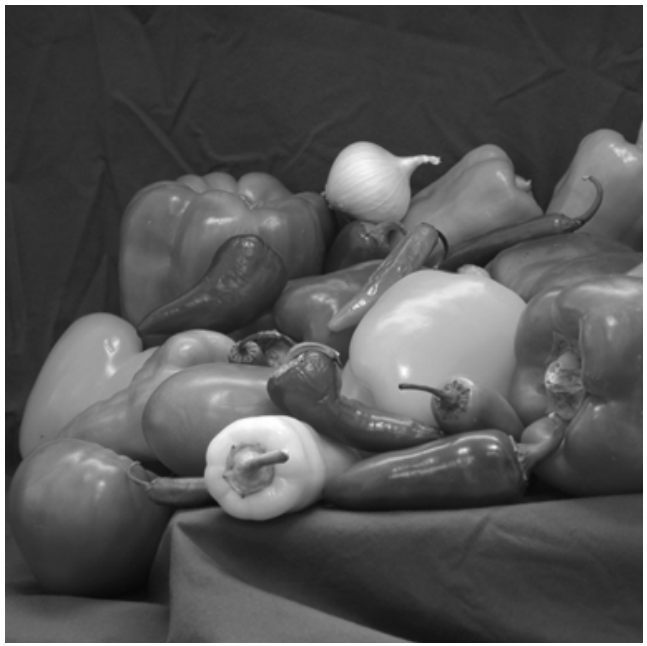
$$K = \begin{bmatrix} \times & \times & \times & & & & & & \\ & \times & \times & \times & \times & & & & \\ & & \times & \times & \times & \times & \times & & \\ & & & \times & \times & \times & \times & \times & \times \end{bmatrix}.$$

The algorithm has short recurrence relations with the number of terms increasing with the number of steps k.

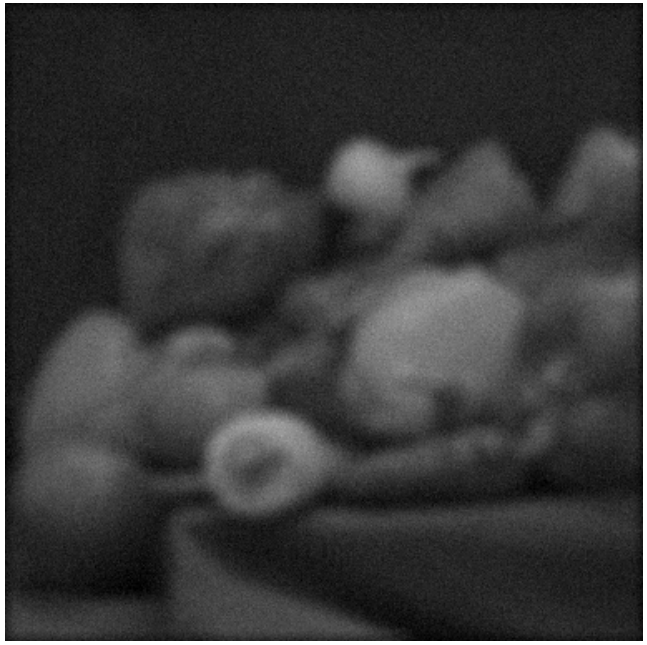
The solution subspace is of the form

$$\begin{aligned}\mathcal{R}(V_k) \;=\; & \text{span}\{A^*b, (A^*A)A^*b, (B^*B)A^*b, \\ & (A^*A)^2A^*b, (B^*B)(A^*A)A^*b, \\ & (A^*A)(B^*B)(A^*A)A^*b, (B^*B)^2A^*b, \ldots \;\}.\end{aligned}$$

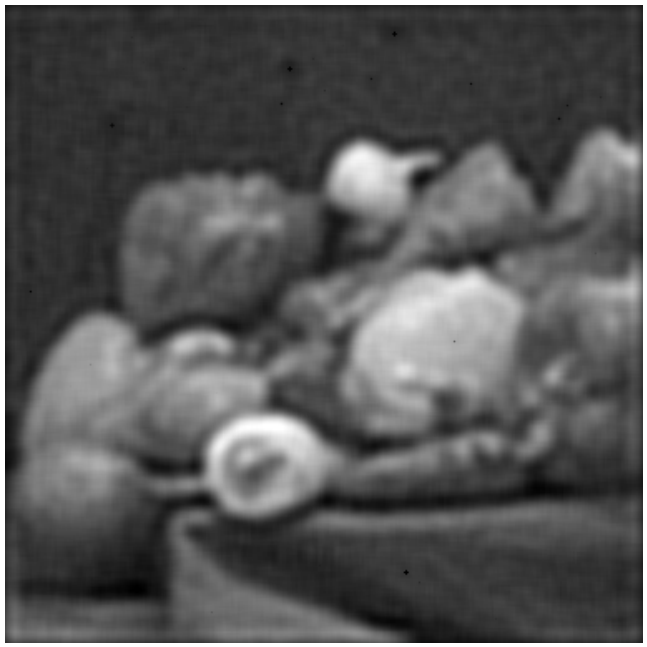
Example: We would like to determine the unavailable noise-free image represented by 384×384 pixels.



The entries of the vector $b \in \mathbf{R}^{384^2}$ store the pixel values, ordered column-wise, of the available image contaminted by Gaussian blur and 10% Gaussian noise.



Restored image after 7 steps and regularization matrix determined by a discretization and linearization of the Perona–Malik operator, similarly as above.



Observations:

- A variety of iterative methods can be derived for the solution of discrete ill-posed problems with pairs of large matrices. Extensions to matrix n-tuplets is straightforward. They are of interest for multiparameter Tikhonov regularization.
- Iterative methods may determine approximate solutions of higher quality than direct solution methods.

The Singular value decomposition applied to large-scale ill-posed problems

Let $A \in \mathbf{R}^{n\times n}$, $b \in \mathbf{R}\backslash\{0\}$. The symmetric Lanczos process applied to A with initial vector b gives the Lanczos decomposition

$$AV_k = V_{k+1}T_{k+1,k},$$

where the matrix

$$V_{k+1} = [V_k, v_{k+1}] = [v_1, v_2, \ldots, v_{k+1}] \in \mathbf{R}^{n\times(k+1)}$$

has orthonormal columns such that

$$\mathcal{R}(V_{k+1}) = \mathbf{K}_{k+1}(A, b) = \text{span}\{b, Ab, \ldots, A^k b\}.$$

Moreover, the matrix

$$T_{k+1,k} = \begin{bmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{k-1} & \alpha_{k-1} & \beta_k \\ & & & \beta_k & \alpha_k \\ & & & & \beta_{k+1} \end{bmatrix} \in \mathbf{R}^{(k+1)\times k}$$

is tridiagonal, and $T_{k,k}$ is the leading $k \times k$ symmetric submatrix.

The Lanczos decomposition can be computed by the symmetric Lanczos algorithm.

The symmetric Lanczos algorithm.

1: **Input:** Symmetric matrix $A \in \mathbf{R}^{n \times n}$, initial vector $b \in \mathbf{R}^m$, number of steps k.

2: $v_0 = 0$, $\beta_1 = \|b\|_2$, $v_1 = b/\beta_1$

3: **for** $j = 1$ **to** k

4: $\quad \widetilde{v} = Av_j - \beta_j v_{j-1}$

5: $\quad \alpha_j = v_j^T \widetilde{v}$

6: $\quad \widetilde{v} = \widetilde{v} - \alpha_j v_j$

7: $\quad \beta_{j+1} = \|\widetilde{v}\|_2$

8: $\quad v_{j+1} = \widetilde{v}/\beta_{j+1}$

9: **end for**

10: **Output:** Lanczos decompositions.

Let A stem from the discretization of an ill-posed problem and assume that b is contaminated by error.

Instead of solving the least-squares problem

$$\min_{x \in \mathbf{R}^n} \|Ax - b\|_2,$$

we compute an approximate solution of the reduced problem

$$\begin{aligned}
\min_{x \in \mathbf{K}_k(A,b)} \|Ax - b\|_2 &= \min_{y \in \mathbf{R}^k} \|AV_k y - b\|_2 \\
&= \min_{y \in \mathbf{R}^k} \|V_{k+1} T_{k+1,k} y - V_{k+1} e_1 \|b\|_2 \|_2 \\
&= \min_{y \in \mathbf{R}^k} \|T_{k+1,k} y - e_1 \|b\|_2 \|_2.
\end{aligned}$$

Define the spectral factorization

$$A = W \Lambda W^T,$$

where $W \in \mathbf{R}^{n \times n}$ is orthogonal and

$$\Lambda = \mathrm{diag}[\lambda_1, \ldots, \lambda_n] \in \mathbf{R}^{n \times n}$$

with

$$|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_n| \geq 0.$$

Theorem 1: Let A be symmetric positive semidefinite. Assume that the Lanczos process applied to A does not break down, i.e., that n steps can be carried out. Define $\beta_{n+1} = 0$. Then

$$\prod_{j=2}^{k+1} \beta_j \leq \prod_{j=1}^{k} \lambda_j, \qquad k = 1, 2, \ldots, n.$$

Proof: Define the monic polynomial $p_k(t) = \prod_{j=1}^{k}(t - \lambda_j)$ defined by the k largest eigenvalues of A. Then

$$\|p_k(A)\|_2 = \|p_k(\Lambda)\|_2 = \max_{k+1 \leq j \leq n} |p_k(\lambda_j)| \leq |p_k(0)| = \prod_{j=1}^{k} \lambda_j.$$

Therefore,

$$\|p_k(A)b\|_2 \leq \|b\|_2 \prod_{j=1}^{k} \lambda_j.$$

Application of n steps of the Lanczos process gives

$$AV_n = V_n T_n, \qquad V_n \in \mathbf{R}^{n\times n} \text{ orthogonal}$$

Hence,

$$p_k(A)b = \widehat{V}_n p_k(T_n)\widehat{V}_n^T b = \widehat{V}_n p_k(T_n)e_1\|b\|$$

and

$$\|p_k(A)b\|_2 = \|p_k(T_n)e_1\|_2\|b\|_2 \geq \|b\|_2 \prod_{j=2}^{k+1} \beta_j.$$

The last inequality follows by direct computations. □

Corollary 1. Let $A \in \mathbf{R}^{n\times n}$ be symmetric positive semidefinite. Assume that the eigenvalues of A "cluster" at the origin and that the Lanczos method applied to A does not break down. Further, assume that there is a constant M independent of j such that

$$\beta_{j+1} \leq M \min\{\beta_1, \beta_2, \ldots, \beta_j\}, \quad j = 1, 2, \ldots \ .$$

Then both the subdiagonal and diagonal entries of $T_{\ell+1,\ell}$ decrease to zero as the row number increases (and is large enough).

Proof. The decrease of the subdiagonal entries of $T_{\ell+1,\ell}$ follows from Theorem 1. The matrix T_n is similar to A. Therefore its eigenvalues cluster at zero. Since the

off-diagonal of T_n entries are tiny, the eigenvalues are close to the diagonal entries. They therefore also have to be tiny. $\square$

Corollary 2: Let A be symmetric. Assume that the Lanczos process applied to A does not break down, i.e., that n steps can be carried out. Define $\beta_{n+1} = 0$. Then

$$\prod_{j=2}^{k+1} \beta_j \leq \prod_{j=1}^{k} (2|\lambda_j|), \qquad k = 1, 2, \ldots, n.$$

The requirement that n steps of the Lanczos process can be removed by bounding $k < n$.

Corollary 3: Under the conditions of Corollary 2, the span of the Lanczos vector v_k is an accurate approximations the span of kth eigenvector for large k.

Proof: This follows from the fact that $\beta_j \searrow 0$ as j increases. $\square$

Consequences:

- It may not be necessary to compute the EVD of a large matrix - just use a few steps of Lanczos tridiagonalization. It is cheaper.
- If it is convenient to use the EVD of a large matrix A instead of a partial Lanczos tridiagonalization, then its computation requires only very few steps with a restarted Lanczos tridiagonalization method. This follows from the fact that the span of Lanczos vectors with large index is close to the span of corresponding eigenvectors.

Non-symmetric problems

$k \ll n$ steps of Golub-Kahan bidiagonalization (GKB) applied to $A \in \mathbf{R}^{m\times n}$ with initial vector $\widehat{u}_1 = b/\|b\|$ gives the decompositions

$$A\widehat{V}_k = \widehat{U}_{k+1}B_{k+1,k}, \qquad A^T\widehat{U}_k = \widehat{V}_k B_{k,k}^T,$$

where

$$\begin{aligned}
\widehat{U}_{k+1} &= [\widehat{U}_k, \widehat{u}_{k+1}] = [\widehat{u}_1, \widehat{u}_2, \ldots, \widehat{u}_{k+1}] \in \mathbf{R}^{m\times(k+1)}, \\
\widehat{V}_k &\in \mathbf{R}^{n\times k}, \quad \widehat{U}_{k+1}^T\widehat{U}_{k+1} = I, \quad \widehat{V}_k^T\widehat{V}_k = I, \\
\mathcal{R}(\widehat{V}_k) &= \mathbf{K}_k(A^TA, A^Tb) = \mathrm{span}\{A^Tb, \ldots, (A^TA)^{k-1}A^Tb\}.
\end{aligned}$$

Moreover,

$$B_{k+1,k} = \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_k & \alpha_k \\ & & & \beta_{\ell+1} \end{bmatrix} \in \mathbf{R}^{(k+1)\times k}$$

is lower bidiagonal with leading $k \times k$ submatrix $B_{k,k}$.

Instead of solving the original least-squares problem, we solve the reduced problem

$$\begin{aligned}\min_{x\in\mathbf{K}_k(A^TA,A^Tb)} \|Ax-b\|_2 &= \min_{y\in\mathbf{R}^k} \|A\widehat{V}_k y - b\|_2 \\ &= \min_{y\in\mathbf{R}^k} \|B_{k+1,k}y - e_1\|b\|_2\|_2 \longrightarrow y_k.\end{aligned}$$

The solution $x_k^{\mathrm{GKB}} := \widehat{V}_k y_k$ is cheaper to compute than x_k^{TSVD}.

Theorem 2: Let $A \in \mathbf{R}^{m \times n}$, $m \geq n$, have the singular values $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0$. Assume that the GKB applied to A with initial vector $u_1 = b/\|b\|_2$ does not break down. Let

$$C_{k+1,k} = \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_k & \alpha_k \\ & & & \beta_{k+1} \end{bmatrix}.$$

Then

$$\prod_{j=2}^{k+1} \alpha_j \beta_j \leq \prod_{j=1}^{k} \sigma_j^2, \qquad k = 1, 2, \ldots, n-1.$$

Assume that there is a constant M such that

$$\alpha_{j+1}\beta_{j+1} \leq M \min\{\alpha_1\beta_1, \alpha_2\beta_2, \ldots, \alpha_j\beta_j\}, \quad j = 1, 2, \ldots \ .$$

Then the products $\alpha_j\beta_j \searrow 0$ as j increases.

Proof: The result can be shown, e.g., by first considering the application of the symmetric Lanczos method to a symmetric positive definite matrix. Application of GKB to A is equivalent to application of the symmetric Lanczos method to $A^T A$. $\square$

Corollary 4: Under the conditions of Theorem 2, the span of the GKB vector $\widehat{v}_k$ is an accurate approximations the span of kth left singular vector for large k.

Proof: This follows from the fact that $\alpha_j \beta_j \searrow 0$ as j increases. $\square$

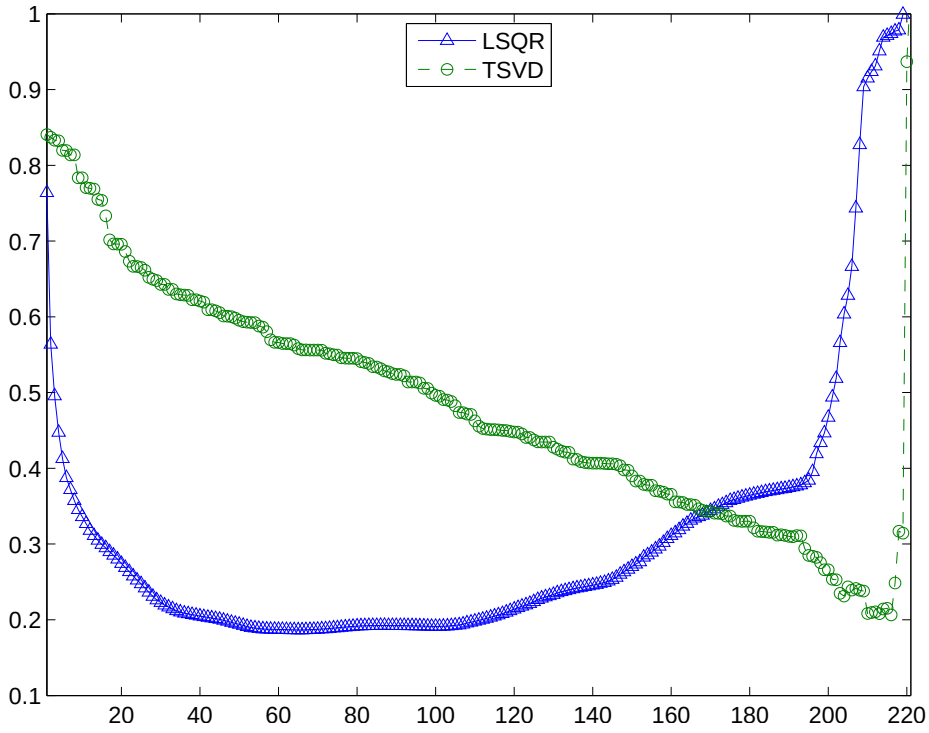
Consequences:

- It may not be necessary to compute the SVD of a large matrix - just use GKB. It is cheaper.
- If it is convenient to use the SVD of a large matrix A instead of a GKB, then its computation requires only very few steps with a restarted Lanczos bidiagonalization method, This follows from the fact that the span of GKB vectors with large index is close to the span of corresponding singular vectors.

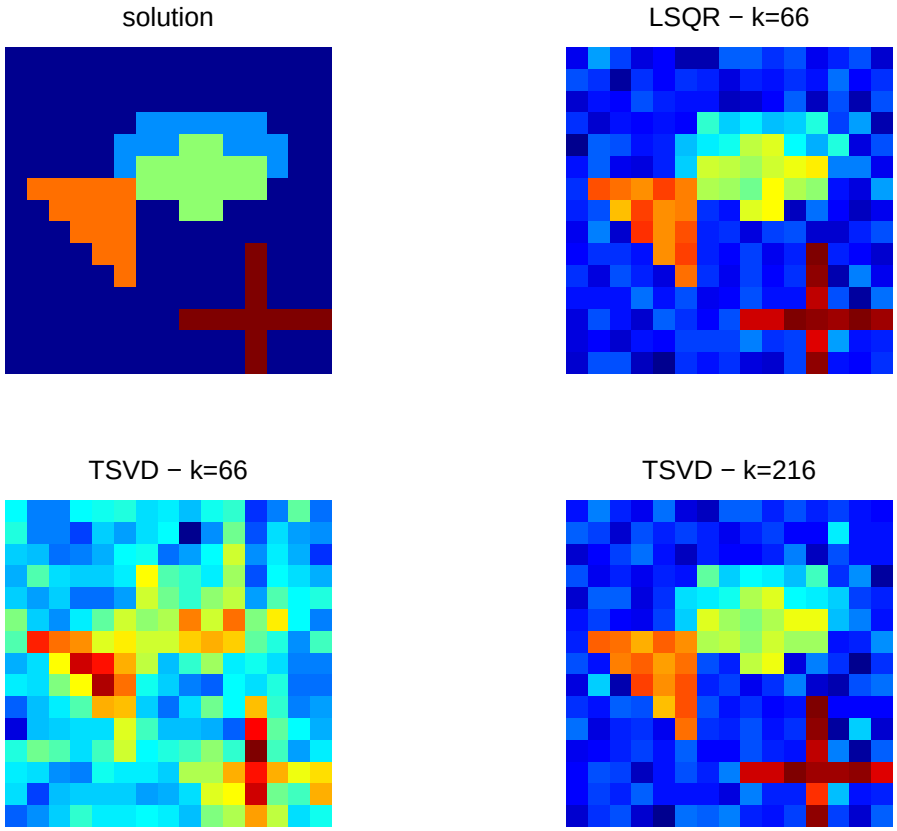
Example: Test problem Tomo from Regularization Tools by Hansen. It arises from the discretization of a 2D tomography problem. Yields a linear system

$$Ax = b, \qquad A \in \mathbf{R}^{225\times 225}, \quad x, b \in \mathbf{R}^{225}.$$

1% relative error in b.



Convergence history for GKB and TSVD. GKB solution error minimal at step $\ell = 66$; TSVD solution error minimal at step $\ell = 216$.



Exact and computed solutions by GKB (=LSQR) at step 66 and TSVD at steps 66 and 216.

Example: Discretaization of integral equation "baart" from Regularization Tools by Hansen,

$$\int_0^\pi \exp(-st)x(t)dt = 2\frac{\sinh(s)}{s}, \qquad 0 \le s \le \frac{\pi}{2},$$

by Galerkin method with box functions as test and trial functions. Gives matrix $A \in \mathbf{R}^{500\times 500}$.

Apply restarted Lanczos bidiagonalization method to determine the k largest singular triplets.

Number of desired singular triplets k	Size of the largest bidiagonal matrix	Number of matrix-vector products
10	$\lceil 1.5k \rceil$	30
15	$\lceil 1.5k \rceil$	46
20	$\lceil 1.5k \rceil$	60
25	$\lceil 1.5k \rceil$	76

Number of desired singular triples k	Size of the largest bidiagonal matrix	Number of matrix-vector products
10	$k+1$	22
15	$k+1$	32
20	$k+1$	42
25	$k+1$	52