Numerical Linear Algebra

Least squares problems

Gerard Sleijpen and Martin van Gijzen

November 9, 2016

Program Lecture 9

- Least squares problems
- The SVD
- Regularisation
- CG for the normal equations
- LSQR and Bi-diagonalization

Least squares problems

In this lesson we consider the problem

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

with $\mathbf{A} \in \mathbb{C}^{n \times k}$, $\mathbf{x} \in \mathbb{C}^k$, and $\mathbf{b} \in \mathbb{C}^n$ (as in Exercise 3.18).

Furthermore,

- The system may be inconsistent $(\mathbf{b} \notin \mathcal{R}(\mathbf{A}))$.
- Usually $k \ll n$.
- The rank of $\mathbf{A}$ may be smaller than k.

Least squares problems (2)

The system $\mathbf{A}\mathbf{x} = \mathbf{b}$ may be inconsistent. We therefore solve it in the **sense of least squares**, meaning that we solve the minimisation problem $\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2,$ i.e.,

$$\mathbf{x}_{LS} = \text{argmin}_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2$$

Solutions $\mathbf{x}_{LS}$ to this problem satisfy the **normal equations**

$$\mathbf{A}^* \mathbf{A}\mathbf{x}_{LS} = \mathbf{A}^* \mathbf{b}$$

and hence

$$\mathbf{r}_{LS} = \mathbf{b} - \mathbf{A}\mathbf{x}_{LS} \perp \mathcal{R}(\mathbf{A})$$

If $\text{rank}(\mathbf{A}) < k$ the least squares solution is not unique.

Least squares problems (3)

Suppose rank$(\mathbf{A}) < k$ and $\mathbf{x}_{LS}$ is a least-squares solution. Then

$$\widehat{\mathbf{x}} = \mathbf{x}_{LS} + \mathbf{y} \quad \text{with} \quad \mathbf{y} \in \mathcal{N}(\mathbf{A})$$

is also a least squares solution.

The least square solution with minimum norm is unique.
This **Least Square Minimal Norm** solution $\mathbf{x}_{LSMN}$ solves the constrained problem

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 \quad \text{subject to} \quad \mathbf{x} \perp \mathcal{N}(\mathbf{A}).$$

The Singular Value Decomposition

Let $\mathbf{A} \in \mathbb{C}^{n\times k}$ be a matrix of rank r.

There exist unitary matrices $\mathbf{U} \in \mathbb{C}^{n\times n}$ and $\mathbf{V} \in \mathbb{C}^{k\times k}$ such that

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^*, \qquad \Sigma = \begin{pmatrix} \Sigma_r & 0 \\ 0 & 0 \end{pmatrix},$$

where $\Sigma \in \mathbb{R}^{n\times k}$ and $\Sigma_r = \text{diag}(\sigma_1, \sigma_2, \cdots, \sigma_r)$, such that

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0 = \sigma_{r+1} = \ldots = \sigma_{\min(k,n)}.$$

$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^*$ is the **SVD** of $\mathbf{A}$ with **singular triples** $(\sigma_i, \mathbf{v}_i, \mathbf{u}_i)$:

σ_i is called a **singular value** of $\mathbf{A}$,

with **left singular vector** $\mathbf{u}_i \equiv \mathbf{U}\mathbf{e}_i$,

and **right singular vector** $\mathbf{v}_i \equiv \mathbf{V}\mathbf{e}_i$.

The SVD (2)

$\mathbf{A}$ maps from $\mathbb{C}^k$ to $\mathbb{C}^n$.

The columns of $\mathbf{U}$ form an orthonormal basis of the image space $\mathbb{C}^n$, while the columns of $\mathbf{V}$ form an orthonormal basis if the domain space $\mathbb{C}^k$. The diagonal matrix Σ is the matrix representation of $\mathbf{A}$ with respect to these basis.

If $\mathbf{U}_r$ and $\mathbf{V}_r$ consists of the first r columns of $\mathbf{U}$ and $\mathbf{V}$ respectively, then

$$\mathbf{A} = \mathbf{U}_r \, \Sigma_r \, \mathbf{V}_r^*$$

is the **economical** form of the **SVD**.

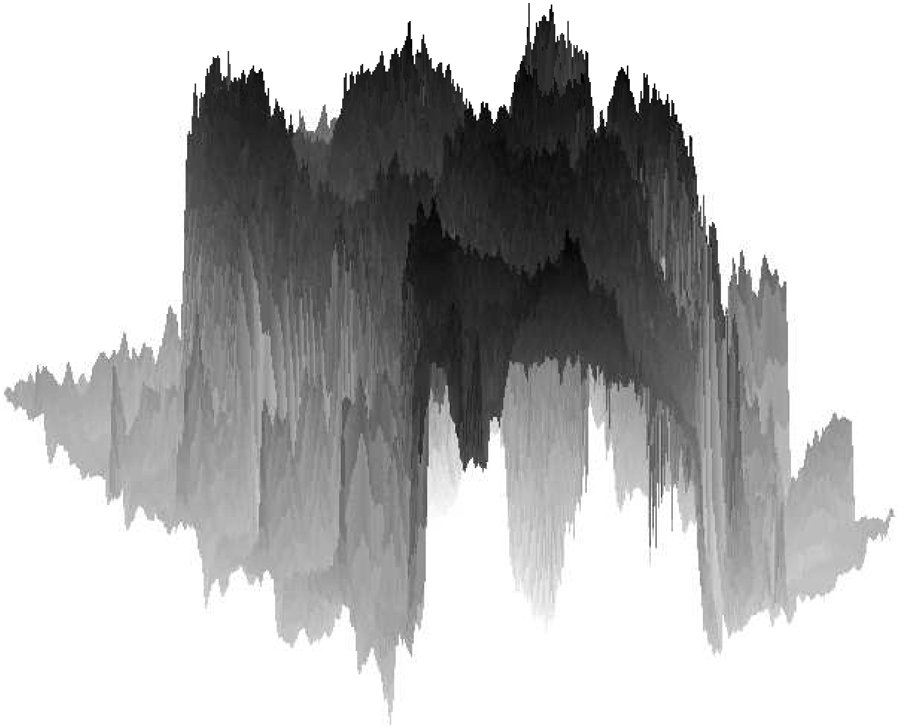
$\mathbf{U}_r$ spans $\mathcal{R}(\mathbf{A})$, while $\mathbf{V}_r$ spans $\mathcal{N}(\mathbf{A})^\perp$, r is the rank of $\mathbf{A}$.

The SVD, applications

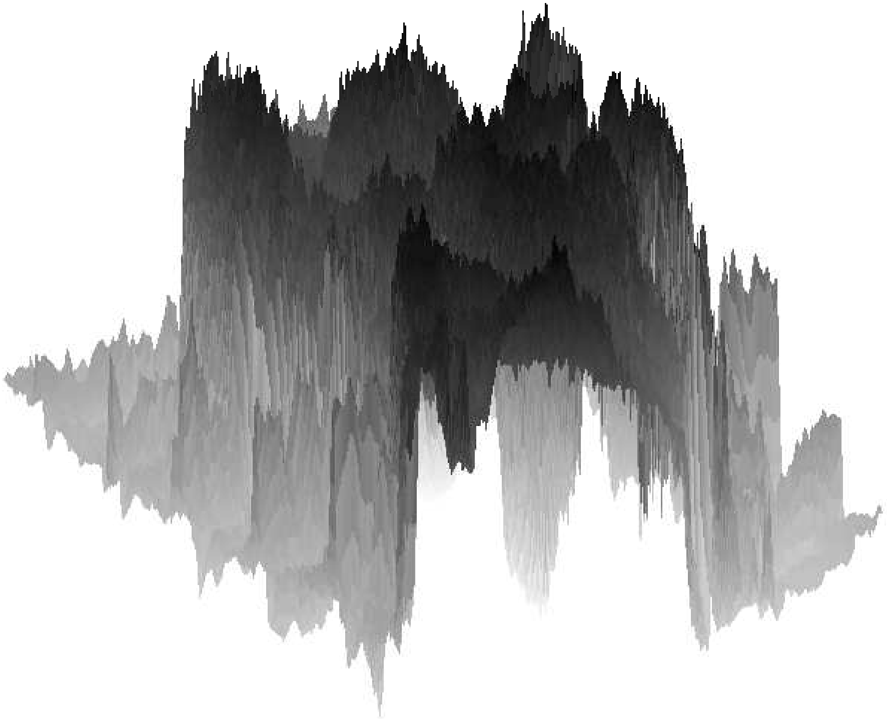
- For numerically solving LSMN problems of modest dim..
- For **theoretical analysis** of LSMN problems.
- $\|\mathbf{A}\|_2 = \max\|\mathbf{A}\mathbf{x}\|_2 = \sigma_1,\ \ \min\|\mathbf{A}\mathbf{x}\|_2 = \sigma_k,\ \ \mathcal{C}_2(\mathbf{A}) = \frac{\sigma_1}{\sigma_k}$,

  max and min over all normalized $\mathbf{x}$, i.e., $\|\mathbf{x}\|_2 = 1$.
- Geometry: angles between spaces $\mathcal{V}$ and $\mathcal{W}$ can be obtained from the SVD of $\mathbf{V}^*\mathbf{W}$, where the orthonormal matrices $\mathbf{V}$ and $\mathbf{W}$ span the space $\mathcal{V}$ and $\mathcal{W}$, respectively.
- Compression: Often the first, say p, singular vectors and singular values contain the essential information of the matrix (p can be modest even if $\sigma_{p+1}, \ldots$ are not negligible see the following example). This fact is exploited by methods as **Principal Orthogonal Directions**.

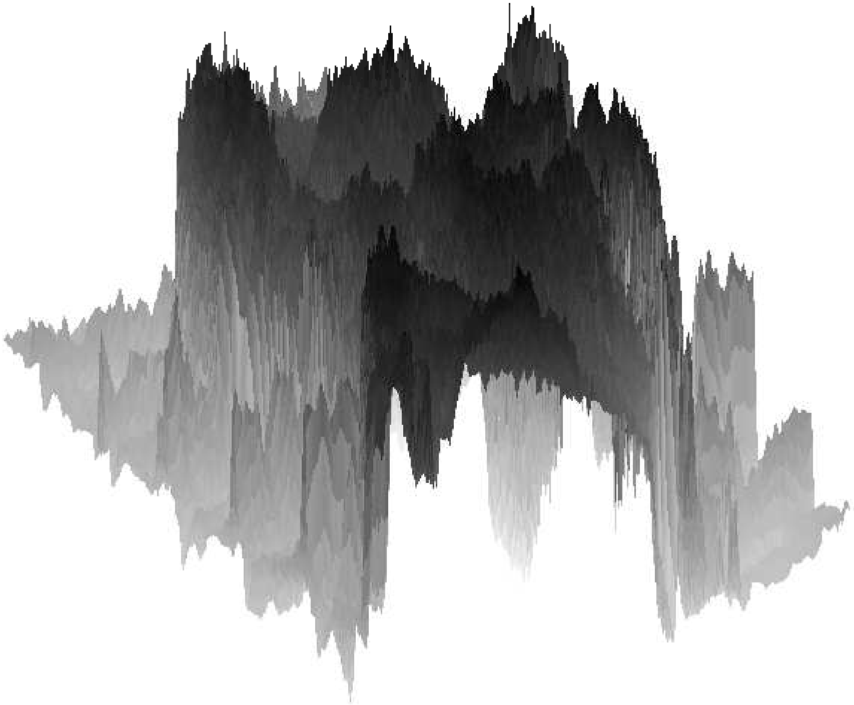
Matrices can be viewed as functions on a (discrete) rectangular domain: $(i, j) \rightsquigarrow a_{ij}$.



Matrices can be viewed as functions on a (discrete) rectangular domain: $(i, j) \rightsquigarrow a_{ij}$.

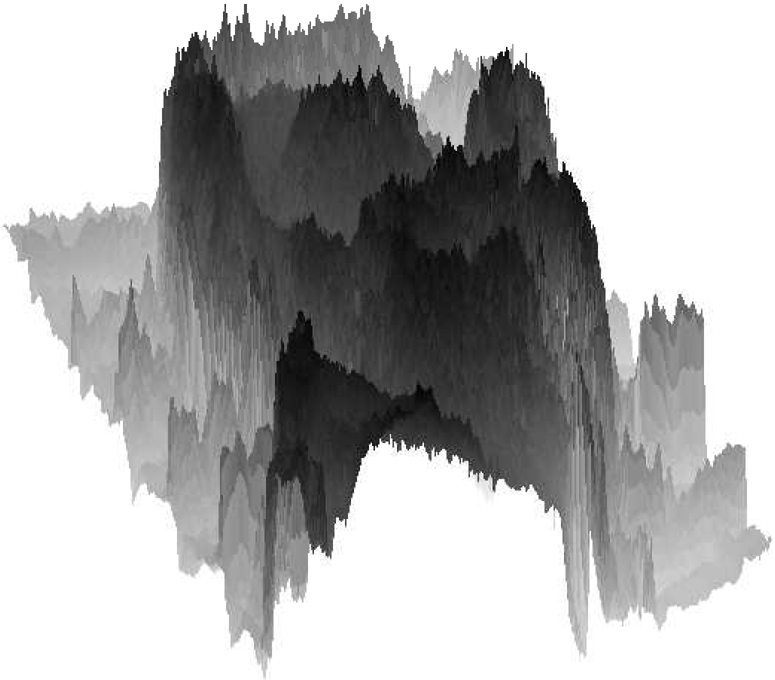


Matrices can be viewed as functions on a (discrete) rectangular domain: $(i, j) \rightsquigarrow a_{ij}$.

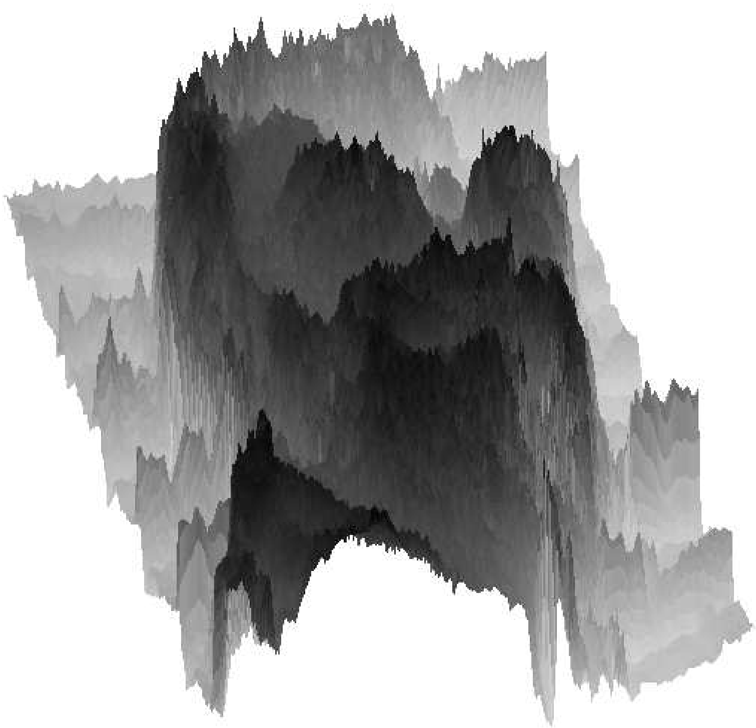


Matrices can be viewed as functions on a (discrete) rectangular domain: $(i, j) \rightsquigarrow a_{ij}$.

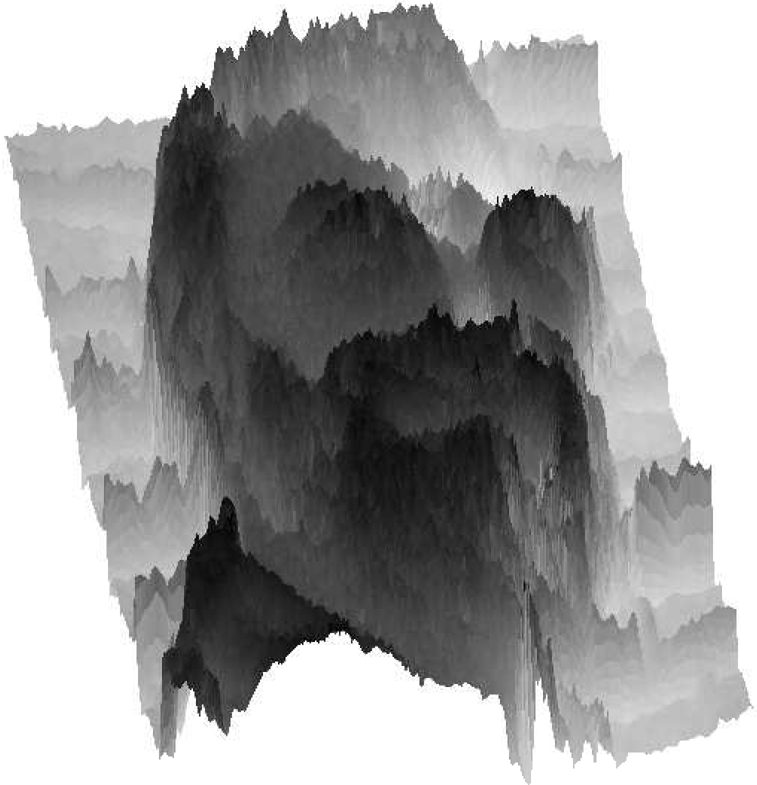
Matrices can be viewed as functions on a (discrete) rectangular domain: $(i, j) \rightsquigarrow a_{ij}$.



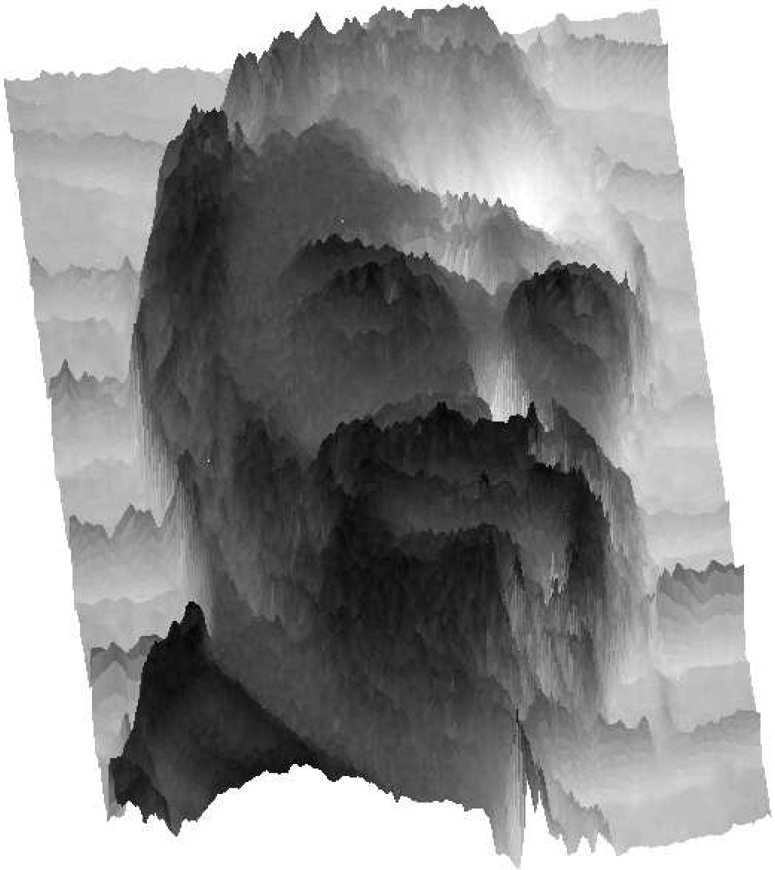
Matrices can be viewed as functions on a (discrete) rectangular domain: $(i, j) \rightsquigarrow a_{ij}$.



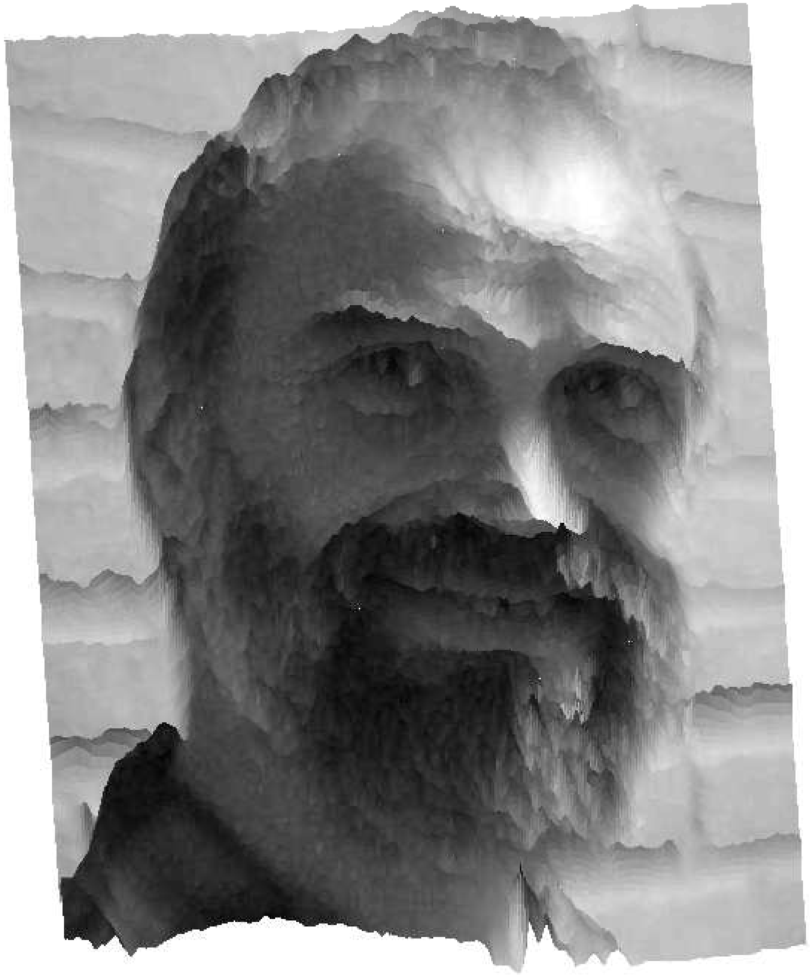
Matrices can be viewed as functions on a (discrete) rectangular domain: $(i,j) \rightsquigarrow a_{ij}$.



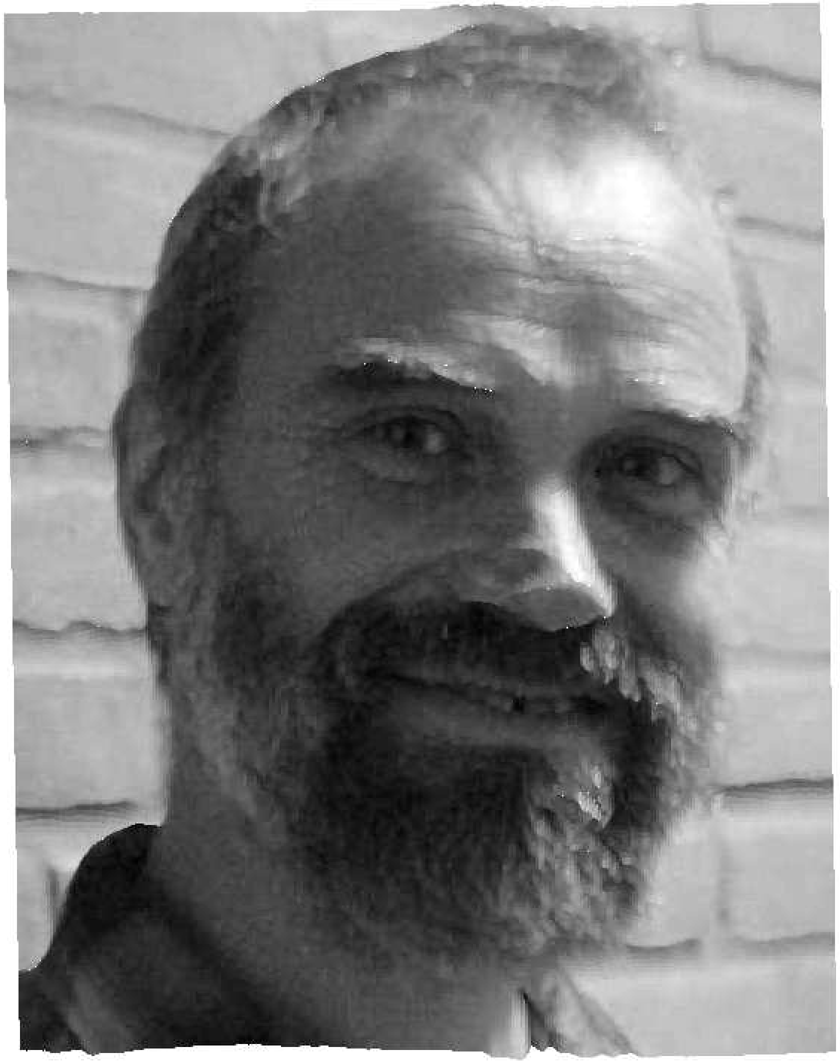
Matrices can be viewed as functions on a (discrete) rectangular domain: $(i, j) \rightsquigarrow a_{ij}$.



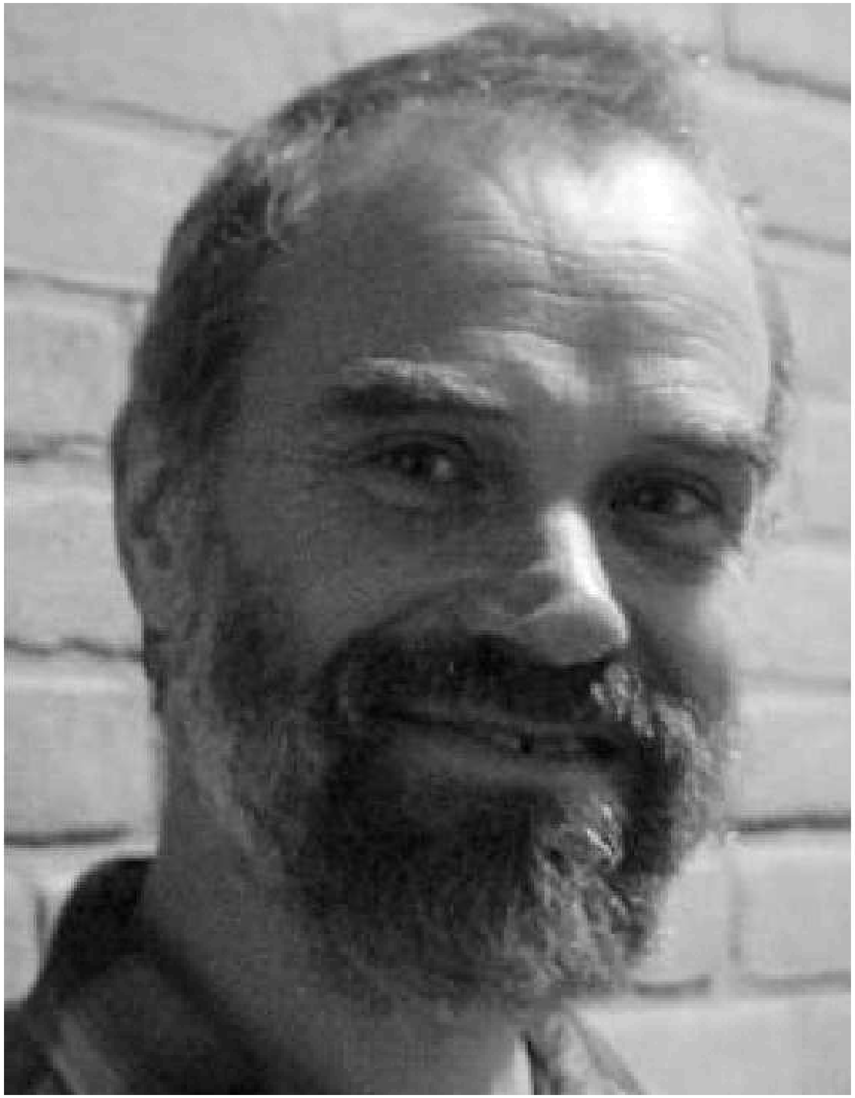
Matrices can be viewed as functions on a (discrete) rectangular domain: $(i, j) \rightsquigarrow a_{ij}$.



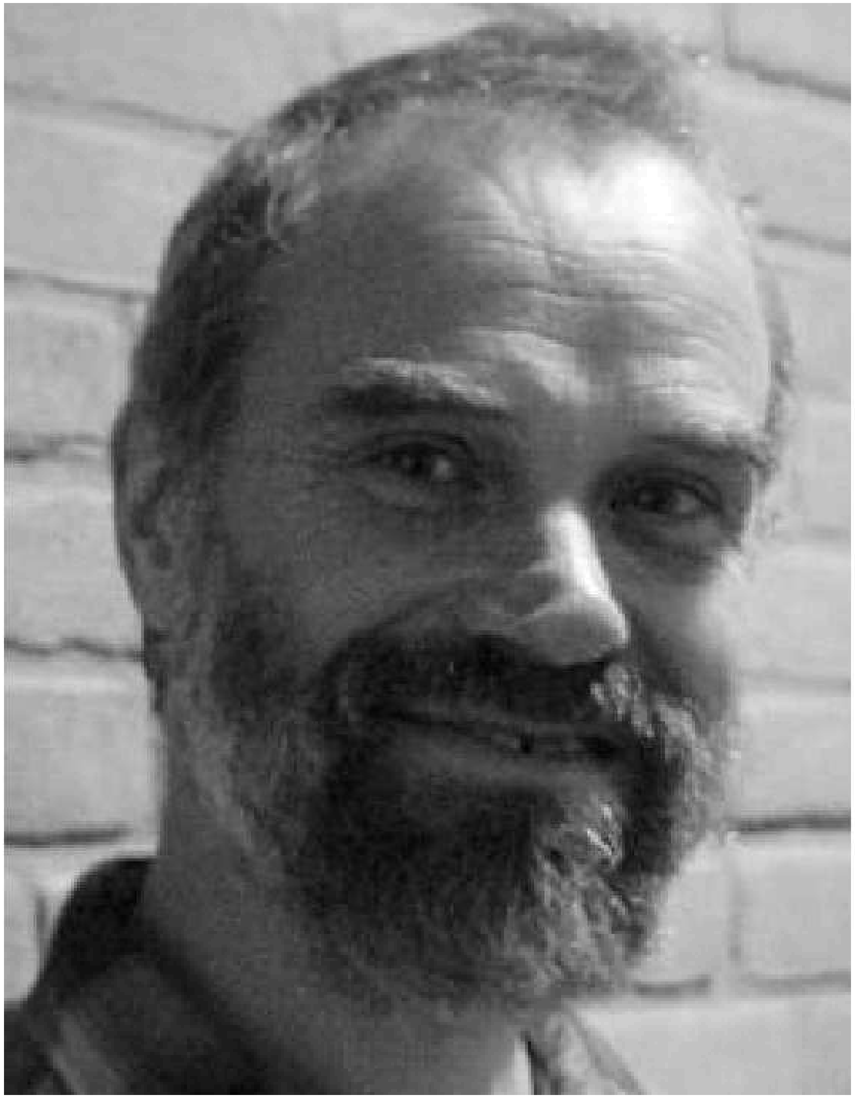
Matrices can be viewed as functions on a (discrete) rectangular domain: $(i, j) \rightsquigarrow a_{ij}$.



Matrices can be viewed as functions on a (discrete) rectangular domain: $(i, j) \rightsquigarrow a_{ij}$.



Matrices can be viewed as pictures (in gray scale colors).



Partial SVD

$$\mathbf{A} = \mathbf{U}_r\, \Sigma_r\, \mathbf{V}_r^* = \sum_{i=1}^{r} \sigma_i\, \mathbf{u}_i \mathbf{v}_i^*$$

The following pictures show for some matrix $\mathbf{A}$,

for $j = 2, 3, \ldots, r$

- at the left the matrix

$$\sum_{i=1}^{j-1} \sigma_i\, \mathbf{u}_i \mathbf{v}_i^*$$

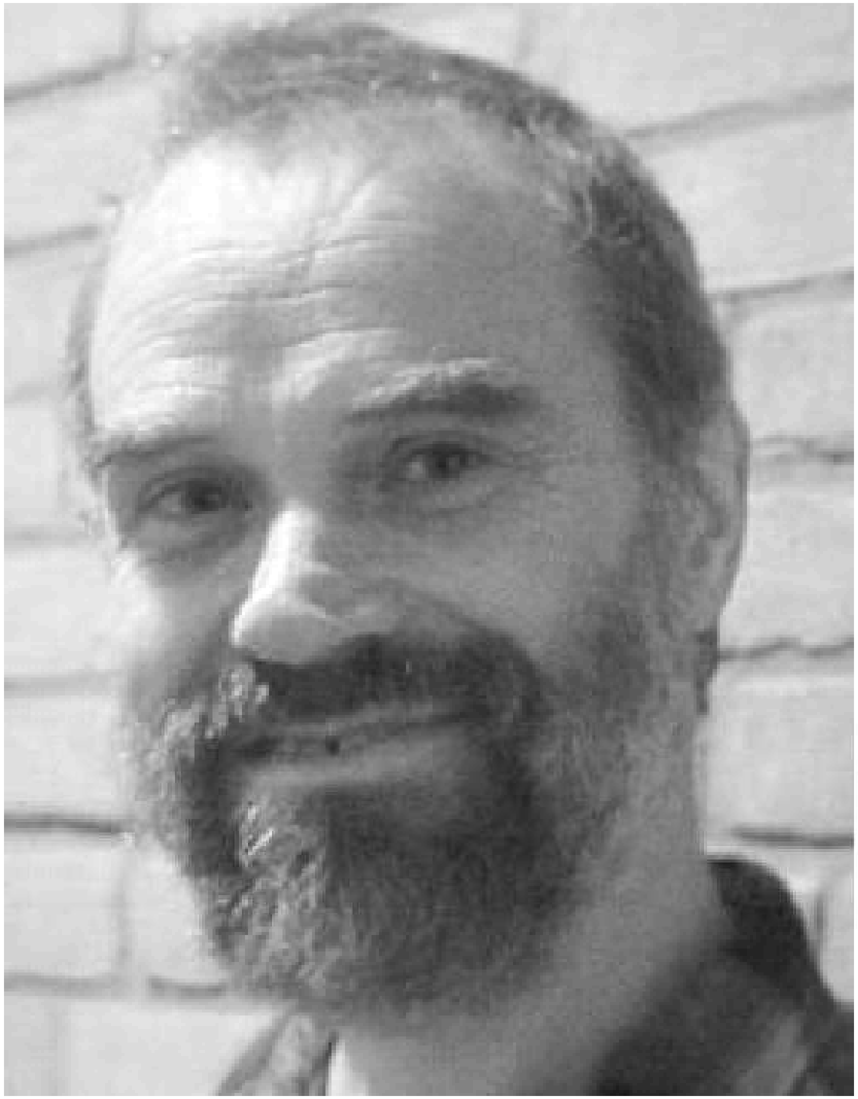
from the partial SVD and

- at the right the rank 1 matrix

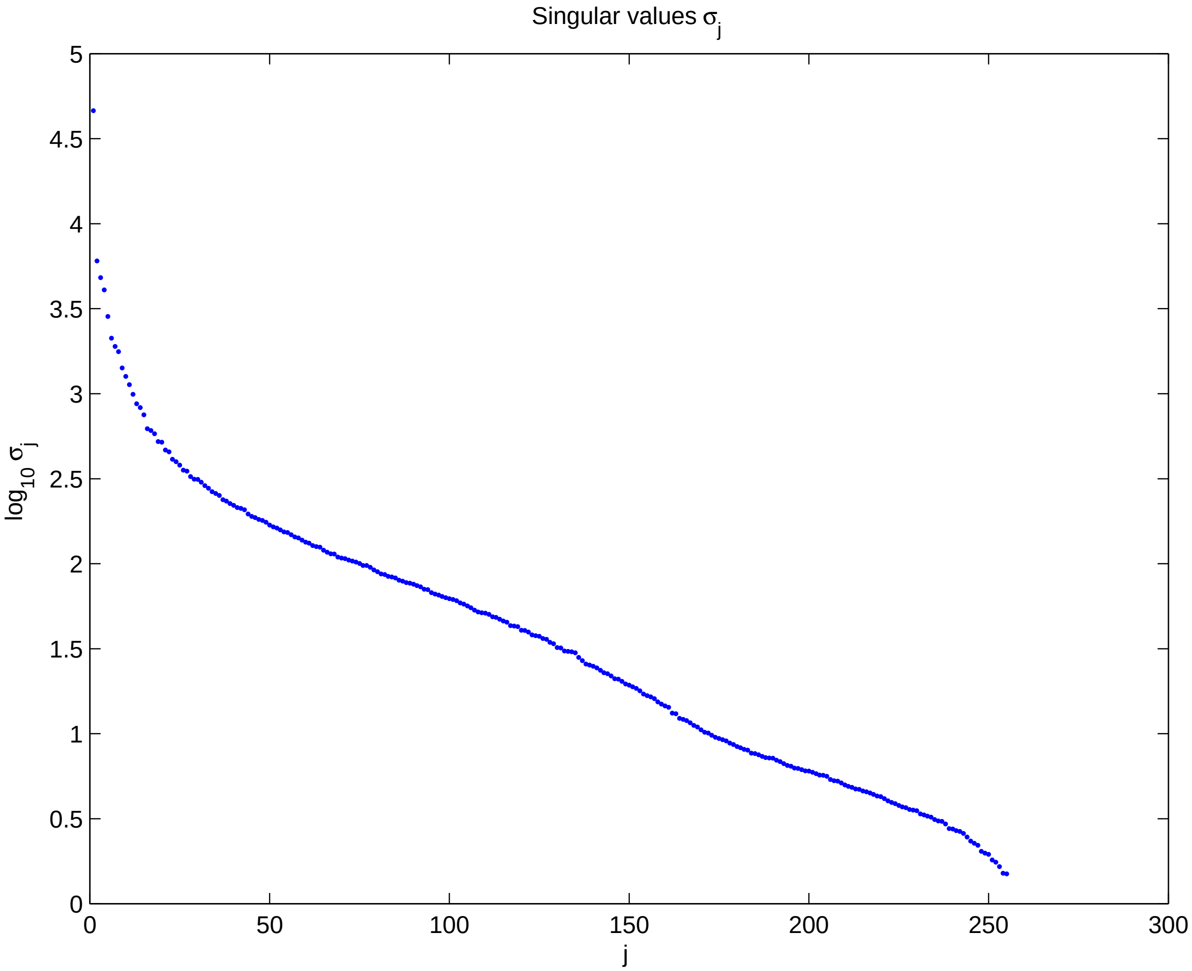
$$\sigma_j\, \mathbf{u}_j \mathbf{v}_j^*.$$

The matrix $\mathbf{A}$.

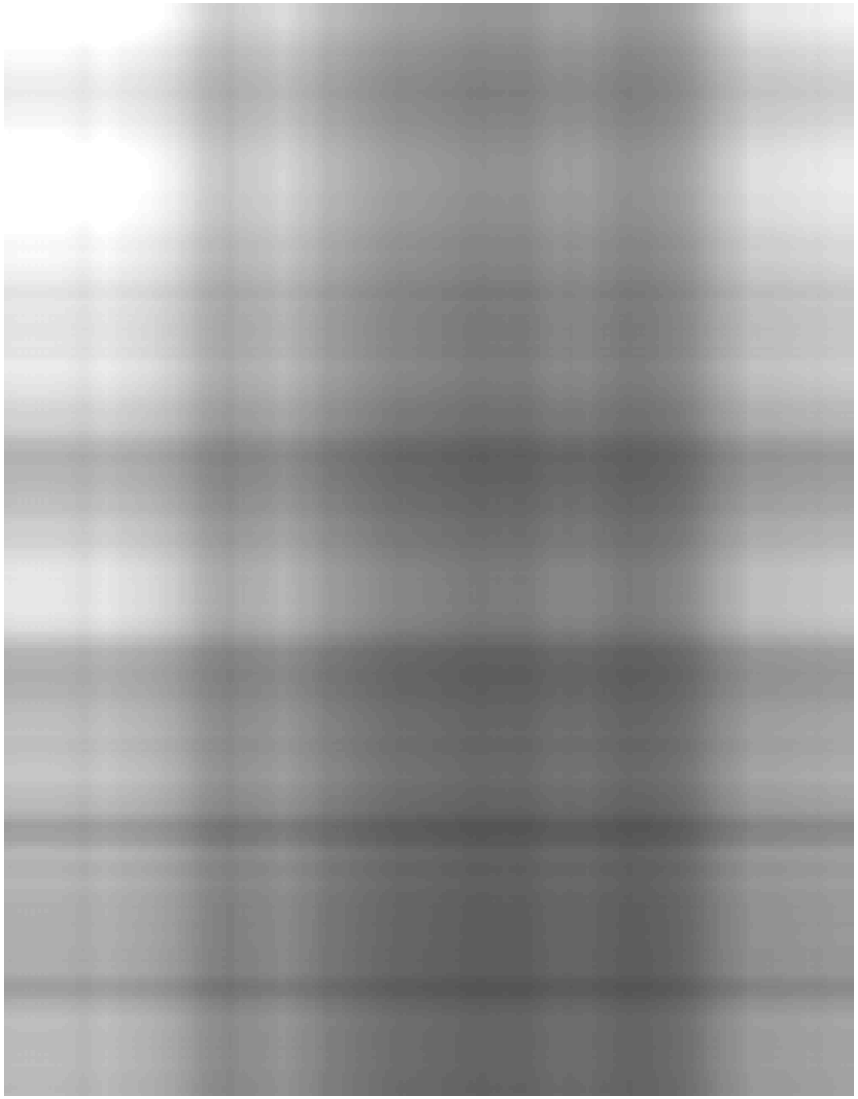
Original picture



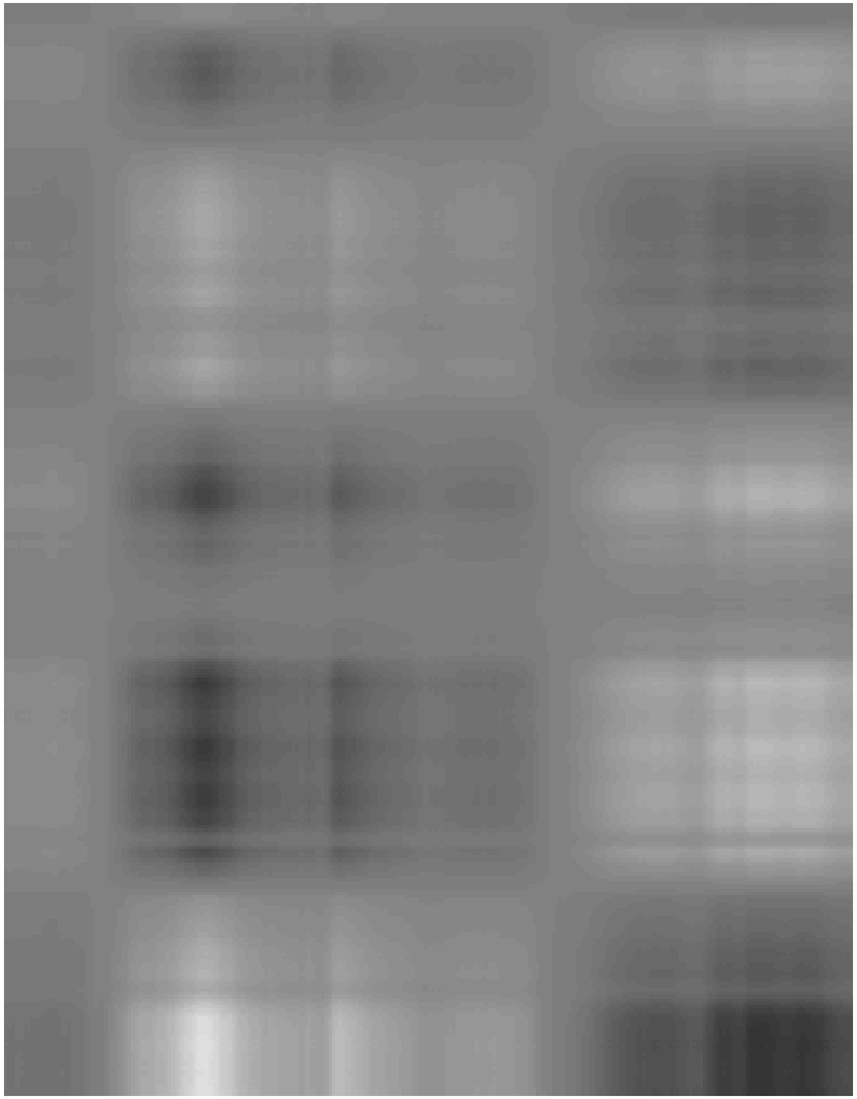
Singular values of $\mathbf{A}$.



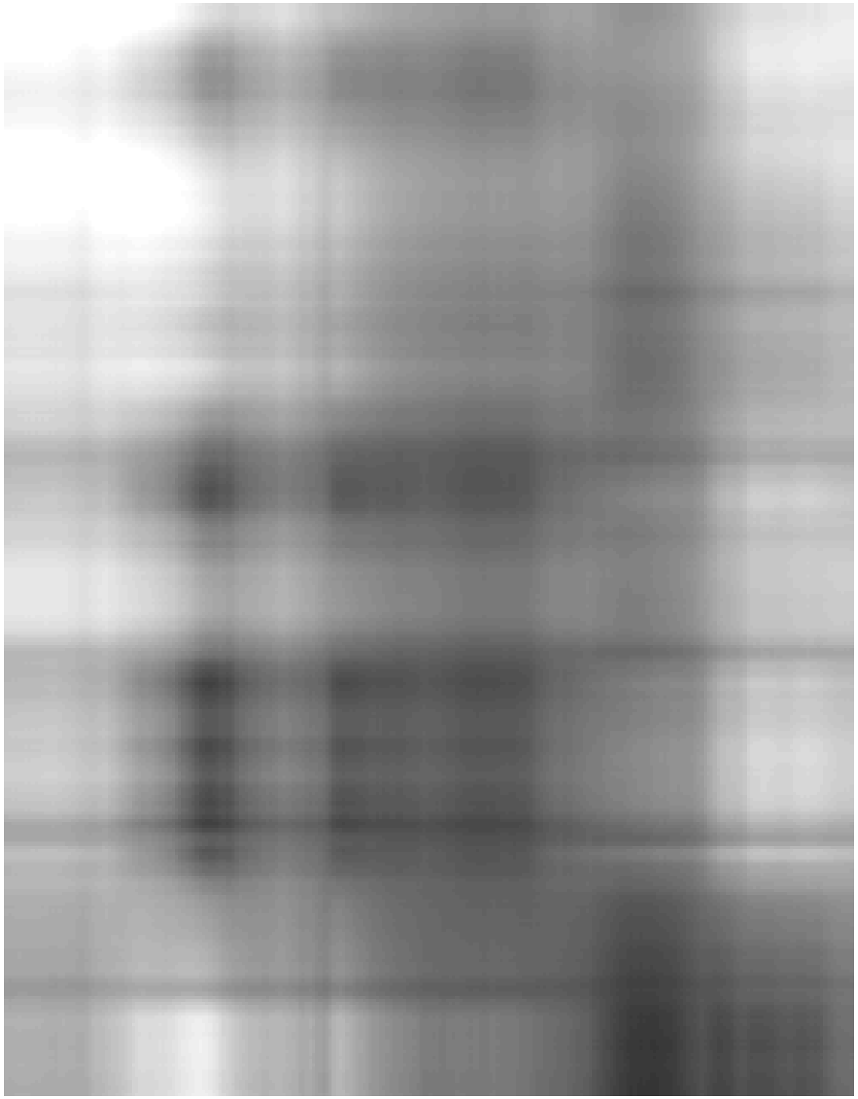
partial svd based on 1 singular values



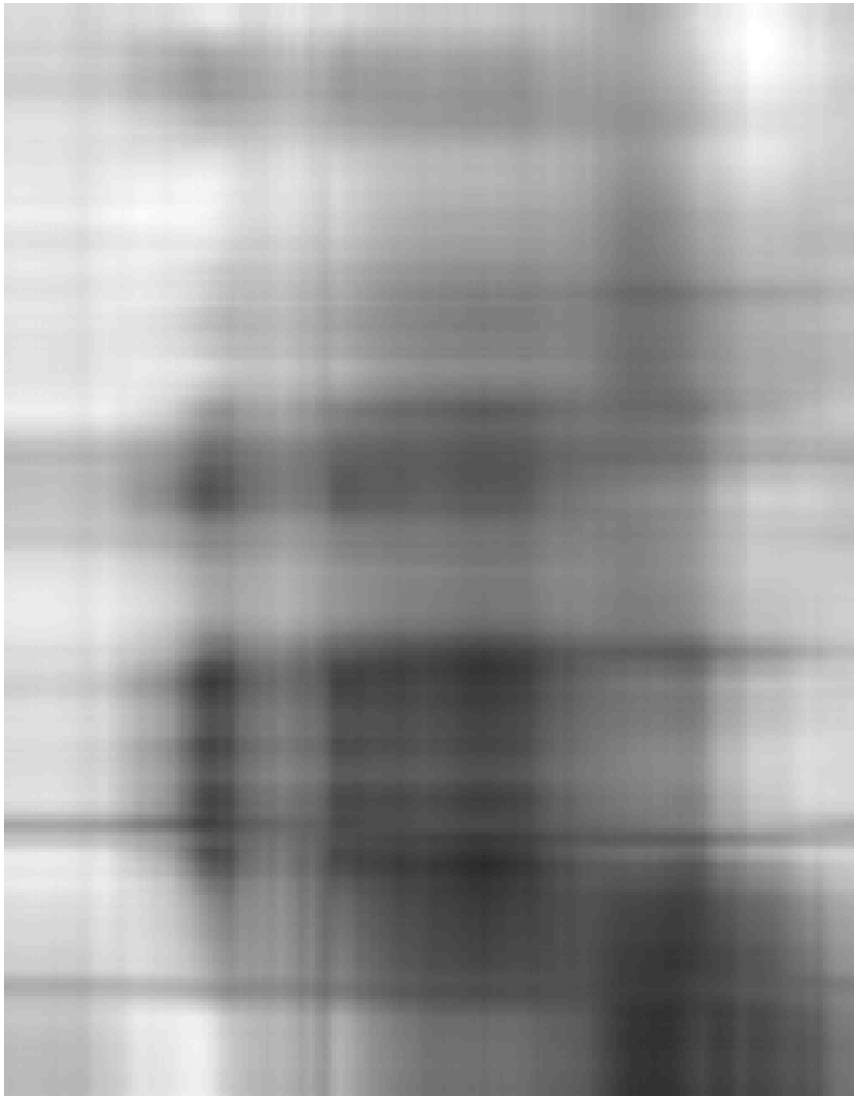
$c_j\, u_j\, v_j^*$ for $j = 2$



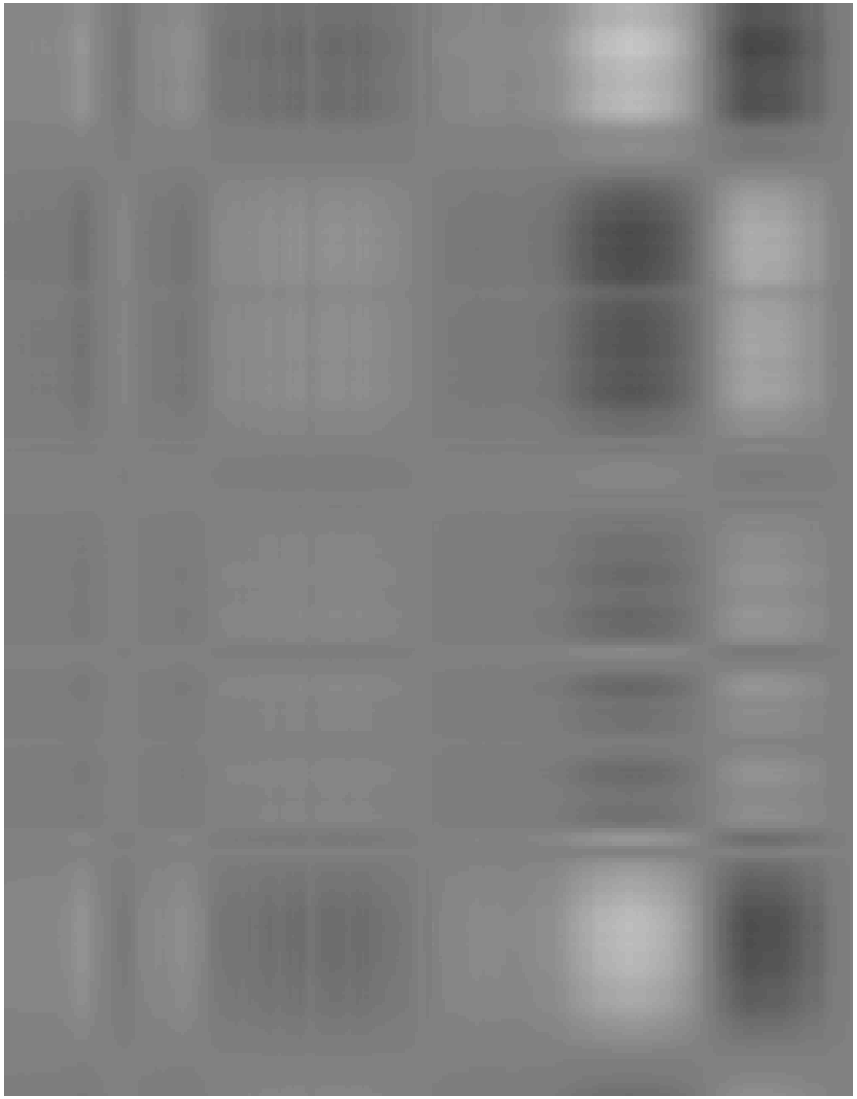
partial svd based on 2 singular values



$c_j \, u_j \, v_j^*$ for $j = 3$

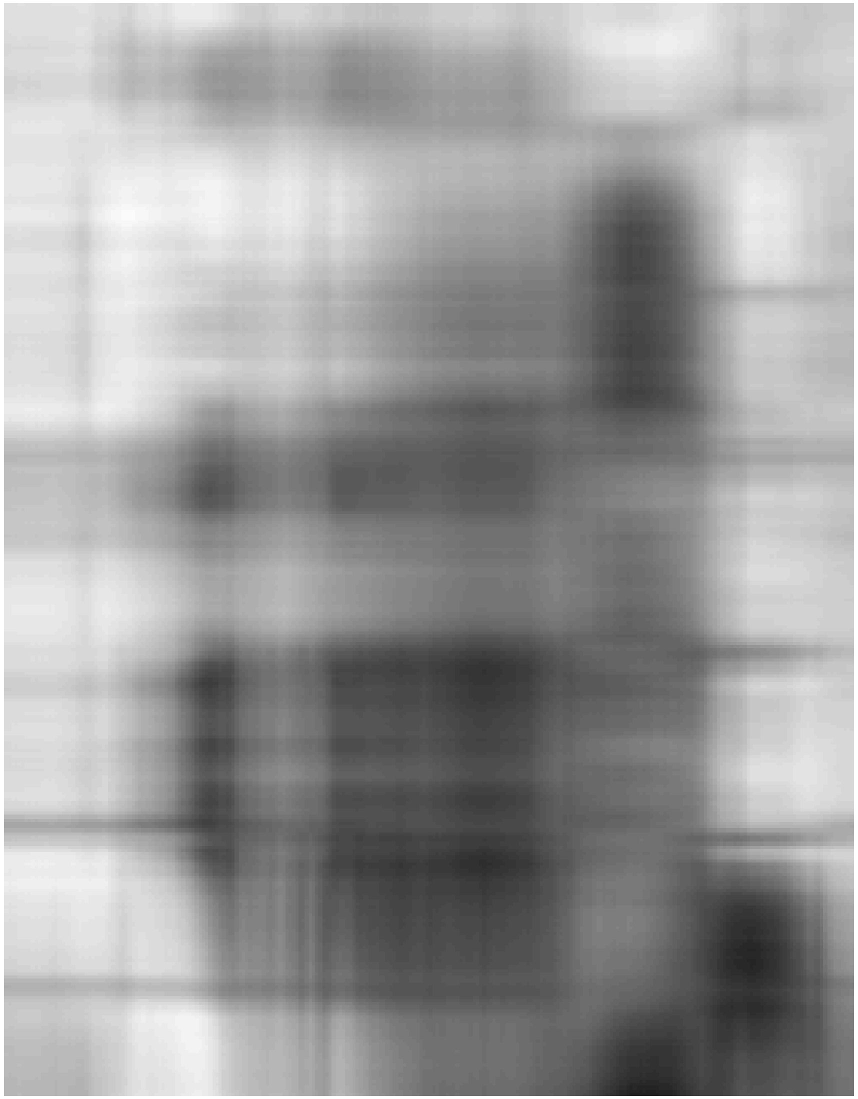


partial svd based on 3 singular values

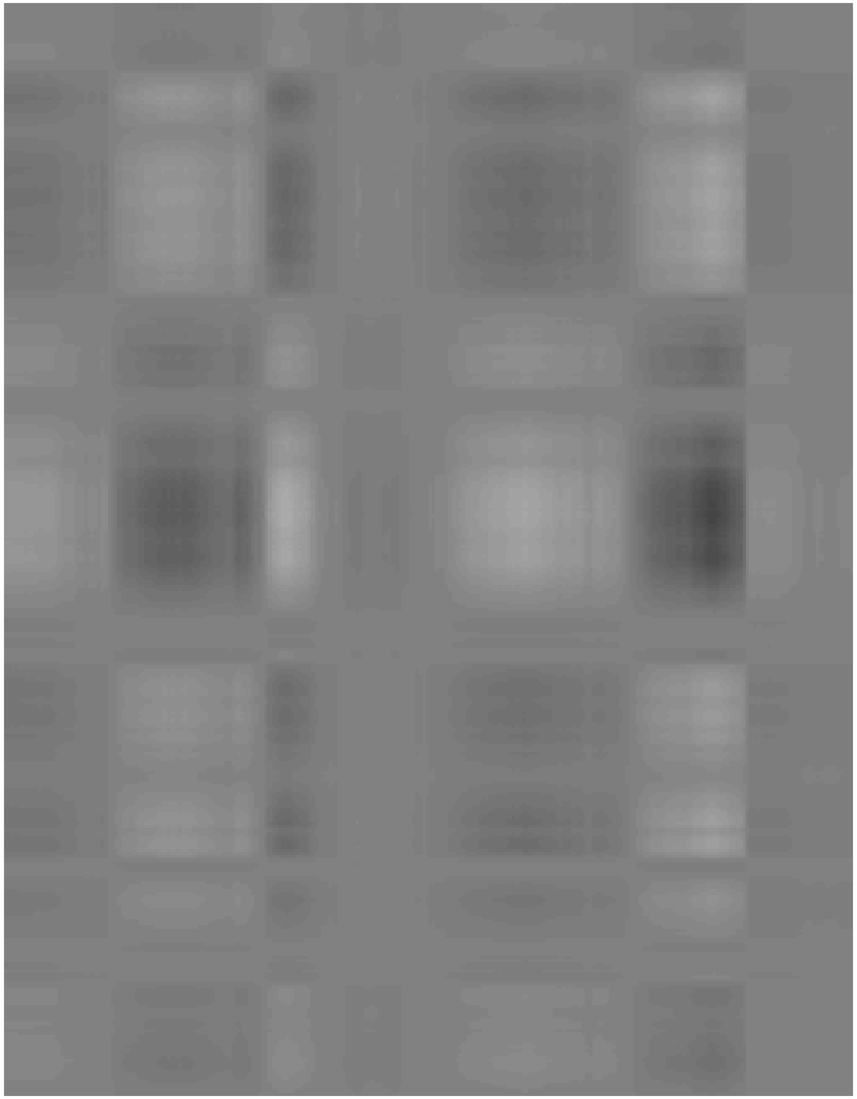


$c_j \, u_j \, v_j^*$ for $j = 4$

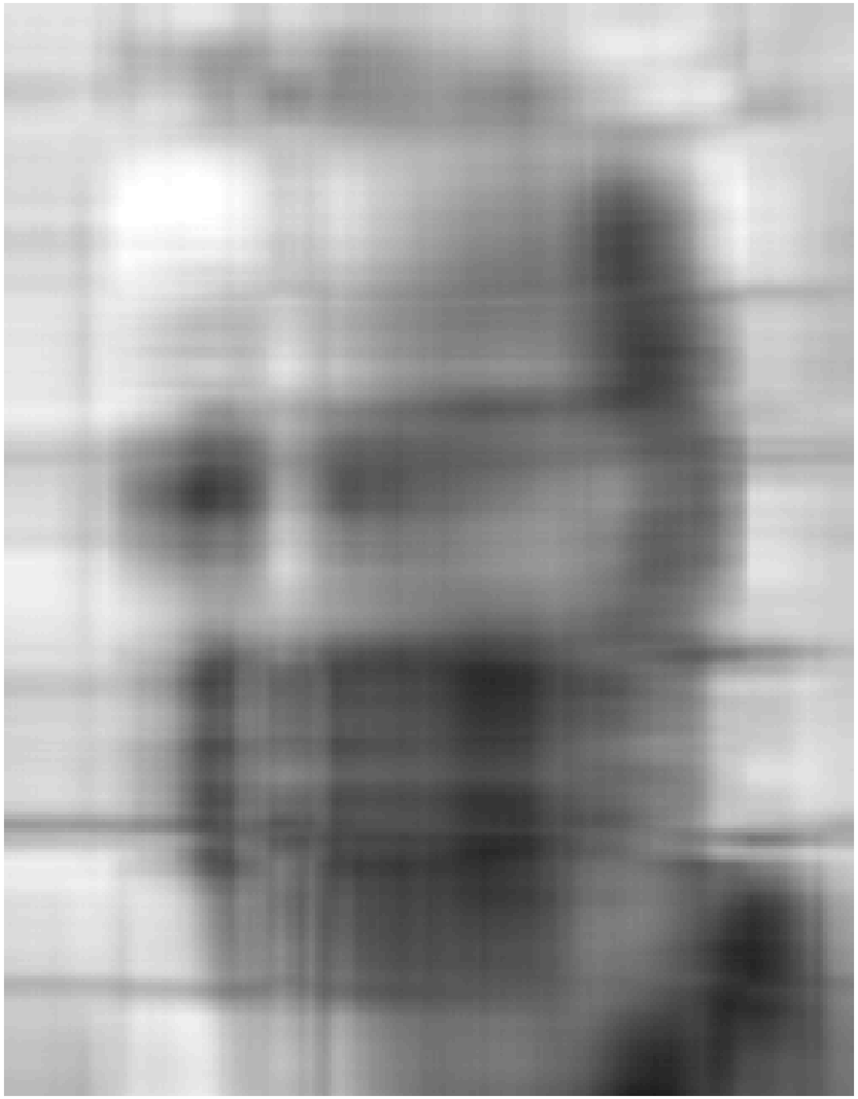
partial svd based on 4 singular values



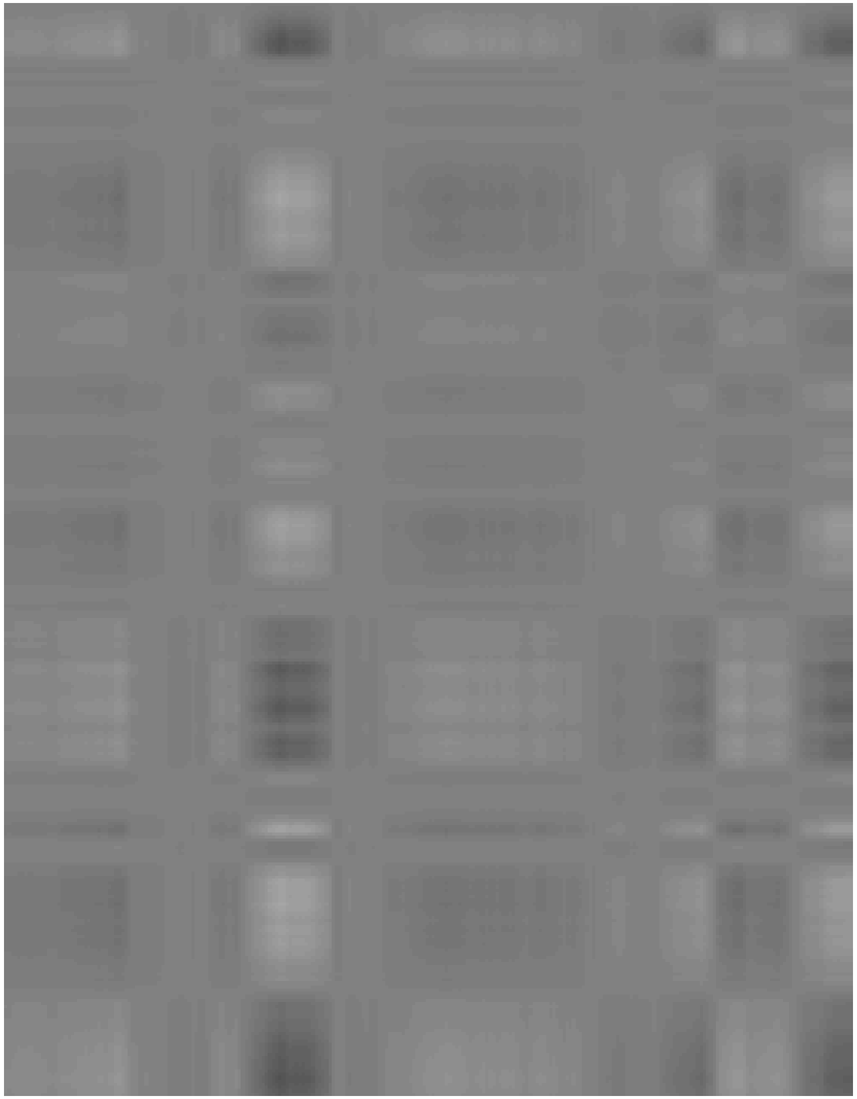
$c_j\, u_j\, v_j^*$ for $j = 5$



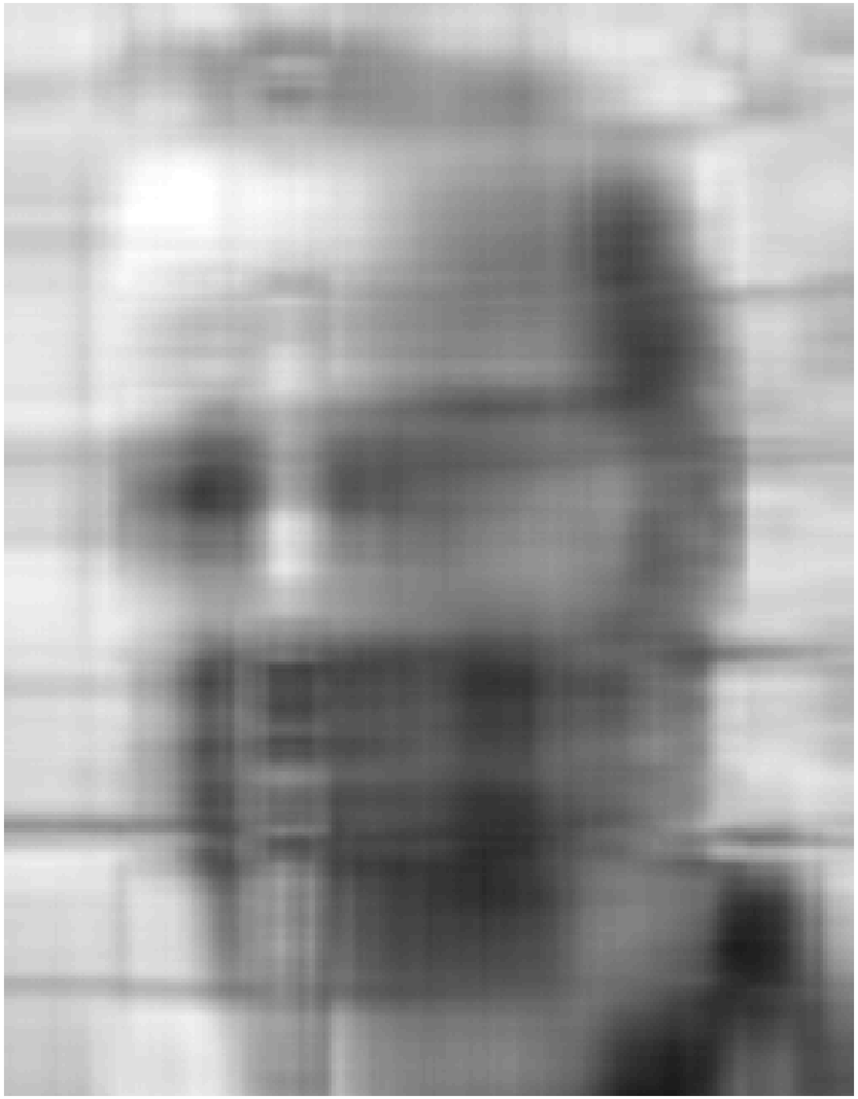
partial svd based on 5 singular values



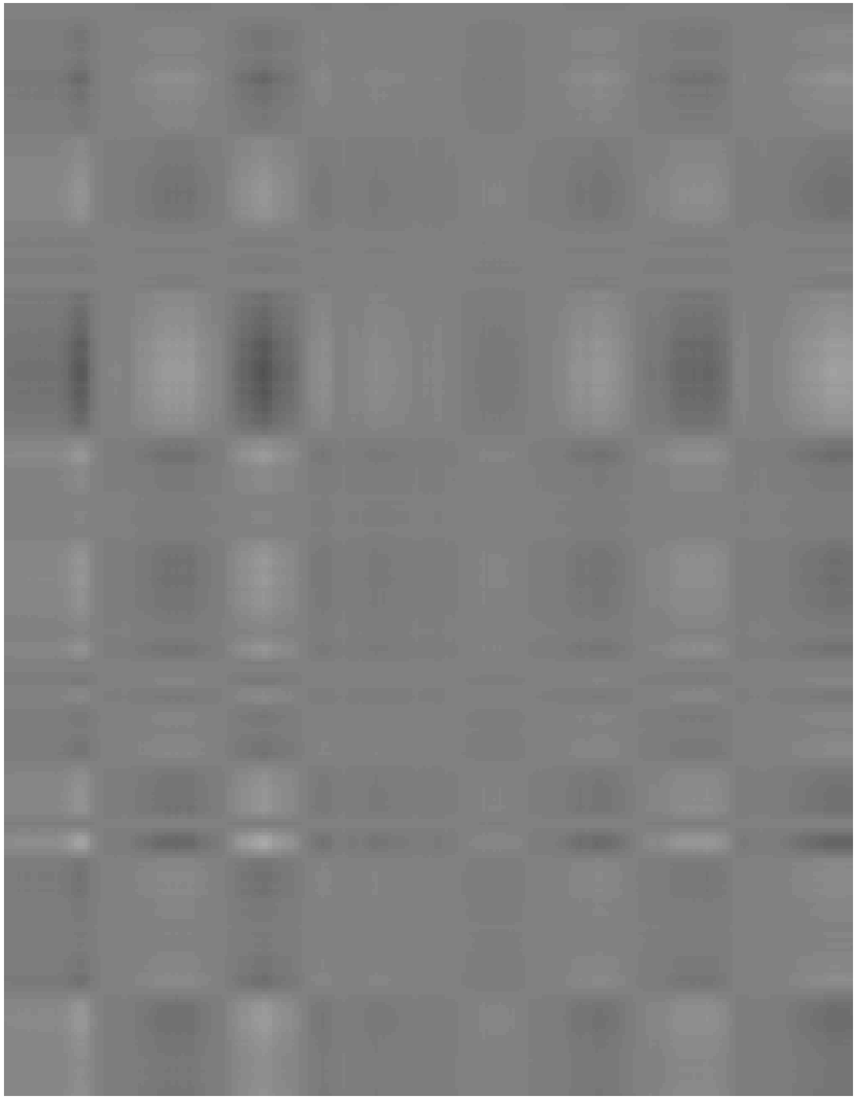
$c_j u_j v_j^*$ for $j = 6$



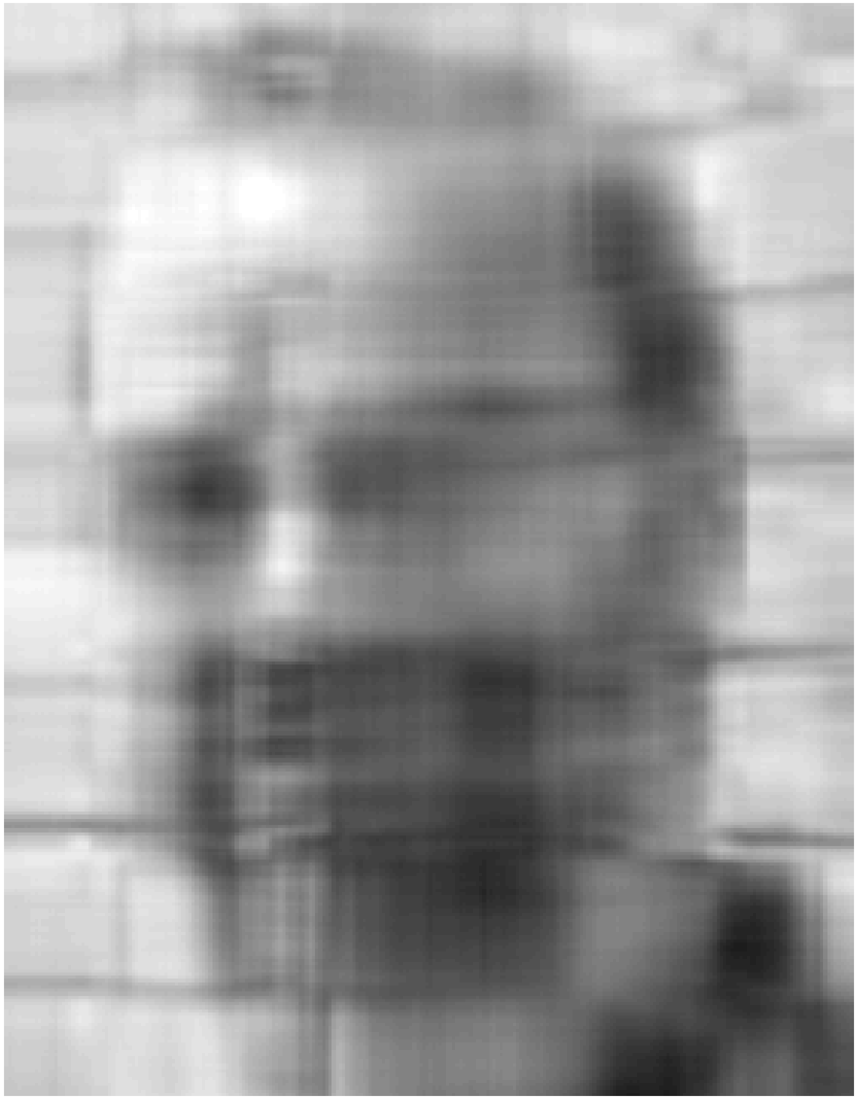
partial svd based on 6 singular values



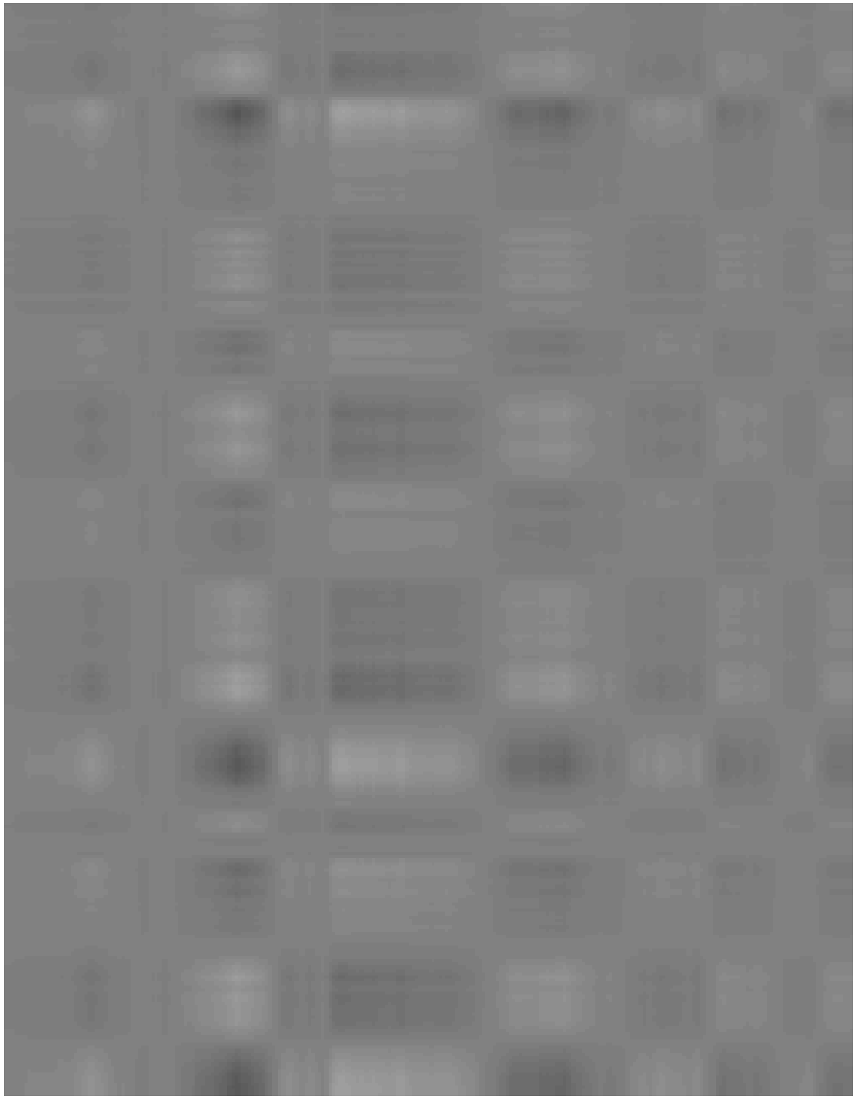
$c_j \, u_j \, v_j^*$ for $j = 7$



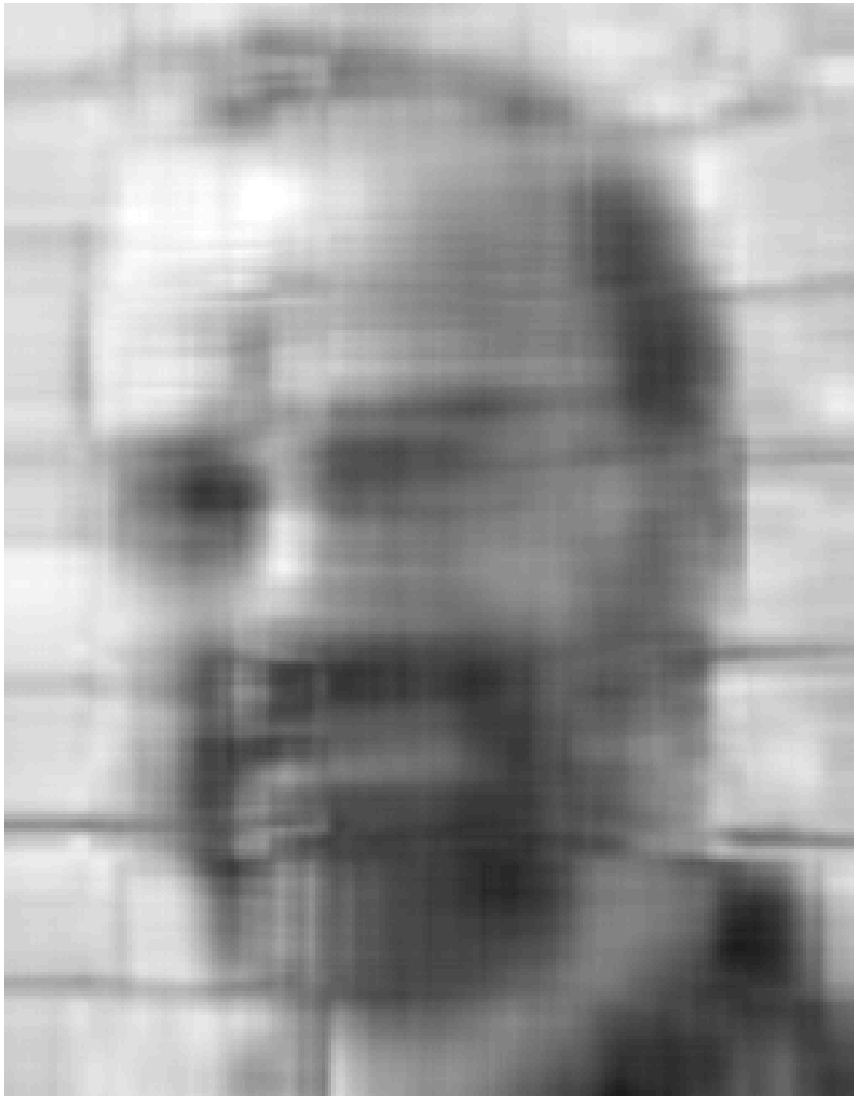
partial svd based on 7 singular values



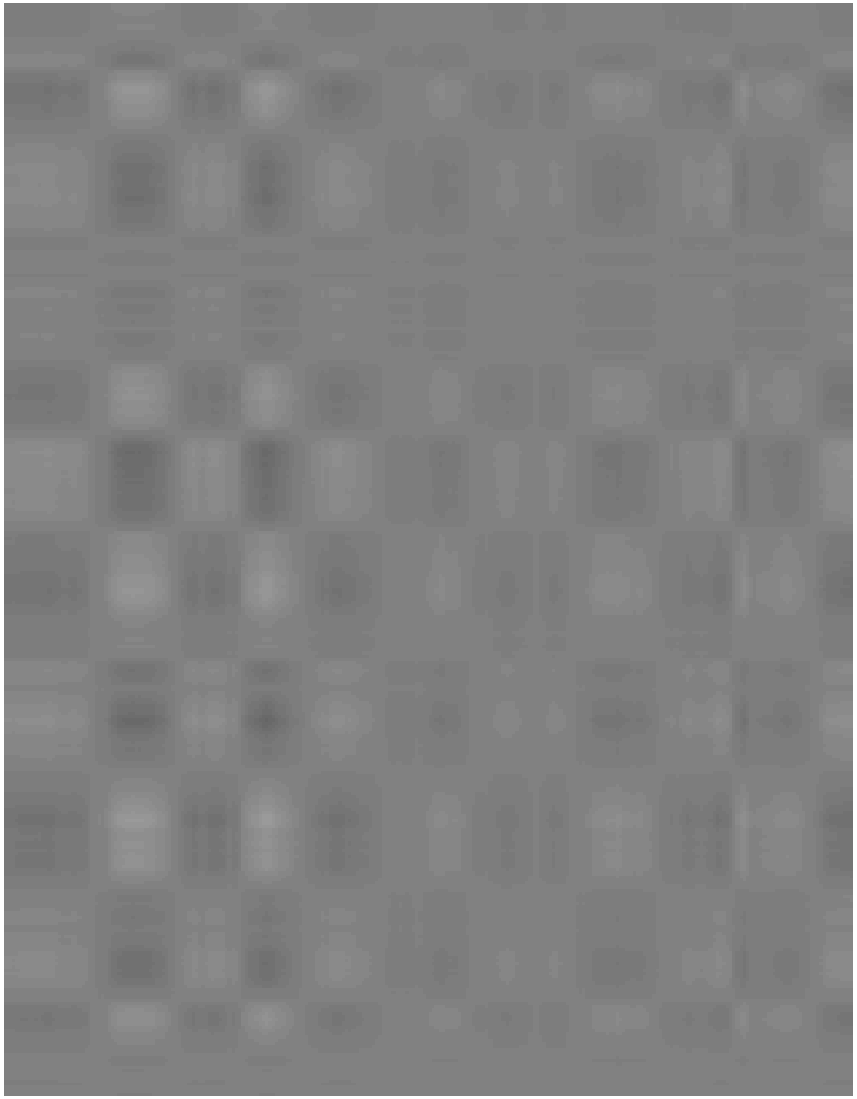
$c_j\, u_j\, v_j^*$ for $j = 8$

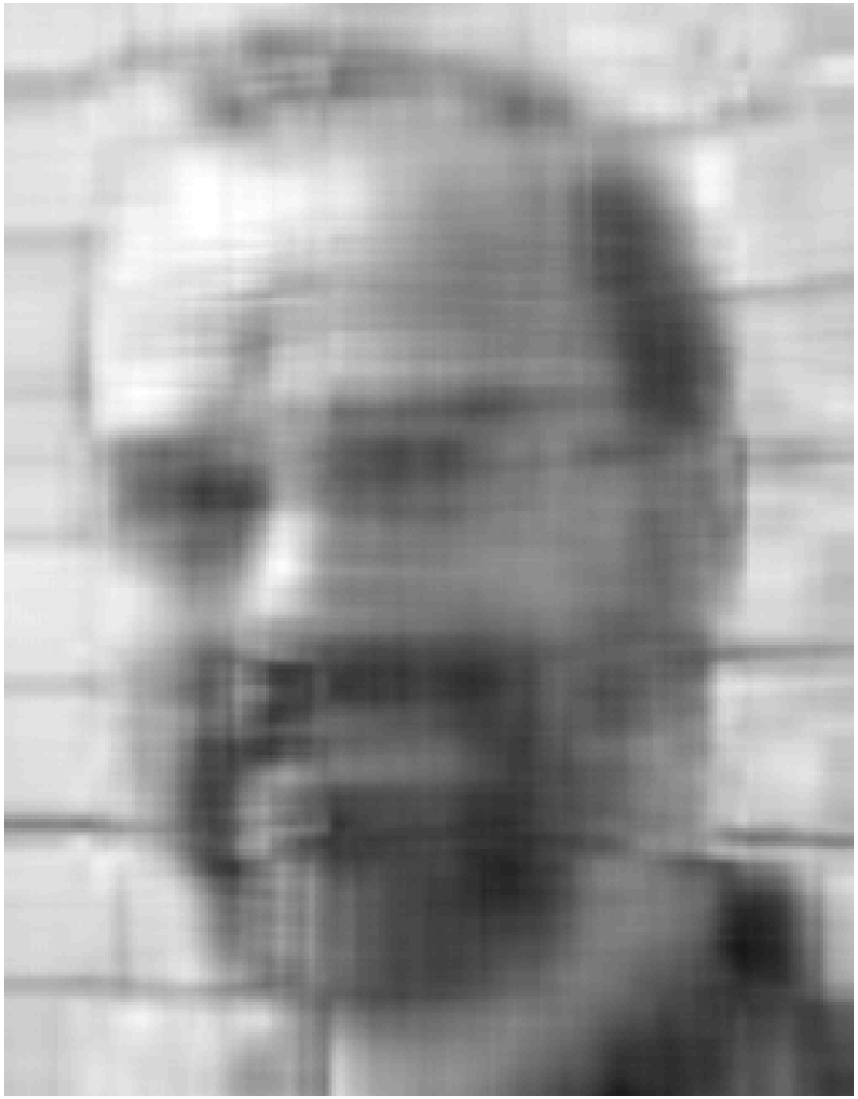


partial svd based on 8 singular values

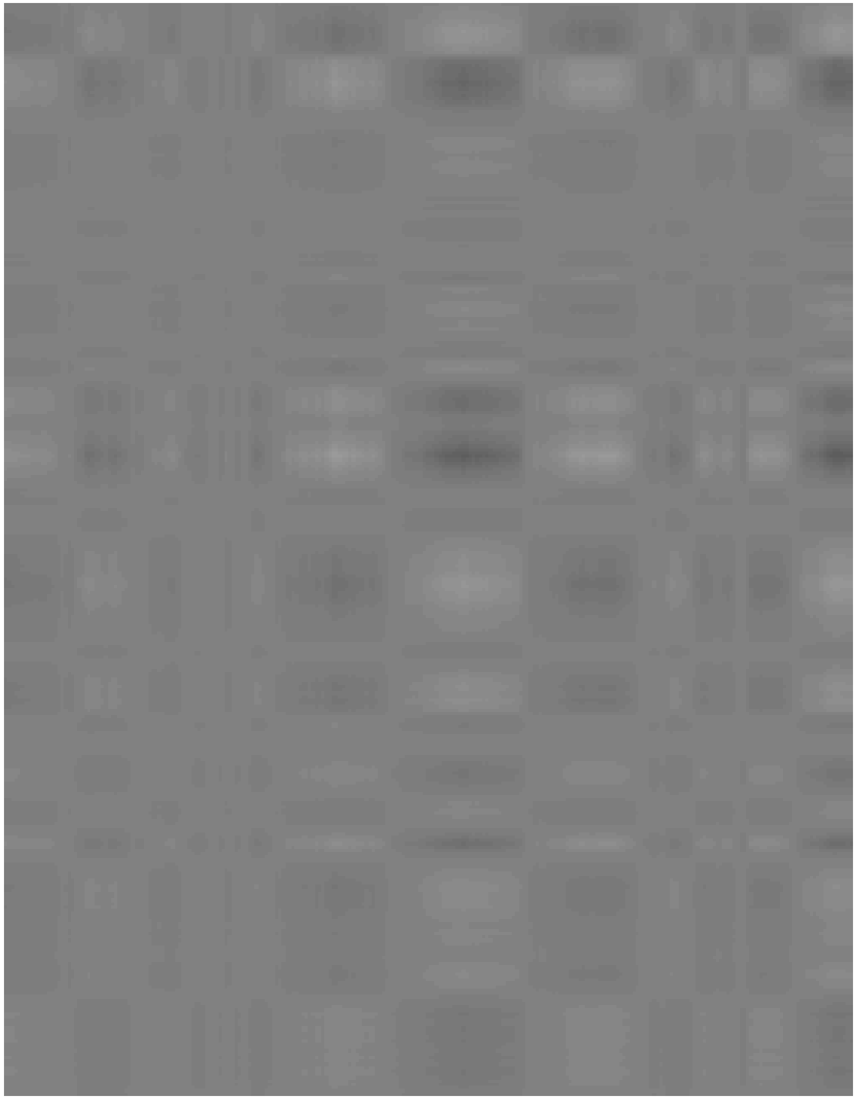


$c_j \, u_j \, v_j^*$ for j = 9



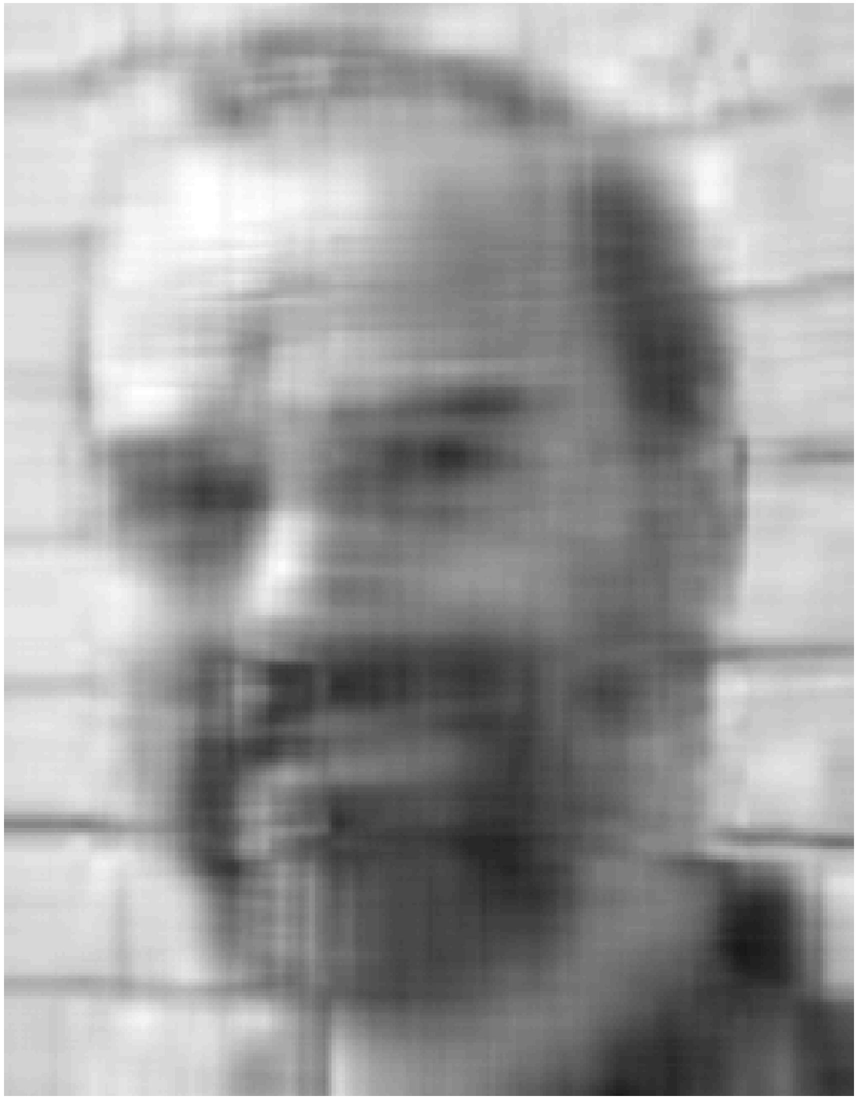


partial svd based on 9 singular values

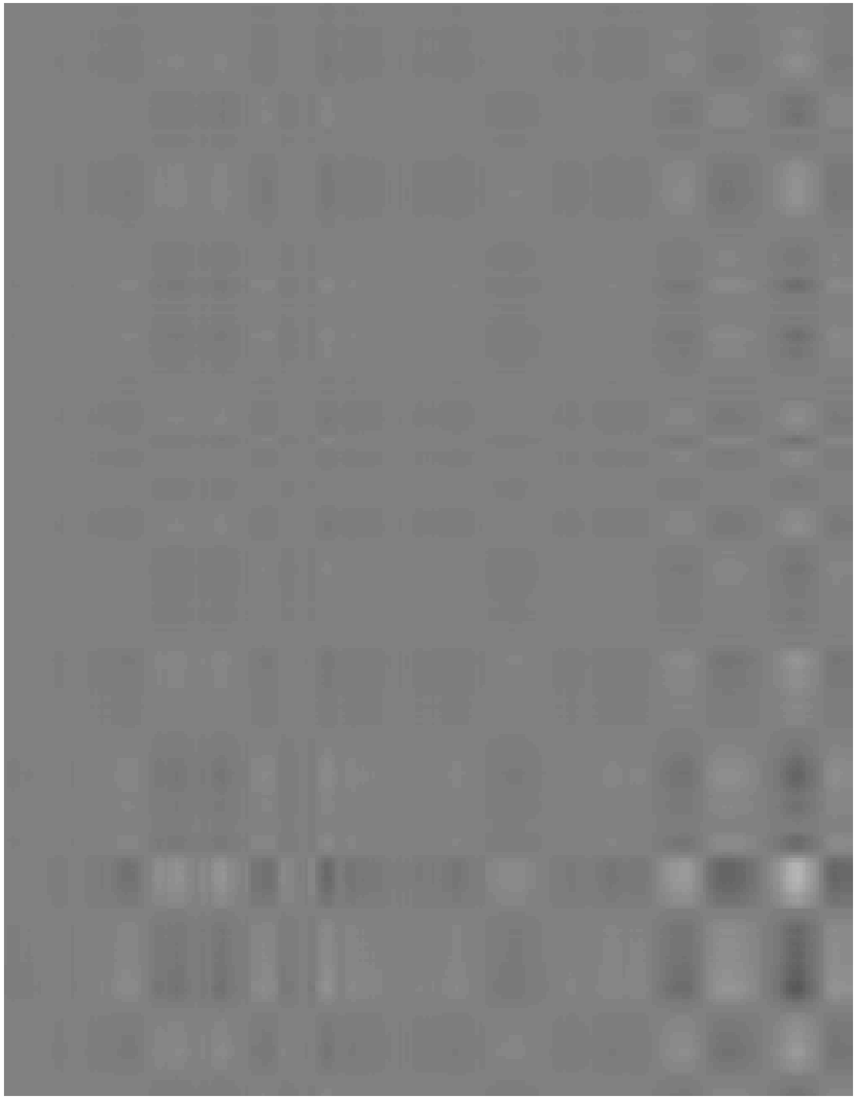


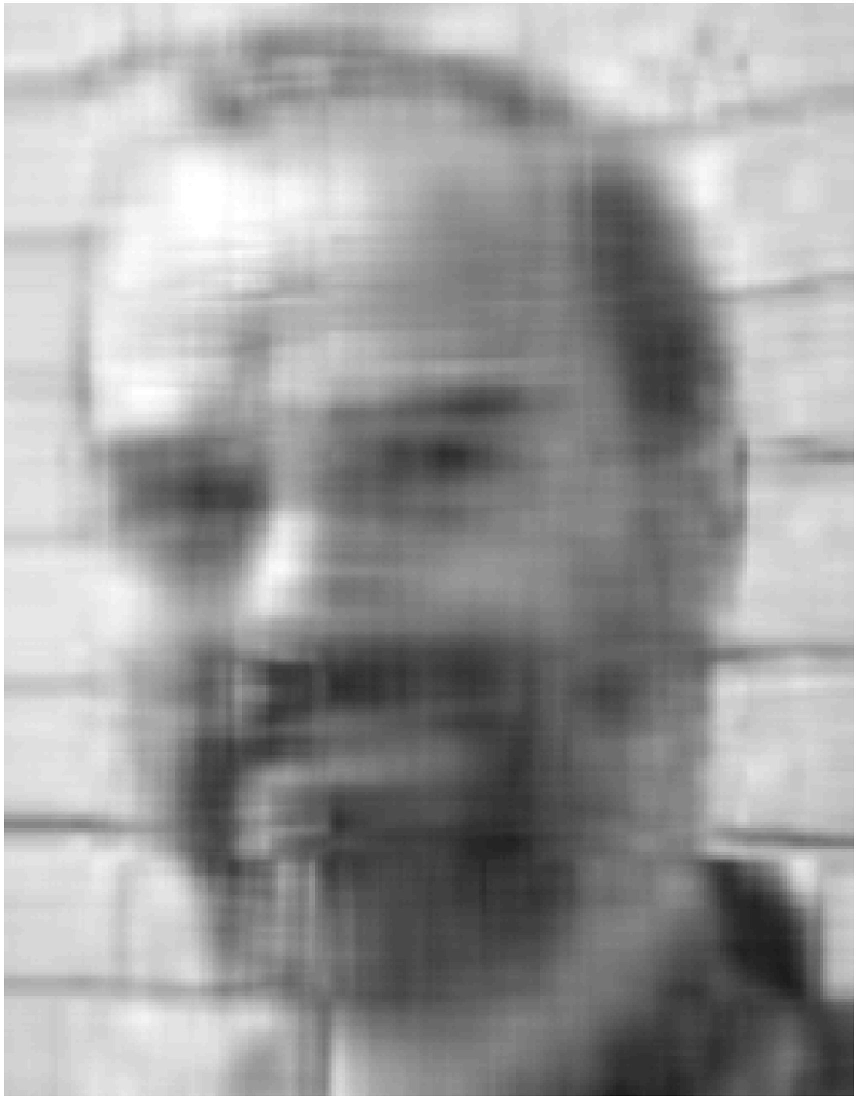
$c_j \, u_j \, v_j^*$ for $j = 10$

partial svd based on 10 singular values

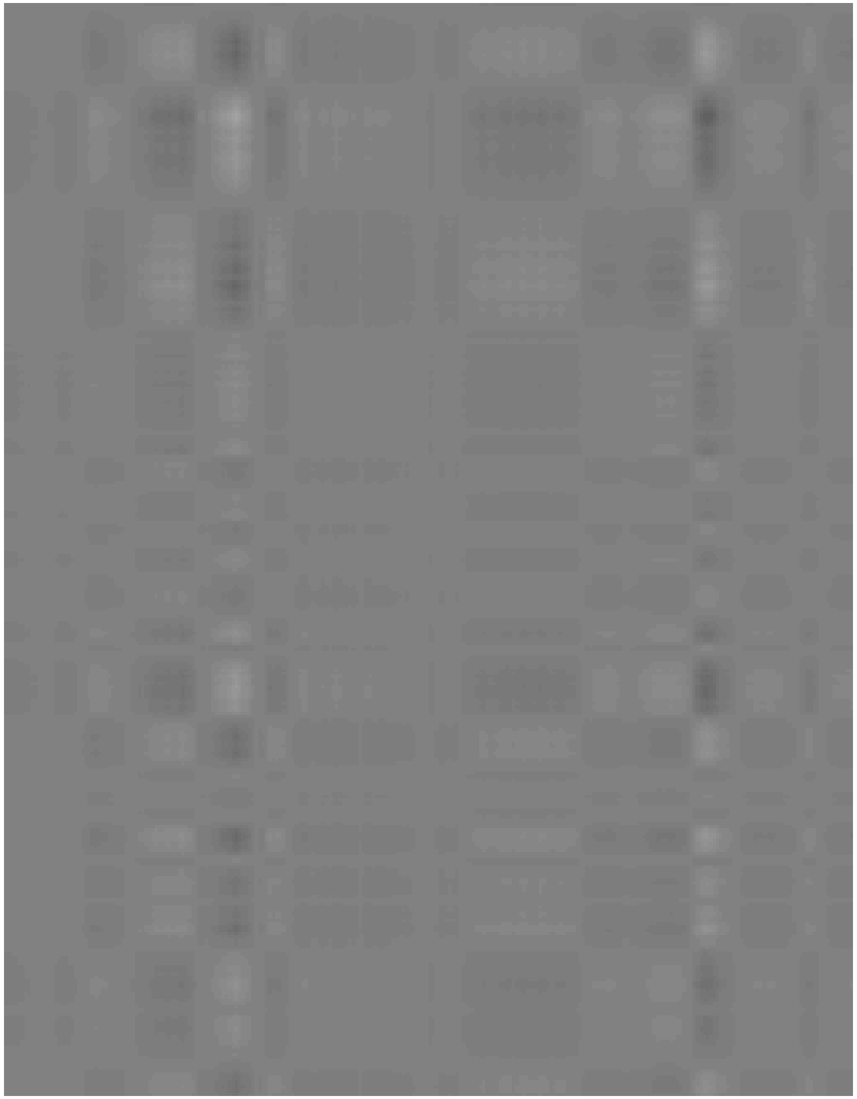


$c_j \, u_j \, v_j^*$ for $j = 11$



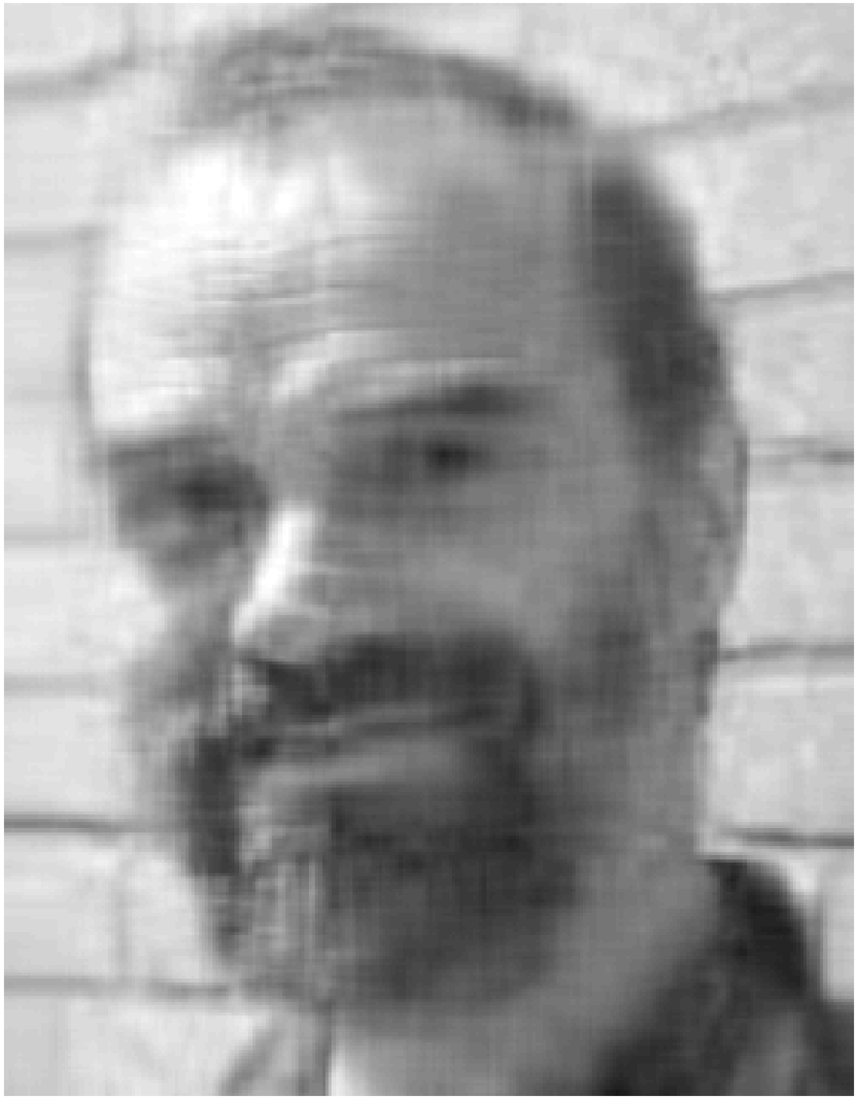


partial svd based on 11 singular values

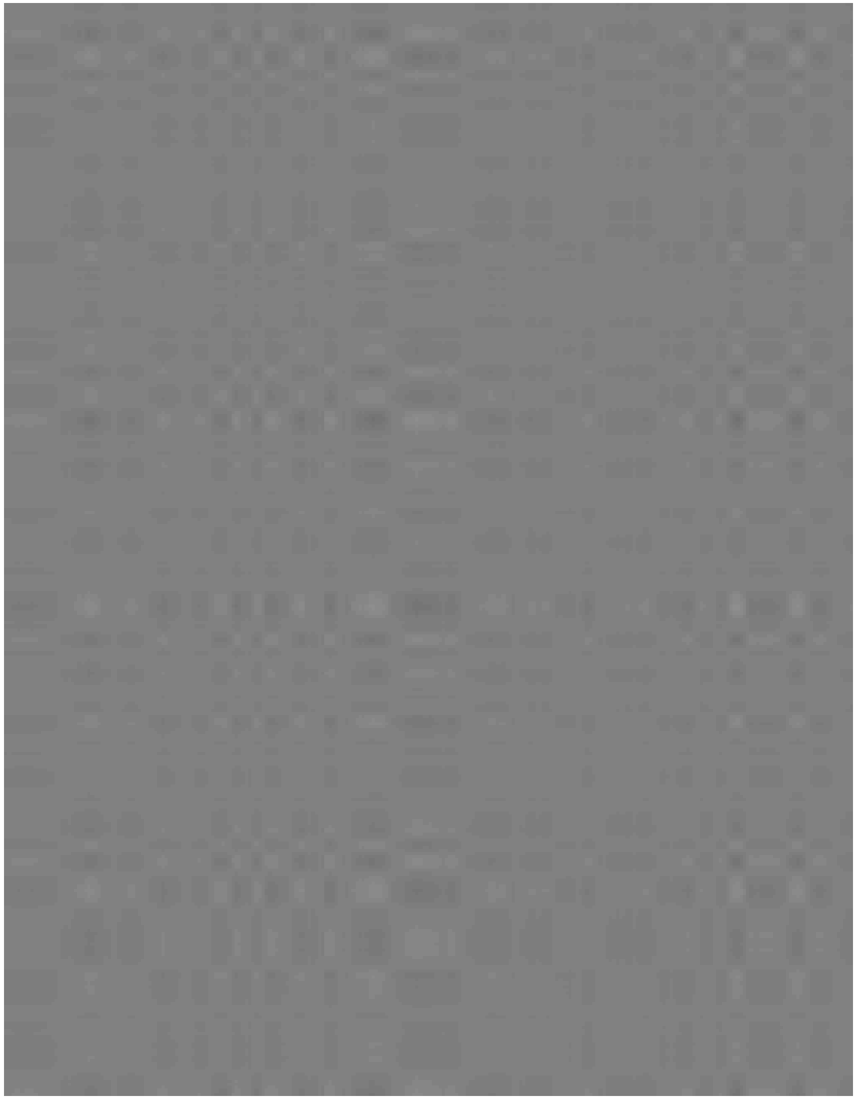


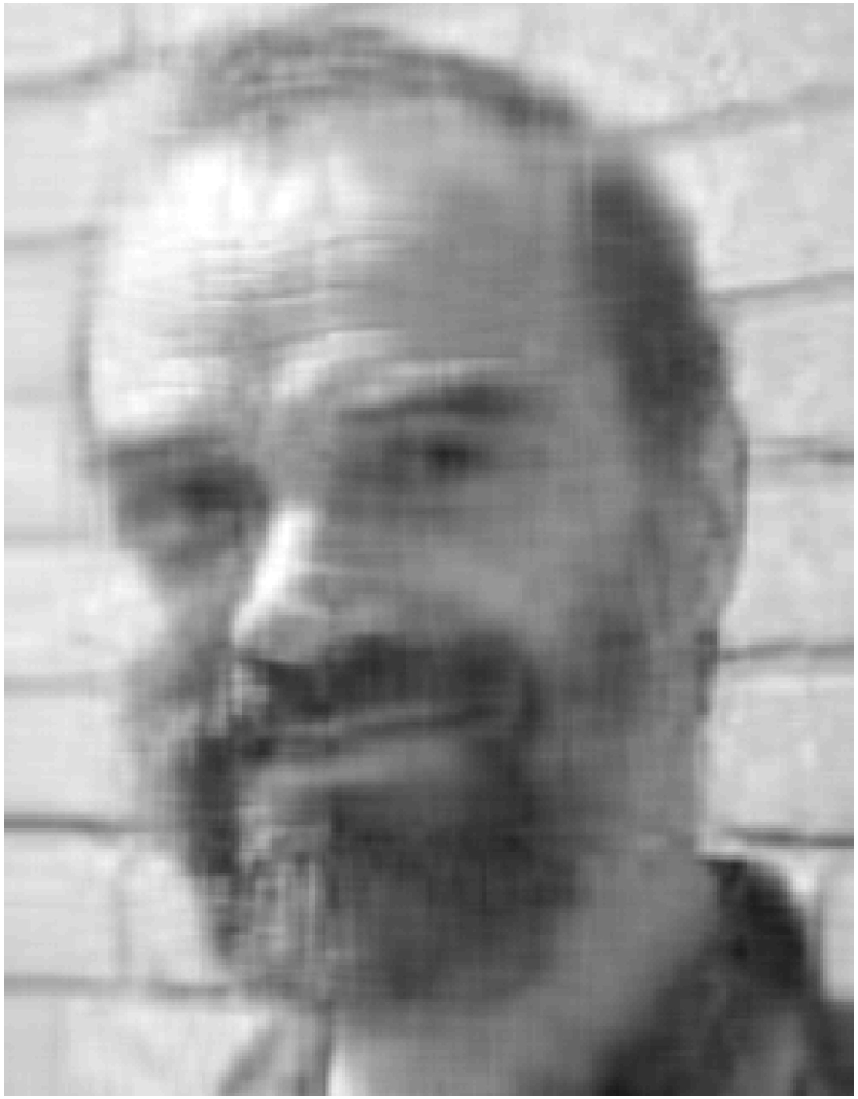
$c_j\, u_j\, v_j^*$ for $j = 12$

partial svd based on 18 singular values

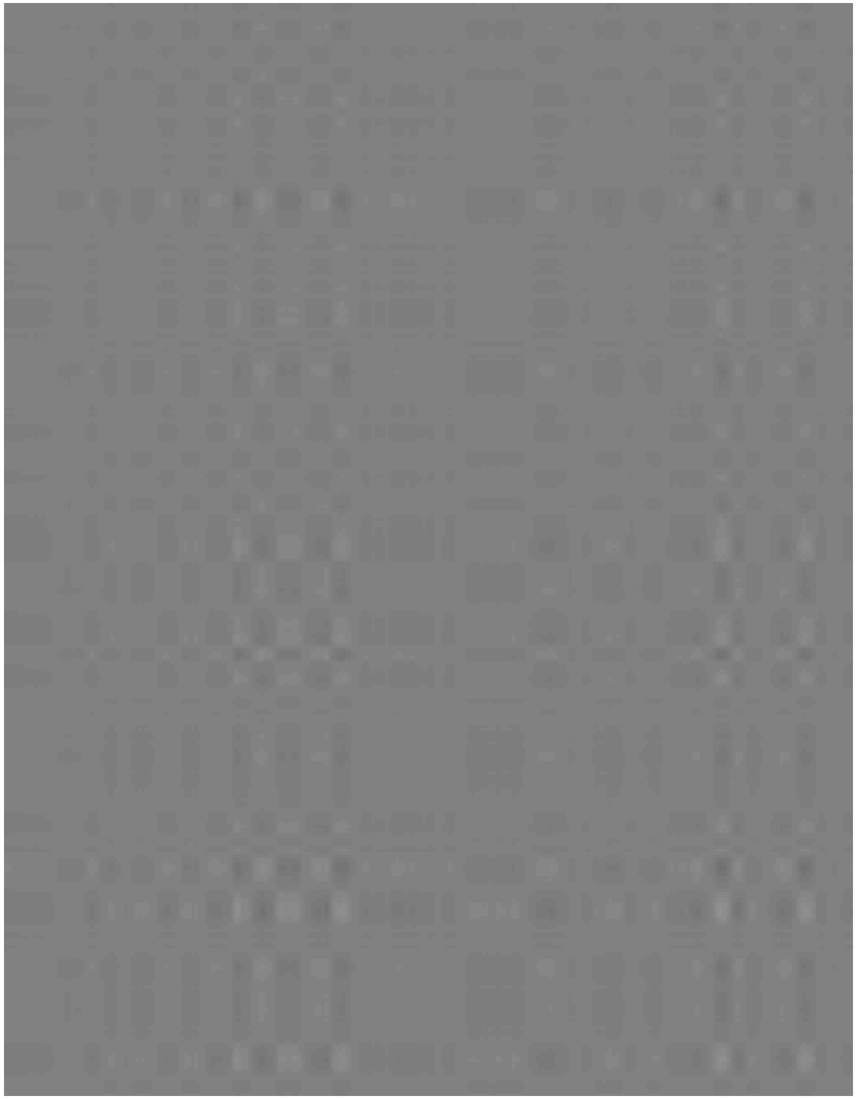


$c_j\, u_j\, v_j^*$ for $j = 19$



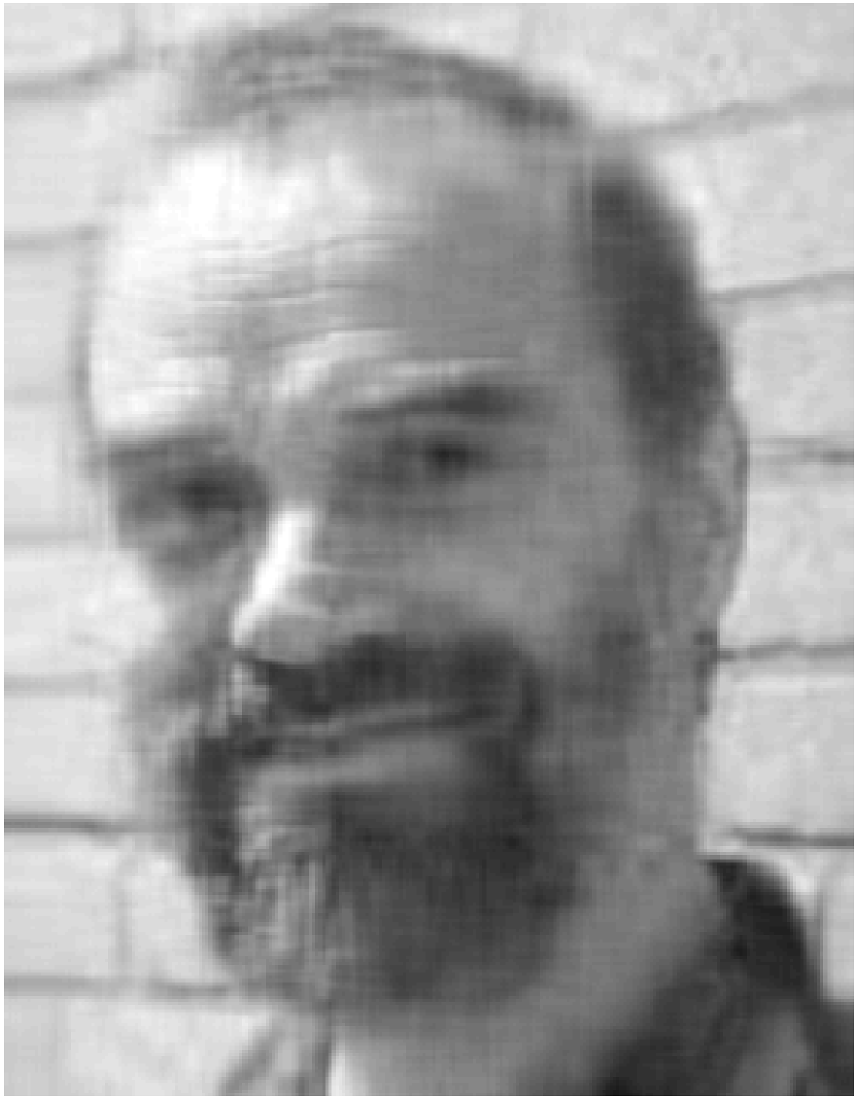


partial svd based on 19 singular values

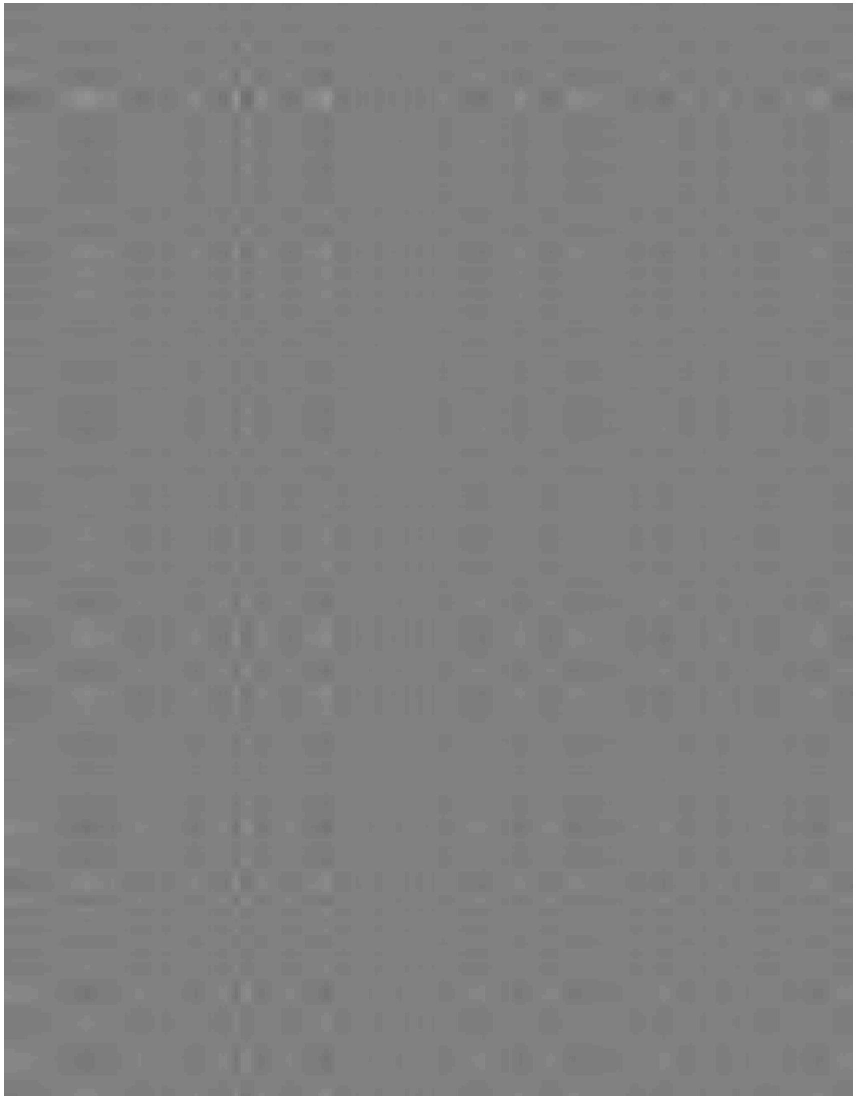


$c_j\ u_j\ v_j^*$ for $j = 20$

partial svd based on 20 singular values



$c_j \, u_j \, v_j^*$ for $j = 21$



The SVD and the LSMN solution

The least-squares minimum norm solution can be computed using the SVD by

$$\mathbf{x}_{LSMN} = \mathbf{V} \begin{bmatrix} \Sigma_r^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{U}^* \mathbf{b} = \mathbf{V}_r \, \Sigma_r^{-1} \, \mathbf{U}_r^* \, \mathbf{b}$$

The matrix

$$\mathbf{A}^\dagger \equiv \mathbf{V} \begin{bmatrix} \Sigma_r^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{U}^* = \mathbf{V}_r \, \Sigma_r^{-1} \, \mathbf{U}_r^*$$

is called the **Moore-Penrose pseudoinverse** of $\mathbf{A}$.

The SVD and the LSMN solution (2)

Theorem. $\mathbf{x}_{LSMN} = \mathbf{A}^{\dagger}\,\mathbf{b}.$

Proof. With $\mathbf{z} \in \mathbb{C}^k$ and $\mathbf{c} \in \mathbb{C}^n$ s.t. $\mathbf{x} = \mathbf{V}\mathbf{z}$ and $\mathbf{b} = \mathbf{U}\mathbf{c}$, partition

$$\mathbf{z} = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix}, \qquad \text{where} \quad \mathbf{z}_1 \equiv \mathbf{V}_r^*\mathbf{x}, \;\; \mathbf{c}_1 \equiv \mathbf{U}_r^*\mathbf{b}.$$

Then, $\mathbf{A}\mathbf{x} = \mathbf{A}\mathbf{V}\mathbf{z} = \mathbf{U}\Sigma\mathbf{z}$ and $\|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2 = \|\mathbf{U}(\mathbf{c} - \Sigma\mathbf{z})\|_2 =$

$$= \left\| \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix} - \begin{bmatrix} \Sigma_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix} \right\|_2 = \left\| \begin{bmatrix} \mathbf{c}_1 - \Sigma_r\mathbf{z}_1 \\ \mathbf{c}_2 \end{bmatrix} \right\|_2 :$$

$\|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2$ is minimised by $\mathbf{z}_1 = \Sigma_r^{-1}\mathbf{c}_1$ and $\|\mathbf{x}\|_2$ by $\mathbf{z}_2 = \mathbf{0}.$

The SVD and the LSMN solution (3)

Theorem. $\mathbf{x}_{LSMN} = \mathbf{A}^\dagger\,\mathbf{b}$.

Note that, except for the ordering, the

$$\frac{1}{\sigma_i} \quad \text{for} \quad \sigma_i > 0,$$

form the non-zero singular values of the solution operator $\mathbf{A}^\dagger$.

If $(\sigma_i, \mathbf{v}_i, \mathbf{u}_i)$ is a singular triple of $\mathbf{A}$, then $(\frac{1}{\sigma_i}, \mathbf{u}_i, \mathbf{v}_i)$ ($(0, \mathbf{u}_i, \mathbf{v}_i)$ if $\sigma_i = 0$) is a singular triple of $\mathbf{A}^\dagger$.

With $\beta_i \equiv \mathbf{u}_i^*\mathbf{b}$, $\beta_i\mathbf{u}_i$ is the component of $\mathbf{b}$ in the direction $\mathbf{u}_i$.
With $\alpha_i \equiv \mathbf{v}_i^*\mathbf{x}_{LSMN}$, $\alpha_i\mathbf{v}_i$ is the comp. of $\mathbf{x}_{LSMN}$ is the dir. $\mathbf{v}_i$.
Then

$$\alpha_i = \frac{1}{\sigma_i}\beta_i \quad (i = 1, \ldots, r), \qquad \alpha_i = 0 \quad (i > r).$$

Noisy problems

In least-squares problems $\mathbf{b}$ often corresponds to measured date, which means that we are actually solving the **noisy problem**

$$\mathbf{A}\mathbf{x} = \mathbf{b} + \delta_b .$$

On average noise is equally large in the direction of all singular vectors, while the 'ideal' solution $\mathbf{x}$ typically has small components in the 'singular direction' with small singular values.

These small singular values have a dramatic effect on the LSMN-solution (why?)!!!

This is an example of a so-called **ill-posed problem**: small perturbations in the data give a large perturbation in the solution.

Regularization

Limiting this effect is called **regularisation**.

Several regularization methods have been proposed:

- Set small singular values to 0. This requires the explicit calculation of the SVD, which is not possible for large scale problems.

If σ_i is a singular value of $\mathbf{A}$, then

$$\begin{cases} \frac{1}{\sigma_i} & \textit{if} \quad \sigma_i \geq \delta \\ 0 & \textit{if} \quad \sigma_i < \delta \end{cases}$$

is a singular value of the solution operator for this method.

Regularization

Limiting this effect is called **regularisation**.

Several regularization methods have been proposed:

- Set small singular values to 0. This requires the explicit calculation of the SVD, which is not possible for large scale problems.

If σ_i is a singular value of $\mathbf{A}$, then

$$\begin{cases} \frac{1}{\sigma_i} & \textit{if} \quad \sigma_i \geq \delta \\ 0 & \textit{if} \quad \sigma_i < \delta \end{cases}$$

is a singular value of the solution operator for this method.

Regularization

Limiting this effect is called **regularisation**.

Several regularization methods have been proposed:

- **Tykhonov regularisation.** Solve the **damped** least squares problem:

$$\min_{\mathbf{x}} \left\| \begin{bmatrix} \mathbf{A} \\ \tau \mathbf{I} \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \right\|_2 .$$

If σ_i *is a singular value of* $\mathbf{A}$*, then*

$$\frac{\sigma_i}{\sigma_i^2 + \tau^2}$$

is a singular value of the solution operator for Tykhonov reg..

Regularization

Limiting this effect is called **regularisation**.

Several regularization methods have been proposed:

- Use an iterative method as CG for the normal equations.

If σ_i is a singular value of $\mathbf{A}$, then

$$q_j(\sigma_i^2)\,\sigma_i$$

is a singular value of this solution operator at step k.

Here, q_j is the $j-1$ degree solution polynomial: $1-\lambda q_j(\lambda)$ is the CG residual polynomial. Note that $q_j(\lambda) \approx \frac{1}{\lambda}$ for non-zero eigenvalues $\lambda = \sigma_i^2$ of $\mathbf{A}^*\mathbf{A}$. Moreover, convergence to small eigenvalues λ (i.e., small singular values) is slow.

Regularization

Limiting this effect is called **regularisation**.
Several regularization methods have been proposed.

Limiting the effect of amplification of small perturbations in the data introduces approximation errors. Regularization techniques have to balance between these two effects, i.e., find an appropriate values for δ, τ, and j, respectively.

If, for instance, for each τ (i.e., many τ), $\mathbf{x}_\tau$ is the solution as obtained by Tykhonov regularisation, then inspection of the so-called **L-curve** $\tau \rightsquigarrow (\|\mathbf{b} - \mathbf{A}\mathbf{x}_\tau\|_2, \|\mathbf{x}_\tau\|_2)$ may suggest an appropriate value for τ: select a τ that corresponds to a point in the 'elbow' of the L-curve.

Conjugate Gradient Least Square

CG can always be applied to the normal equations

$$\mathbf{A}^*\,\mathbf{A}\mathbf{x} = \mathbf{A}^*\,\mathbf{b}$$

since $\mathbf{A}^*\,\mathbf{A}$ is Hermitian positive semi-definite.

The stability can be improved by replacing inner products

$$\mathbf{u}^*\,(\mathbf{A}^*\,\mathbf{A}\mathbf{u}) \quad \text{by inner products} \quad (\mathbf{A}\mathbf{u})^*\,\mathbf{A}\mathbf{u}$$

which leads to the algorithm **CGLS**.

Note that, with $\mathbf{x}_0 = \mathbf{0}$, we have that $\mathbf{x}_j \in \mathcal{R}(\mathbf{A}^*) = \mathcal{N}(\mathbf{A})^\perp$.

CGLS (2)

```
$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0,$
$\mathbf{s}_0 = \mathbf{A}^* \mathbf{r}_0,\ \mathbf{u}_0 = \mathbf{s}_0,\ \rho_0 = \mathbf{s}_0^* \mathbf{s}_0$          %% Initialization
for $j = 0, 1, \ldots,$ do
    $\mathbf{c}_j = \mathbf{A}\mathbf{u}_j$
    $\sigma_j = \mathbf{c}_j^* \mathbf{c}_j,\ \alpha_j = \frac{\rho_j}{\sigma_j}$
    $\mathbf{x}_{j+1} = \mathbf{x}_j + \alpha_j \mathbf{u}_j$          %% update iterate
    $\mathbf{r}_{j+1} = \mathbf{r}_j - \alpha_j \mathbf{c}_j$          %% update residual
    $\mathbf{s}_{j+1} = \mathbf{A}^* \mathbf{r}_{j+1}$          %% residual normal equations
    $\rho_{j+1} = \mathbf{s}_{j+1}^* \mathbf{s}_{j+1},\ \beta_j = \frac{\rho_{j+1}}{\rho_j}$
    $\mathbf{u}_{j+1} = \mathbf{s}_{j+1} + \beta_j \mathbf{u}_j$          %% update direction vector
end for
```

CGLS (2)

```
$\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}_0,\ \mathbf{x} = \mathbf{x}_0$
$\mathbf{s} = \mathbf{A}^*\mathbf{r},\ \mathbf{u} = \mathbf{s},\ \rho = \mathbf{s}^*\mathbf{s}$        %% Initialization
while $\ldots$ do
    $\mathbf{c} = \mathbf{A}\mathbf{u}$
    $\sigma = \mathbf{c}^*\mathbf{c},\ \alpha = \frac{\rho}{\sigma}$
    $\mathbf{x} \leftarrow \mathbf{x} + \alpha\,\mathbf{u}$        %% update iterate
    $\mathbf{r} \leftarrow \mathbf{r} - \alpha\,\mathbf{c}$        %% update residual
    $\mathbf{s} = \mathbf{A}^*\mathbf{r}$        %% residual normal equations
    $\rho_0 = \rho,\ \rho = \mathbf{s}^*\mathbf{s},\ \beta = \frac{\rho}{\rho_0}$
    $\mathbf{u} \leftarrow \mathbf{s} + \beta\,\mathbf{u}$        %% update direction vector
end for
```

CGLS (3)

CGLS can also be used for solving nonsymmetric square systems. However, this has two important disadvantages:

- The work per iteration is twice as much as in CG;
- $\mathcal{C}_2(\mathbf{A}^*\mathbf{A}) = \mathcal{C}_2(\mathbf{A})^2$,
  which means that convergence is often very slow.

CGLS (4), Assignment

Assuming that $\mathcal{C}_2 \equiv \mathcal{C}_2(\mathbf{A}) = 100$ and $\mathbf{A}$ is Hermitian:

1. Give an upper bound on the number of CG iterations

   required to satisfy $\frac{\|\mathbf{x}-\mathbf{x}_j\|_A}{\|\mathbf{x}-\mathbf{x}_0\|_A} < 10^{-6}$.

   Hint: use the upper bound

$$\frac{\|\mathbf{x}-\mathbf{x}_j\|_A}{\|\mathbf{x}-\mathbf{x}_0\|_A} \leq 2\left(\frac{\sqrt{\mathcal{C}_2}-1}{\sqrt{\mathcal{C}_2}+1}\right)^j \leq 2\exp(-\frac{2j}{\sqrt{\mathcal{C}_2}}).$$

2. Answer the same question for CGLS.

CGLS (4), Assignment

Assuming that $\mathcal{C}_2 \equiv \mathcal{C}_2(\mathbf{A}) = 100$ and $\mathbf{A}$ is Hermitian:

1. Give an upper bound on the number of CG iterations

   required to satisfy $\frac{\|\mathbf{x}-\mathbf{x}_j\|_A}{\|\mathbf{x}-\mathbf{x}_0\|_A} < 10^{-6}$.

   Hint: use the upper bound

$$\frac{\|\mathbf{x}-\mathbf{x}_j\|_A}{\|\mathbf{x}-\mathbf{x}_0\|_A} \leq 2\left(\frac{\sqrt{\mathcal{C}_2}-1}{\sqrt{\mathcal{C}_2}+1}\right)^j \leq 2\,\exp(-\frac{2j}{\sqrt{\mathcal{C}_2}}).$$

2. Answer the same question for CGLS.

Answer: CG: 73, CGLS 726

Orthonormal basis

Arnoldi and Lanczos (GMRES, $\ldots$) construct an orthonormal (partial) basis $\mathbf{V}$ and exploit this basis in both image space and domain space since the transformation to this basis preserves

- eigen structure: $(\mathbf{V}^*\mathbf{A}\mathbf{V})\mathbf{y} = \lambda\mathbf{y} \quad \Leftrightarrow \quad \mathbf{A}\mathbf{x} = \lambda\mathbf{x}$ with $\mathbf{x} = \mathbf{V}\mathbf{y}$.
- matrix polynomial structure: $\mathbf{V}^* p_j(\mathbf{A})\mathbf{V} = p_j(\mathbf{V}^*\mathbf{A}\mathbf{V})$.

This is not the case if we select different basis in image space basis and domain space. However, in the problems of this lecture image space ($\mathbb{C}^n$) is different from domain space ($\mathbb{C}^k$) anyway. From the SVD, we already learnt that it may be helpfull to work with different basis.

Bidiagonalisation

With finitely many Householder reflections, we can obtain

$$\mathbf{AV} = \mathbf{UB}$$

with $\mathbf{V}$ $k \times k$ unitary, $\mathbf{U}$ $n \times n$ unitary,
and $\mathbf{B} = (b_{ij})$ $n \times k$ lower bidiagonal: $b_{ij} = 0$ if $i \not\in \{j, j+1\}$.

Bidiagonal is as close to diagonal as we can get in finite steps.

See also Theorem 3.5.

This suggest to construct recursively, partial $\mathbf{V}$ and $\mathbf{U}$ for high dimensional $\mathbf{A}$.

Bidiagonalisation (2)

Bidiagonalisation algorithm (Golub and Kahan)

```
$\beta_1 \mathbf{u}_1 = \mathbf{b},\ \alpha_1 \mathbf{v}_1 = \mathbf{A}^* \mathbf{u}_1$
for $i = 2, 3, \ldots$ do
    $\beta_i \mathbf{u}_i = \mathbf{A}\mathbf{v}_{i-1} - \alpha_{i-1}\mathbf{u}_{i-1}$
    $\alpha_i \mathbf{v}_i = \mathbf{A}^* \mathbf{u}_i - \beta_i \mathbf{v}_{i-1}$
end for
```

with $\alpha_i > 0$ and $\beta_i > 0$ such that $\|\mathbf{u}_i\| = \|\mathbf{v}_i\| = 1$.

Normalisation leads to orthogonalisation: $\mathbf{u}_i \perp \mathbf{u}_j,\ \ \mathbf{v}_i \perp \mathbf{v}_j\ \ (j < i)$.

Moreover, $\mathbf{v}_k \in \mathcal{R}(\mathbf{A}^*) = \mathcal{N}(\mathbf{A})^\perp$ and $\mathbf{u}_k \in$ span$(\mathbf{b}, \mathcal{R}(\mathbf{A}))$.

See also Exercise 9.6.

Bidiagonalisation (3)

With $\mathbf{U}_j \equiv [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_j]$, $\mathbf{V}_k \equiv [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_j]$,

$$\underline{B}_j \equiv \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \beta_3 & \ddots & \\ & & \ddots & \alpha_j \\ & & & \beta_j \end{bmatrix},$$

and B_j the $j \times j$ upper block of $\underline{B}_j$, we have

$$\beta_1 \mathbf{U}_j e_1 = \mathbf{b}, \quad \mathbf{A} \mathbf{V}_{j-1} = \mathbf{U}_j \underline{B}_{j-1}, \quad \mathbf{A}^* \mathbf{U}_j = \mathbf{V}_j B_j^* .$$

Least Square QR

LSQR constructs approximate solutions of the form $\mathbf{x}_j = \mathbf{V}_j\, y_j$. Then

$$\begin{aligned} \mathbf{r}_j &\equiv \mathbf{b} - \mathbf{A}\mathbf{x}_j = \beta_1 \mathbf{U}_{j+1}\, e_1 - \mathbf{A}\mathbf{V}_j\, y_j \\ &= \beta_1 \mathbf{U}_{j+1} e_1 - \mathbf{U}_{j+1}\, \underline{B}_j\, y_j \\ &= \mathbf{U}_{j+1}(\beta_1 e_1 - \underline{B}_j\, y_j). \end{aligned}$$

Note that $\mathbf{x}_j \perp \mathcal{N}(\mathbf{A})$ and $\mathbf{b} \in$ span$(\mathbf{U}_{j+1})$.

Minimizing $\|\mathbf{r}_j\|_2$ is equivalent to solving the least squares problem

$$\text{min}_y \|\beta_1 e_1 - \underline{B}_j\, y\|_2.$$

LSQR (2)

In LSQR

$$y_j = \text{argmin}_y \|\beta_1 e_1 - \underline{B}_j y\|_2$$

is solved using the QR-decomposition of $\underline{B}_j$:

$$\mathbf{x}_j = (\mathbf{V}_j R_j^{-1})(\underline{Q}_j^*(\beta_1 e_1)), \quad \text{where} \quad \underline{B}_j = \underline{Q}_j R_j.$$

The QR-decomposition in LSQR is based on Givens rotations, R_j is bidiagonal.

LSQR (3)

LSQR is famous for its robustness.

However,

$$\mathbf{A}^*\mathbf{A}\mathbf{V}_j = \mathbf{A}^*\mathbf{U}_{j+1}\underline{B}_j = \mathbf{V}_{j+1}B^*_{j+1}\underline{B}_j.$$

Therefore, $\mathbf{x}_j \in \mathcal{K}_j(\mathbf{A}^*\mathbf{A}, \mathbf{A}^*\mathbf{b})$. Similarly, $\mathbf{r}_j \in \mathcal{K}_{j+1}(\mathbf{A}\mathbf{A}^*, \mathbf{b})$.

Convergence of LSQR is as of CGLS.

Bidiagonalisation and Lanczos

Note that the normal equation

$$\mathbf{A}^*\mathbf{A}\mathbf{x} = \mathbf{A}^*\mathbf{b}$$

is equivalent to the augmented system

$$\begin{bmatrix} \mathbf{I} & \mathbf{A} \\ \mathbf{A}^* & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix}.$$

We will now argue that LSQR can be viewed as a MINRES variant applied to the ‘augmented’ form of the normal equations. In particular, we will derive the bidiagonalisation by applying Lanczos to the augmeneted system.

Bidiagonalisation and Lanczos

Applying Lanczos to

$$\begin{bmatrix} \mathbf{I} & \mathbf{A} \\ \mathbf{A}^* & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \quad \text{starting with} \quad \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix}.$$

This leads to the Lanczos vectors

$$\begin{bmatrix} \mathbf{u}_1 \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{0} \\ \mathbf{v}_1 \end{bmatrix}, \begin{bmatrix} \mathbf{u}_2 \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{0} \\ \mathbf{v}_2 \end{bmatrix}, \begin{bmatrix} \mathbf{u}_3 \\ \mathbf{0} \end{bmatrix}, \ldots$$

with $\mathbf{u}_i$ and $\mathbf{v}_i$ as in bidiagonalisation.

Lanczos and LSQR

Rearranging the basis, substitution in the augmented system, using $\mathbf{x}_j = \mathbf{V}_j\, y_j$, $\mathbf{r}_j = \mathbf{U}_{j+1}\, t_j$, and and the Gallerkin condition, gives

$$\begin{bmatrix} \mathbf{U}_{j+1}^* & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_j^* \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{A} \\ \mathbf{A}^* & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{U}_{j+1}\, t_j \\ \mathbf{V}_j\, y_j \end{bmatrix} = \begin{bmatrix} \mathbf{U}_{j+1}^* \mathbf{b} \\ \mathbf{0} \end{bmatrix},$$

which leads to the lower dimensional (tridiagonal) system

$$\begin{bmatrix} I & \underline{B}_j \\ \underline{B}_j^* & 0 \end{bmatrix} \begin{bmatrix} t_j \\ y_j \end{bmatrix} = \begin{bmatrix} \beta_1 e_1 \\ 0 \end{bmatrix}$$

which is equivalent to $y_j =$ argmin$_y \|\beta_1 e_1 - \underline{B}_j\, y\|_2$.

Lanczos and LSQR (2)

Apparently, Lanczos applied to the (block) structured problem allows computational savings: there is no need to save the block of zeros in the Lanczos vectors, inner products with zeros are trivial. Bidiagonalisation exploits these savings.

Note. Similarly computational savings strategies can sometimes be applied to other structured problems as well.

Final remarks

Today we have seen CG-type methods for the normal equations.

These methods can also be applied to nonsymmetric systems.

The disadvantage of this approach is that the condition number may be squared compared to the original system. This may lead to slow convergence and/or an inaccurate solution.

However, there are also classes of problems for which the normal equations approach works quite well, in particular if $\mathbf{A}$ is close to an orthogonal matrix.