Matrix theory, Data science

and

Model reduction of large-scale systems

Thanos Antoulas

PART I

MOR-MathCoRe MPI Magdeburg – Summer 2018

Outline

Matrix factorizations

Given $\mathbf{A} \in \mathbb{R}^{n \times m}$

1.	EVD ($n = m$)	$\mathbf{A} = \mathbf{X}\Lambda\mathbf{X}^{-1}$	$\det \mathbf{X} \neq 0$, Λ = diagonal
2.	SVD	$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$	$\mathbf{U}, \mathbf{V}$: orthogonal $\mathbf{\Sigma}$ = diagonal
3.	Schur decomposition ($n = m$)	$\mathbf{A} = \mathbf{U}\Delta\mathbf{U}^*$	$\mathbf{U}$ orthogonal, Δ upper triangular
4.	QR factorization	$\mathbf{A} = \mathbf{QR}$	$\mathbf{Q}$ orthogonal, $\mathbf{R}$ upper triangular
5.	LU factorization ($n = m$)	$\mathbf{A} = \mathbf{LU}$	$\mathbf{L}$ lower triang, $\mathbf{U}$ upper triang.
6.	Cholesky factorization ($n = m$)	$\mathbf{A} = \mathbf{LL}^*$	$\mathbf{L}$ lower (upper) triangular
7.	CUR factorization	$\mathbf{A} \approx \mathbf{CUR}$	$\mathbf{C} = \mathbf{A}(:, J), \mathbf{U} = \mathbf{A}(I, J)^{-1}, \mathbf{R} = \mathbf{A}(I, :)$

The eigenvalue decomposition (EVD). Given a square matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, its EVD is given by:

$$\mathbf{A} = \mathbf{X}\Lambda\mathbf{X}^{-1} \quad \Leftrightarrow \quad \mathbf{A}\mathbf{X} = \mathbf{X}\Lambda,$$

where $\det \mathbf{X} \neq 0$ and Λ is the diagonal matrix of *eigenvalues*:

$$\Lambda = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}, \quad \lambda_i \in \mathbb{C}$$

Therefore the columns of $\mathbf{X} = [\mathbf{x}_1 \ \cdots \ \mathbf{x}_n]$, are the *eigenvectors* corresponding to the λ_i's:

$$\mathbf{A}\mathbf{x}_i = \lambda_i \mathbf{x}_i, \ i = 1, \ \cdots \ , n.$$

Issues concerning the above decomposition:

- It is *not* always possible to diagonalize a matrix in this form. For instance the *Jordan block*

$$\mathbf{A} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix},$$

  is not diagonalizable because its eigenvectors do not form a basis for $\mathbb{R}^2$.

- The eigenvalues may be complex. For instance, the eigenvalues of

$$\mathbf{A} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix},$$

which are the roots of the characteristic polynomial, are purely imaginary:

$$\det(\lambda \mathbf{I} - \mathbf{A}) = 0 \quad \Rightarrow \quad \lambda^2 + 1 = 0 \quad \text{are} \quad \lambda_{1,2} = \pm j \in \mathbb{C}.$$

- The EVD of *symmetric* matrices $\mathbf{A}$, i.e. $\mathbf{A} = \mathbf{A}^T$, or *Hermitian* matrices, i.e. $\mathbf{A} \in \mathbb{C}^{n \times n}$, $\mathbf{A} = \mathbf{A}^*$, has two important properties.

(i) The eigenvalues are real.

(ii) Eigenvectors corresponding to distinct eigenvalues are orthogonal.

Consequence: there always exists an orthonornal basis of eigenvectors for $\mathbb{R}^n$, i.e. $\mathbf{X}$ can be chosen orthogonal $\mathbf{X}\mathbf{X}^T = \mathbf{I}_n$ (or unitary: $\mathbf{X}\mathbf{X}^* = \mathbf{I}_n$). Hence

$$\mathbf{A} = \mathbf{X}\Lambda\mathbf{X}^T \quad \text{or} \quad \mathbf{A} = \mathbf{X}\Lambda\mathbf{X}^*.$$

The singular value decomposition (SVD). Given a matrix $\mathbf{A} \in \mathbb{C}^{n\times m}$, $n \leq m$, there exist unitary matrices $\mathbf{U} \in \mathbb{C}^{n\times n}$, $\mathbf{U}\mathbf{U}^* = \mathbf{I}_n$, and $\mathbf{V} \in \mathbb{C}^{m\times m}$, $\mathbf{V}\mathbf{V}^* = \mathbf{I}_m$, such that

$$\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^*$$

where $\boldsymbol{\Sigma}$ is an $n \times m$ matrix with $\boldsymbol{\Sigma}_{ii} = \sigma_i$, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$, $i = 1, \cdots, n$, and zero elsewhere.

This is the **singular value decomposition (SVD)** of the matrix $\mathbf{A}$; σ_i are the **singular values** of $\mathbf{A}$ while the columns of $\mathbf{U}$ and $\mathbf{V}$

$$\mathbf{U} = (\mathbf{u}_1 \quad \mathbf{u}_2 \quad \cdots \quad \mathbf{u}_n), \quad \mathbf{V} = (\mathbf{v}_1 \quad \mathbf{v}_2 \quad \cdots \quad \mathbf{v}_m)$$

are called the **left** and **right singular vectors** of $\mathbf{A}$, respectively. Since

$$\mathbf{A}\mathbf{A}^* = \mathbf{U} \underbrace{\begin{bmatrix} \sigma_1^2 & & \\ & \ddots & \\ & & \sigma_n^2 \end{bmatrix}}_{\boldsymbol{\Sigma}\boldsymbol{\Sigma}^*} \mathbf{U}^* \quad \text{and} \quad \mathbf{A}^*\mathbf{A} = \mathbf{V} \underbrace{\begin{bmatrix} \boldsymbol{\Sigma}\boldsymbol{\Sigma}^* & \\ & \mathbf{0} \end{bmatrix}}_{\boldsymbol{\Sigma}^*\boldsymbol{\Sigma}} \mathbf{V}^*,$$

these singular vectors are the *eigenvectors* of $\mathbf{A}\mathbf{A}^*$ and $\mathbf{A}^*\mathbf{A}$, respectively. From (6) follows:

$$\mathbf{A}\mathbf{v}_i = \sigma_i \mathbf{u}_i, \quad i = 1, \quad \cdots \quad , n$$

Example 1. Consider the matrix

$$\mathbf{A} = \begin{bmatrix} 1 & -3 \\ 3 & -1 \end{bmatrix}.$$

It readily follows that the eigenvalue decompositions of the matrices

$$\mathbf{A}\mathbf{A}^* = \begin{bmatrix} 10 & 6 \\ 6 & 10 \end{bmatrix} \quad \text{and} \quad \mathbf{A}^*\mathbf{A} = \begin{bmatrix} 10 & -6 \\ -6 & 10 \end{bmatrix},$$

are:

$$\mathbf{A}\mathbf{A}^* = \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^* \quad \text{and} \quad \mathbf{A}^*\mathbf{A} = \mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^*,$$

where

$$\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2] = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$$

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix}, \ \sigma_1 = 4, \ \sigma_2 = 2$$

$$\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2] = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}.$$

Remarks:

- σ_i^2 are the eigenvalues of $\mathbf{AA}^*$ and $\mathbf{u}_i$ are the corresponding eigenvectors, i=1,2.
- σ_i^2 are also the eigenvalues of $\mathbf{A}^*\mathbf{A}$ and $\mathbf{v}_i$ are the corresponding eigenvectors, i=1,2.
- $\mathbf{A} = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^* + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^*$. Furthermore, $\mathbf{A}$ maps $\mathbf{v}_1 \mapsto \sigma_1 \mathbf{u}_1$ and $\mathbf{v}_2 \mapsto \sigma_2 \mathbf{u}_2$ (see figure 1). This shows that the SVD maps the **unit circle** into an **ellipsoid**, where $\mathbf{Av}_1 = \sigma_1 \mathbf{u}_1$ and $\mathbf{Av}_2 = \sigma_2 \mathbf{u}_2$ give the major and minor axes of the ellipsoid, respectively. The **maximum amplification** factor is given by σ_1, the largest singular value. (In `MATLAB` use the command `eigshow`).

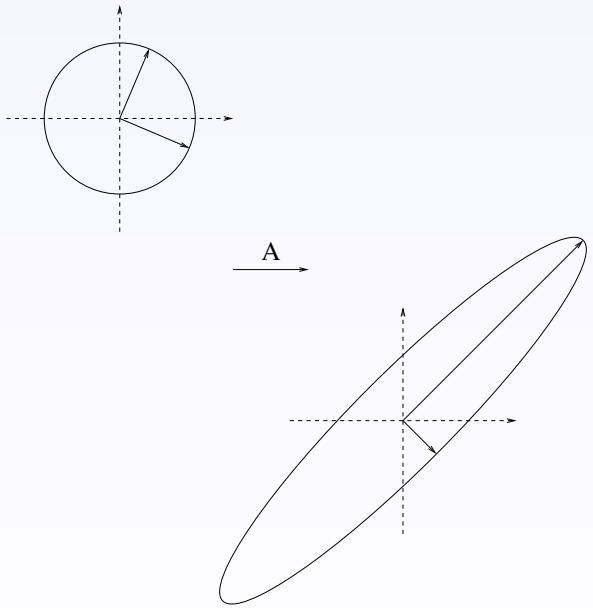


Figure: Quantities describing the singular value decomposition in $\mathbb{R}^2$

Example 2. Let

$$\mathbf{A} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix},$$

clearly $\mathbf{A}$ does not have an EVD, since it is not square, but it has and SVD $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$:

$$\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3] = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{\Sigma} = \begin{pmatrix} \sqrt{2} \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{V} = [\mathbf{v}_1] = 1. \qquad \blacksquare$$

Example 3. Consider $\mathbf{A} = \begin{bmatrix} 3 & 4 \\ 0 & 0 \end{bmatrix}$. Again, $\mathbf{A}$ is not invertible so an EVD is not possible.

Let us proceed for its SVD: $\mathbf{A}\mathbf{A}^* = \begin{bmatrix} 25 & 0 \\ 0 & 0 \end{bmatrix}$ and $\mathbf{A}^*\mathbf{A} = \begin{bmatrix} 9 & 12 \\ 12 & 16 \end{bmatrix}$

$$\mathbf{A}\mathbf{A}^* = \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^* \quad \text{and} \quad \mathbf{A}^*\mathbf{A} = \mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^*, \text{ where } \mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix}, \ \sigma_1 = 5, \ \sigma_2 = 0 \ \text{ and } \ \mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2] = \frac{1}{5}\begin{bmatrix} 3 & 4 \\ -4 & 3 \end{bmatrix}.$$

Notice how $\mathbf{A}$ maps $\mathbf{v}_1 \mapsto 5\mathbf{u}_1$ and $\mathbf{v}_2 \mapsto 0$, so that the ellipsoid in figure 1 is reduced to the interval $[-5, 5]$ on the x-axis. This is because $\sigma_2 = 0$. $\blacksquare$

Properties of the SVD

1. Assume that in (6) $\sigma_r > 0$ while $\sigma_{r+1} = 0$; the matrices $\mathbf{U}$, $\mathbf{\Sigma}$, $\mathbf{V}$ are partitioned compatibly in two blocks, the first having r columns:

$$\mathbf{U} = [\mathbf{U}_1 \quad \mathbf{U}_2], \quad \mathbf{\Sigma} = \begin{pmatrix} \mathbf{\Sigma}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_2 \end{pmatrix} \in \mathbb{R}^{n \times m} \quad \text{and} \quad \mathbf{V} = [\mathbf{V}_1 \quad \mathbf{V}_2],$$

$$\mathbf{\Sigma}_1 = \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{pmatrix} > \mathbf{0}, \quad \mathbf{\Sigma}_2 = \mathbf{0} \in \mathbb{R}^{(n-r)\times(m-r)},$$

   where $\mathbf{U}_1$, $\mathbf{U}_2$ have r, $n-r$ columns, and $\mathbf{V}_1$, $\mathbf{V}_2$ have r, $m-r$ columns respectively. Given (6) and (1) the following statements hold.

2. $\operatorname{rank} \mathbf{A} = r$.

3. $$\begin{cases} \operatorname{im} \mathbf{A} = \text{span col } [\mathbf{u}_1, \cdots, \mathbf{u}_r], & \ker \mathbf{A} = \text{span col } [\mathbf{v}_{r+1}, \cdots, \mathbf{v}_m], \\ \operatorname{im} \mathbf{A}^T = \text{span col } [\mathbf{v}_1, \cdots, \mathbf{v}_r], & \ker \mathbf{A}^T = \text{span col } [\mathbf{u}_{r+1}, \cdots, \mathbf{u}_n]. \end{cases}$$

4. **Dyadic decomposition**. Decomposition as a sum of r outer products of rank one:

$$\mathbf{A} = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^* + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^* + \quad \cdots \quad + \sigma_r \mathbf{u}_r \mathbf{v}_r^*.$$

5. The largest singular value of a matrix $\mathbf{A}$ is equal to its induced 2-norm: $\sigma_1 = \|\mathbf{A}\|_2$, where the induced 2-norm of a $\mathbf{A}$ is defined as:

$$\|\mathbf{A}\|_2 = \sup_{\mathbf{x} \neq 0} \frac{\|\mathbf{A}\mathbf{x}\|_2}{\|\mathbf{x}\|_2}.$$

Optimal approximation in the 2-norm

The problem of *approximating* a matrix by one of lower rank is as follows.

Problem.
- Given $\mathbf{A} \in \mathbb{C}^{n\times m}$, $\operatorname{rank} \mathbf{A} = r \leq n \leq m$,
- Find $\mathbf{X} \in \mathbb{C}^{n\times m}$, $\operatorname{rank} \mathbf{X} = k < r$, such that
- **the 2-norm of the error matrix $\mathbf{E} = \mathbf{A} - \mathbf{X}$, is minimized**

Remark.

- Given $\mathbf{A}$ of rank r, for all $\mathbf{X}$ of rank less than or equal to k, there holds

$$\|\mathbf{A} - \mathbf{X}\|_2 \geq \sigma_{k+1}(\mathbf{A}).$$

In other words, take **any** matrix $\mathbf{X}$ of rank $\leq k$, then the approximation error can never be smaller than $\sigma_{k+1}(\mathbf{A})$. Finding the best rank k approximant of a matrix is a non-convex (DIFFICULT) optimization problem, but surprisingly, the SVD provides an **explicit** solution!

Solution. **Schmidt, Eckart, Young, Mirsky**. With the notation introduced above

$$\min_{\mathbf{X},\, \mathrm{rank}\mathbf{X}=k} \|\mathbf{A}-\mathbf{X}\|_2 = \sigma_{k+1}(\mathbf{A})$$

provided that $\sigma_k > \sigma_{k+1}$. A (non-unique) minimizer $\mathbf{X}_*$ is obtained by truncating the dyadic decomposition to contain the first k terms:

$$\mathbf{X}_* = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^* + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^* + \ \cdots \ + \sigma_k \mathbf{u}_k \mathbf{v}_k^*.$$

Using the approximation (12), $\mathbf{A} \in \mathbb{C}^{n\times m}$ of rank $r \leq n$ is approximated by a matrix of lower rank $k < r$, by eliminating the $r-k$ smallest singular values, $\sigma_{k+1} \ \cdots \ \sigma_r$:

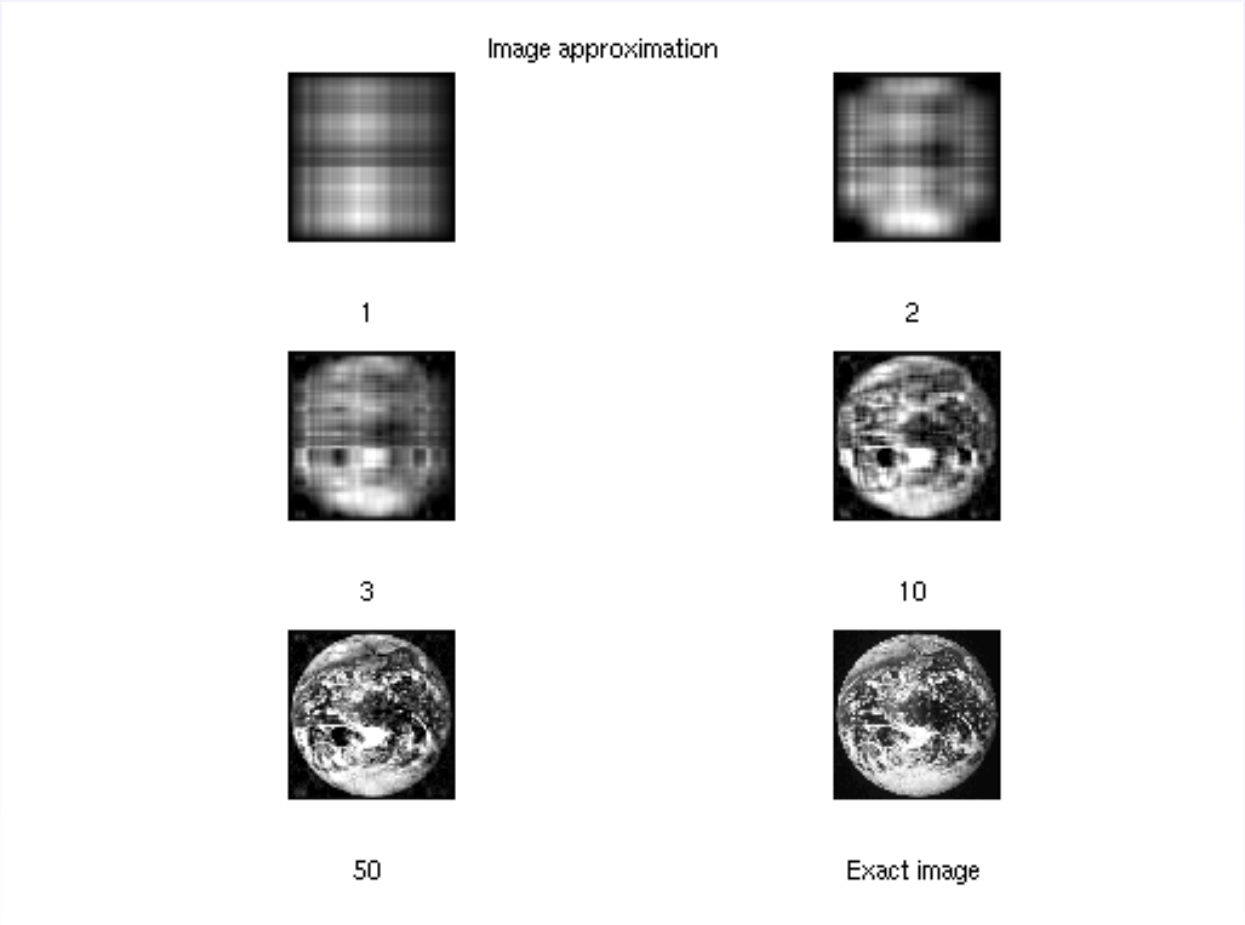
$$\widehat{\mathbf{A}} = \mathbf{U}_k \widehat{\mathbf{\Sigma}}_k \mathbf{V}_k^*, \quad \text{where:}$$

$$\mathbf{U}_k = [\mathbf{u}_1, ..., \mathbf{u}_k] \in \mathbb{C}^{n\times k}, \quad \mathbf{V}_k = [\mathbf{v}_1, ..., \mathbf{v}_k] \in \mathbb{C}^{m\times k}, \quad \hat{\mathbf{\Sigma}}_k = \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_k \end{pmatrix} \in \mathbb{R}^{k\times k}.$$

The storage required is thus reduced from

$$\boxed{\mathbf{n}\times\mathbf{m}} \quad \text{to} \quad \boxed{\mathbf{n}\times\mathbf{k}+\mathbf{k}+\mathbf{k}\times\mathbf{m} \ = \ \mathbf{k}\times(\mathbf{n}+\mathbf{m}+\mathbf{1})}.$$

Example. Application of the theory to the *approximation of static systems*, in particular **image approximation**. Any greyscale image is stored as a matrix, whose entries are the levels of grey corresponding to each pixel. Figure **??** shows a 250×250 image of the earth together with its lower rank approximants. Notice how a rank 50 approximation is indistinguishable from the original rank 250 image.



Earth image approximation by images of lower rank. Compression of the rank 50 approximant:
$62500 \div 2550 \simeq 24 \div 1$

A lower rank k image approximation is obtained by retaining the k most significant singular values, as given by (12) and shown in figure 2.

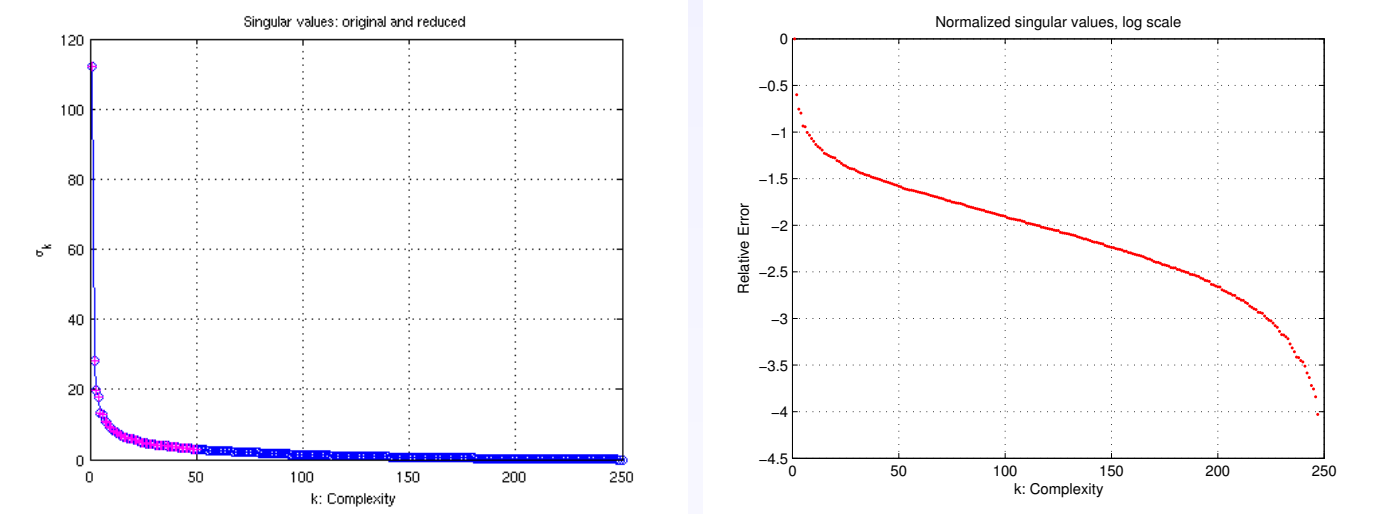


Figure: Left pane: Singular values: original and rank 50 approximation. Right pane: Normalized singular values providing relative approximation error

Furthermore, the singular values provide the

trade-off between accuracy and complexity

This is shown in the normalized singular value plot in figure 2, where specifying a desired relative error on the y-axis gives the required complexity (rank) of the approximation on the x-axis.

Generalized inverses or Pseudoinverses

Let $\mathbf{A} \in \mathbb{R}^{n \times m}$. We call the matrix $\mathbf{A}^{\#} \in \mathbb{R}^{m \times n}$, a generalized inverse or a pseudoinverse of $\mathbf{A}$, provided that the following four conditions are satisfied:

1. $\mathbf{A}\mathbf{A}^{\#}\mathbf{A} = \mathbf{A}$	2. $\mathbf{A}^{\#}\mathbf{A}\mathbf{A}^{\#} = \mathbf{A}^{\#}$
3. $\left[\mathbf{A}\mathbf{A}^{\#}\right]^T = \mathbf{A}\mathbf{A}^{\#}$	4. $\left[\mathbf{A}^{\#}\mathbf{A}\right]^T = \mathbf{A}^{\#}\mathbf{A}$

In particular, if $\mathbf{A}$ has linearly independent columns ($\Rightarrow \mathbf{A}^T\mathbf{A}$ is invertible), the pseudoinverse is a left inverse:

$$\mathbf{A}^{\#} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T \quad \Rightarrow \quad \mathbf{A}^{\#}\mathbf{A} = \mathbf{I}_m.$$

Similarly, if $\mathbf{A}$ has full row rank, the right inverse of $\mathbf{A}$ is

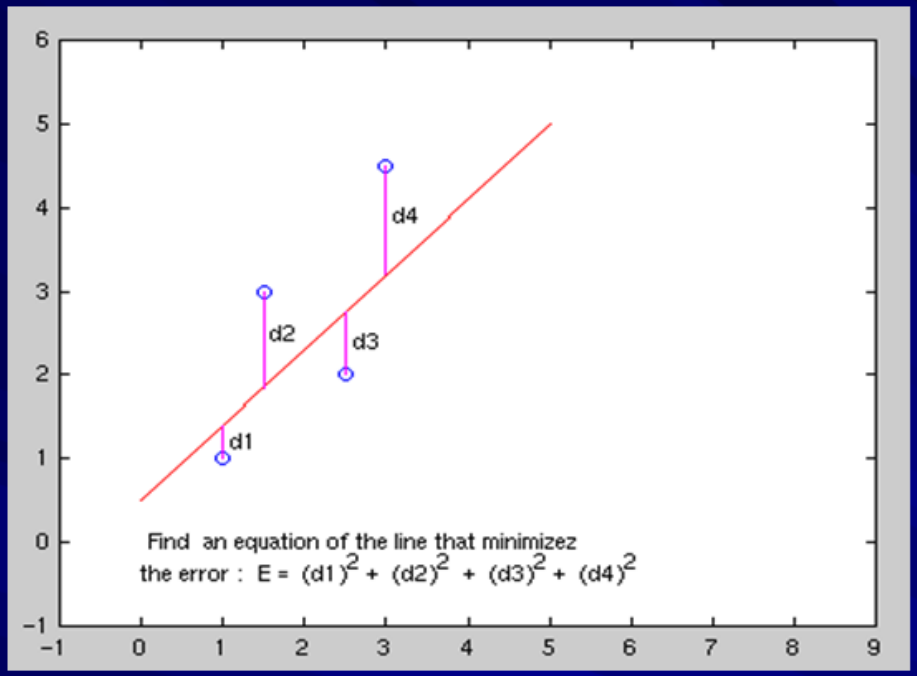
$$\mathbf{A}^{\#} = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1} \quad \Rightarrow \quad \mathbf{A}\mathbf{A}^{\#} = \mathbf{I}_n.$$

Properties of $\mathbf{A}^{\#}$:

1. If $\mathbf{A}$ is invertible, its pseudoinverse is its inverse: $\mathbf{A}^{\#} = \mathbf{A}^{-1}$.
2. The pseudoinverse of a zero matrix is its transpose.
3. The pseudoinverse of the pseudoinverse is the original matrix.
4. If the linear system of equations $\mathbf{A}\mathbf{x} = \mathbf{b}$ is solvable, all solutions are all given by

$$\mathbf{x} = \mathbf{A}^{\#}\mathbf{b} + (\mathbf{I} - \mathbf{A}^{\#}\mathbf{A})\mathbf{w} \quad \text{where} \quad \mathbf{w} \in \mathbb{R}^n, \quad \text{is arbitrary.}$$

Least squares



We want to minimize the vertical distance between the point and the line.

- $E = (d_1)^2 + (d_2)^2 + (d_3)^2 + \ldots + (d_n)^2$ for n data points
- $E = [f(x_1) - y_1]^2 + [f(x_2) - y_2]^2 + \ldots + [f(x_n) - y_n]^2$
- $E = [mx_1 + b - y_1]^2 + [mx_2 + b - y_2]^2 + \ldots + [mx_n + b - y_n]^2$
- $E = \sum (mx_i + b - y_i)^2$

Further properties. • The short form of the SVD is $\mathbf{A} = \mathbf{U}_1 \mathbf{\Sigma}_1 \mathbf{V}_1^*$, where $\mathbf{U}_1 \in \mathbb{R}^{n \times r}$, $\mathbf{\Sigma}_1 \in \mathbb{R}^{r \times r}$, and $\mathbf{V}_1 \in \mathbb{R}^{m \times r}$, where r is the rank of $\mathbf{A}$.

• Moore-Penrose Pseudoinverse: $\mathbf{A}^{\#} = \mathbf{V}_1 \mathbf{S}_1^{-1} \mathbf{U}_1^*$.

• Connection with LS (Least Squares).

Problem: find $\mathbf{x}$ such that $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2$ is minimized. Solution and error:

$$\mathbf{x}_{\mathrm{LS}} = \mathbf{A}^{\#}\mathbf{b} = \sum_{i=1}^{r} \frac{\mathbf{u}_i^* \mathbf{b}}{\sigma_i} \mathbf{v}_i \qquad \rho_{\mathrm{LS}}^2 = \sum_{i=m+1}^{n} (u_i^* \mathbf{b})^2$$

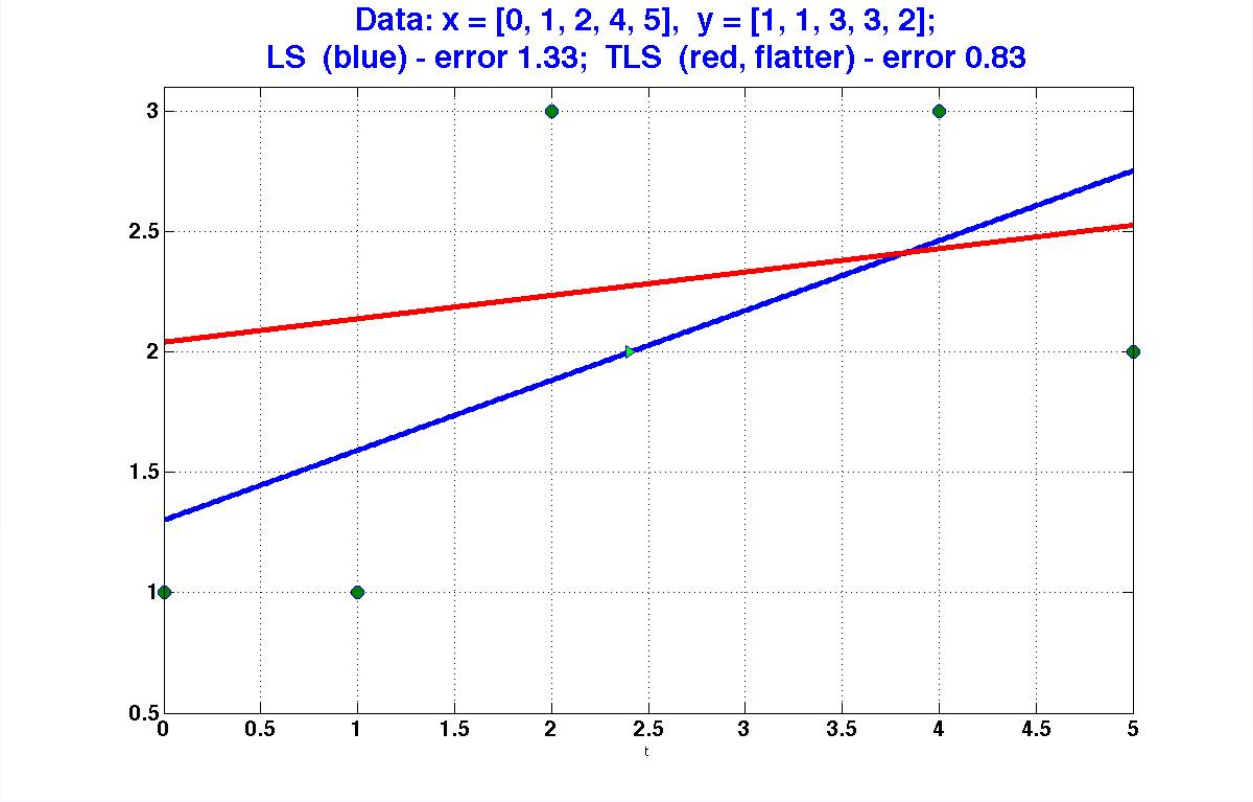
• Uniqueness: the outer products are unique, and thus, given a pair of left, right singular vectors $(\mathbf{u}_i, \mathbf{v}_i)$, $i = 1, \cdots, r$, the only other option for this pair is $(-\mathbf{u}_i, -\mathbf{v}_i)$. On the other hand, the columns of $\mathbf{U}_2$ are arbitrary subject to the constraint that they be linearly independent, normalized, and orthogonal to the columns of $\mathbf{U}_1$. Similarly the columns of $\mathbf{V}_2$ are arbitrary, subject to linear independence, normalization, and orthogonality with the columns of $\mathbf{V}_1$. Thus $\mathbf{U}_2$, $\mathbf{V}_2$ are not necessary for the computation of the SVD of $\mathbf{A}$.

• In MATLAB the command `svd(A)` computes the full SVD of $\mathbf{A}$, while the command `svds(A,k)` computes a short SVD containing k terms, that is the first k singular values and singular vectors. The use of the short SVD is recommended for $\min(n, m) \gg 1$. ■

Example illustrating LS and TLS

```
>> % DATA
>> x=[0 1 2 4 5]';y=[1 1 3 3 2]';A=[x ones(5,1)];
>> % LS solution
>> sol=inv(A'*A)*A'*y
sol =
   2.9070e-01
   1.3023e+00
>> % TLS solution
>> [u,s,v]=svd([A y])
u =
   9.4604e-02  -5.0562e-01  -5.5333e-01   3.1206e-02   6.5441e-01
   1.9007e-01  -2.2358e-01  -6.0493e-01   1.7219e-01  -7.1993e-01
   4.1879e-01  -6.0290e-01   3.9970e-01  -5.2409e-01  -1.6340e-01
   6.0973e-01  -3.8812e-02   2.9650e-01   7.2748e-01   9.7884e-02
   6.3857e-01   5.7391e-01  -2.8321e-01  -4.0679e-01   1.3104e-01
s =
   8.3520e+00            0            0
            0   2.1344e+00            0
            0            0   8.3014e-01
            0            0            0
            0            0            0
v =
   7.9735e-01   6.0200e-01  -4.2835e-02
   2.3369e-01  -3.7340e-01  -8.9775e-01
   5.5644e-01  -7.0581e-01   4.3841e-01
>> v(:,3)/v(3,3)
ans =
  -9.7705e-02
  -2.0477e+00
   1.0000e+00
>> ezplot(.29*t+1.30,[0,5]); hold; ezplot(.097*t+2.04,[0,5]); plot(x,y,'*');
>>
```

```
>> errors = [norm(sol) s(3,3)] =   [1.3344e+00   8.3014e-01]
```



- LS minimizes the sum of the squares of the **vertical distances** to the approximant,
- TLS minimizes the sum of the squares of the **perpendicular distances** to the approximant.

History of the SVD[1]

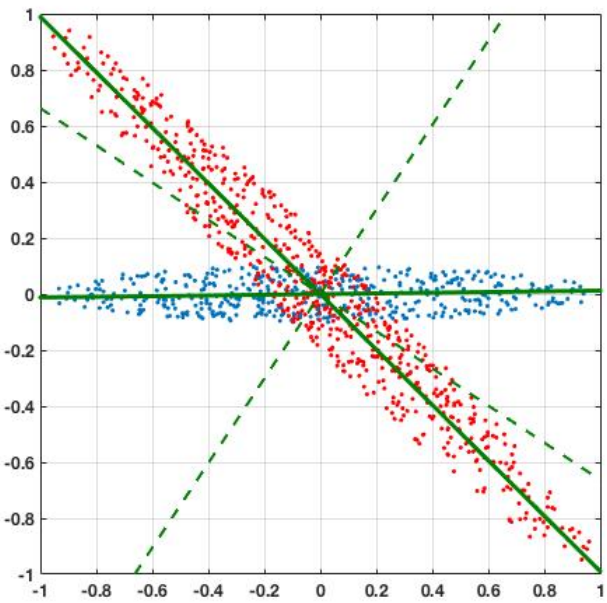
Eugenio Beltrami	(1835-1899)	discovered the SVD in 1873	Lin. Alg.
Camille Jordan	(1838-1921)	co-discoverer in 1874	Lin. Alg.
James Sylvester	(1814-1897)	rediscovered some of these results	Lin. Alg.
		The above papers discuss the diagonalizability of quadratic forms: $\mathbf{y}^T\mathbf{A}\mathbf{x}$.	
Erhard Schmidt[2]	(1876-1959)	Introduced infinite dimensional SVDs and showed how to use it to obtain optimal low rank approximations of an operator (1907).	Int. Eqns
Hermann Weyl	(1885-1955)	Developed a perturbation results for the SVs and used to prove the approximation result (1912).	Int. Eqns
Eckart-Young	1936	C. Eckart, G. Young, The approximation of one matrix by another of lower rank. Psychometrika, Volume 1, Pages 211-218 (1936).	rectangular matrices

[1] G.W. Stewart, On the early history of the singular value decomposition, SIAM Review, **35**: 551-566 (1993).

[2] ... of the Gram-Schmid orthogonalization fame.

Motivating example for the CUR factorization

Find the **principal axes** of the 2-dimensional data set below.



In the above figure the dashed green lines are obtained by means of the SVD while the solid green lines are obtained by means of the CUR factorization.

The CUR factorization

$$\mathbf{\Theta} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \\ \mathbf{C}\mathbf{A}^{-1} & \mathbf{I} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{A} & \\ & \mathbf{\Gamma} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I} & \mathbf{A}^{-1}\mathbf{B} \\ & \mathbf{I} \end{bmatrix},$$

where $\mathbf{\Gamma} = \mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B}$, is the **Schur complement**. This implies

$$\det \mathbf{\Theta} = \det \mathbf{A} \cdot \det \mathbf{\Gamma} = \det \mathbf{\Theta} = \det \mathbf{A} \cdot \det \left(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B} \right)$$

$$\Rightarrow \quad \det \left(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B} \right)^{-1} = \frac{\det \mathbf{A}}{\det \mathbf{\Theta}} \qquad (*)$$

Also

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{I} & -\mathbf{A}^{-1}\mathbf{B} \\ & \mathbf{I} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{A}^{-1} & \\ & \mathbf{\Gamma}^{-1} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I} & \\ -\mathbf{C}\mathbf{A}^{-1} & \mathbf{I} \end{bmatrix}.$$

From (*) follows that

$$\|\mathbf{\Theta}^{-1}\|_C = |\mathbf{\Gamma}^{-1}|.$$

Now given that the 2-norm and the Chebyshev norm are related as:

$$\|\mathbf{M}\|_C \leq \|\mathbf{M}\|_2 \leq n\|\mathbf{M}\|_C,$$

there follows

$$\|\mathbf{\Theta}\|_2 = \sigma_1(\mathbf{\Theta}) \ \text{ and } \ \|\mathbf{\Theta}^{-1}\|_2 = \sigma_1(\mathbf{\Theta}^{-1}) = (\sigma_{\min}(\mathbf{\Theta}))^{-1} = (\sigma_{k+1}(\mathbf{\Theta}))^{-1} \quad \Rightarrow$$

$$\|\mathbf{\Theta}^{-1}\|_2 \quad = \quad (\sigma_{k+1}(\mathbf{\Theta}))^{-1} \leq (k+1)\|\mathbf{\Theta}^{-1}\|_C \quad \Rightarrow$$

$$\|\mathbf{\Theta}^{-1}\|_C^{-1} \quad \leq \quad (k+1)\sigma_{k+1}(\mathbf{\Theta}) \quad \Rightarrow \quad \boxed{|\mathbf{\Gamma}| \leq (k+1)\sigma_{k+1}(\mathbf{\Theta})}$$

Example.

$$\mathbf{K} = \left[\begin{array}{cccc|c} 2 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 2 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \end{array}\right], \quad \mathbf{A} = \begin{bmatrix} 2 & 1 & 0 & 0 \\ 0 & 1 & 1 & 2 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 2 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix},$$

$\mathbf{D} = \mathbf{1}$,

$$\Rightarrow \quad \Gamma = \frac{1}{3}, \qquad \text{adj}\,(\mathbf{K}) = \begin{bmatrix} 1 & 0 & 1 & 1 & -2 \\ 1 & 2 & 1 & -1 & -2 \\ -2 & -2 & 0 & 0 & 4 \\ 1 & 2 & 1 & 1 & -4 \\ -1 & -2 & -3 & -1 & 6 \end{bmatrix}.$$

$$\begin{aligned} \mathbf{S} = \text{svd}\,(\mathbf{K}) &= [4.5224, \quad 2.3243, \quad 1.1997, \quad 0.8176, \quad 0.19398] \\ \text{error bound} &= 5 \cdot \mathbf{S}(1,5) = 0.96991. \end{aligned}$$

The rank 4 CUR factorization is:

$$\mathbf{K}_{\text{cur}} = \mathbf{K}(:, 1:4) \cdot (\mathbf{K}(1:4, 1:4))^{-1} \cdot \mathbf{K}(1:4, :) = \left[\begin{array}{cccc|c} 2 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 2 & 1 \\ \hline 1 & 1 & 1 & 1 & \frac{2}{3} \end{array}\right],$$

and the element-wise error is less than the error bound computed above, namely 0.96991.

The cross approximation algorithm. Given the matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$, let $\mathcal{J}_0$ be the indices of an initial choice of k columns. Let the QR factorization of these k columns be:

$$\mathbf{K}(:, \mathcal{J}_0) = Q_1 R_1, \quad Q_1 \in \mathbb{R}^{n \times k}, \quad R_1 \in \mathbb{R}^{k \times k}.$$

Then determine the k rows given by the index set $\mathcal{I}_1$, so that $|\det \mathbf{K}(I_1, J_0)|$ is maximum among all the $k \times k$ submatrices of Q_1:

$$\mathcal{I}_1 = \text{maxvol}(Q_1, \epsilon).$$

Next, we compute the QR factorization of $\mathbf{K}(\mathcal{I}_1, :)^T$:

$$\mathbf{K}(\mathcal{I}_1, :)^T = Q_2 R_2, \quad Q_2 \in \mathbb{R}^{n \times k}, \quad R_2 \in \mathbb{R}^{k \times k} \quad \Rightarrow \quad Q_2 = \mathbf{K}(\mathcal{I}_1, :)^T R_2^{-1}$$

and let

$$J_1 = \text{maxvol}(Q_2, \epsilon) \quad \text{and} \quad \hat{Q}_2 = Q_2(\mathcal{J}_1, :) = \mathbf{K}(\mathcal{I}_1, \mathcal{J}_1)^T R_2^{-1}$$

The resulting approximant is:

$$\boxed{\mathbf{K}_1} = \mathbf{K}(:, \mathcal{J}_1) \hat{Q}_2^{-T} Q_2^T = \boxed{\mathbf{K}(:, \mathcal{J}_1) \cdot [\mathbf{K}(\mathcal{I}_1, \mathcal{J}_1)]^{-1} \cdot \mathbf{K}(\mathcal{I}_1, :)}$$

Repeat, with the original $\mathcal{J}_0$ replaced by $\mathcal{J}_1$, until the stopping criterion below is met:

$$\frac{\|\mathbf{K}_{r+1} - \mathbf{K}_r\|_F}{\|\mathbf{K}_r\|_F} \leq \delta.$$

Examples.

```
>> %%%%%%%%%% CUR approximants of rank 2
>> %%%% Consider the 5x5 matrix
>>
>> K=round(2*rand(5,5)) =
     2     0     1     0     1
     2     2     0     1     1
     1     2     1     0     1
     1     2     1     1     0
     1     1     0     1     1
>> det(K) =     6
>> svd(K) = sv =
   5.2415e+00
   1.8968e+00
   1.3723e+00
   8.9721e-01
   4.9015e-01

>> %%%%% Cross-approximation algorithm
>>
>> J=[1 2];
>> I = maxvol(K(:,J),0.01) =     1     2
>> J=maxvol(K(I,:).',.01)  =     1     2
>> I=maxvol(K(:,J),.01)    =     1     2
>>
```

```
>> Kapp=K(:,J)*inv(K(I,J))*K(I,:) =>
>> K-Kapp =
[ 0, 0,   0,   0,    0]
[ 0, 0,   0,   0,    0]
[ 0, 0, 3/2,  -1,  1/2]
[ 0, 0, 3/2,   0, -1/2]
[ 0, 0,   0, 1/2,  1/2]
>> Upper_Bound_Of_Error = 3*sv(3,1) =   4.1169e+00

>> %%%%%%% Determining the max volume 2x2 sub matrix
>> I0=[];for k1=1:4; for k2=k1+1:5; I0=[I0;k1 k2];end;end;
>> Kd=zeros(10,10);for k1=1:10;for k2=1:10;Kd(k1,k2)=det(K(I0(k1,:),I0(k2,:)));end;
[  4, -2,  2,  0, -2,  0, -2,  1,  1, -1]
[  4,  1,  0,  1, -2,  0, -2,  0,  0,  0]
[  4,  1,  2, -1, -2,  0, -2,  1, -1, -1]
[  2, -1,  2,  1, -1,  0, -1,  1,  1, -1]
[  2,  2, -1,  1,  2, -2,  0, -1, -1,  1]
[  2,  2,  1, -1,  2,  0, -2, -1, -1, -1]
[  0,  0,  1,  1,  0,  1,  1,  0,  0,  0]
[  0,  0,  1, -1,  0,  2, -2,  1, -1, -1]
[ -1, -1,  1,  0, -1,  2,  1,  1,  1, -1]
[ -1, -1,  0,  1, -1,  1,  2,  1,  1,  1]
>> I1=I0(1,:) =    1     2
>> J1=I0(1,:) =    1     2
>> I2=I0(2,:) =    1     3
>> I3=I0(3,:) =    1     4
```

```
>> K1=K(:,J1)*inv(K(I1,J1))*K(I1,:) =
[ 2, 0,    1,   0,   1]
[ 2, 2,    0,   1,   1]
[ 1, 2, -1/2,   1, 1/2]
[ 1, 2, -1/2,   1, 1/2]
[ 1, 1,    0, 1/2, 1/2]
>> K2=K(:,J1)*inv(K(I2,J1))*K(I2,:) =
[ 2, 0,   1, 0,   1]
[ 2, 2, 3/2, 0, 3/2]
[ 1, 2,   1, 0,   1]
[ 1, 2,   1, 0,   1]
[ 1, 1, 3/4, 0, 3/4]
>> K3=K(:,J1)*inv(K(I3,J1))*K(I3,:) =
[ 2, 0,   1,   0,   1]
[ 2, 2, 3/2,   1, 1/2]
[ 1, 2,   1,   1,   0]
[ 1, 2,   1,   1,   0]
[ 1, 1, 3/4, 1/2, 1/4]
>> K-K1 =
[ 0, 0,   0,   0,    0]
[ 0, 0,   0,   0,    0]
[ 0, 0, 3/2,  -1,  1/2]
[ 0, 0, 3/2,   0, -1/2]
[ 0, 0,   0, 1/2,  1/2]
>> K-K2 =
[ 0, 0,    0, 0,    0]
```

```
[ 0, 0, -3/2, 1, -1/2]
[ 0, 0,    0, 0,    0]
[ 0, 0,    0, 1,   -1]
[ 0, 0, -3/4, 1,  1/4]
>> K-K3 =
[ 0, 0,    0,   0,   0]
[ 0, 0, -3/2,   0, 1/2]
[ 0, 0,    0,  -1,   1]
[ 0, 0,    0,   0,   0]
[ 0, 0, -3/4, 1/2, 3/4]
>> Kapp =
[ 2, 0,    1,   0,   1]
[ 2, 2,    0,   1,   1]
[ 1, 2, -1/2,   1, 1/2]
[ 1, 2, -1/2,   1, 1/2]
[ 1, 1,    0, 1/2, 1/2]
>> Kapp-K1 =
[ 0, 0, 0, 0, 0]
[ 0, 0, 0, 0, 0]
[ 0, 0, 0, 0, 0]
[ 0, 0, 0, 0, 0]
[ 0, 0, 0, 0, 0]
>> Kapp-K2 =
[ 0, 0,    0,   0,    0]
[ 0, 0, -3/2,   1, -1/2]
[ 0, 0, -3/2,   1, -1/2]
```

```
[ 0, 0, -3/2,   1, -1/2]
[ 0, 0, -3/4, 1/2, -1/4]
>> Kapp-K3 =
[ 0, 0,    0, 0,   0]
[ 0, 0, -3/2, 0, 1/2]
[ 0, 0, -3/2, 0, 1/2]
[ 0, 0, -3/2, 0, 1/2]
[ 0, 0, -3/4, 0, 1/4]
>>
%%%%%%%% CUR approximates of rank 2 of a 10x10 matrix
>>
>> K=round(2*rand(10,10)) =
     1     1     2     0     1     0     2     1     1     1
     1     2     1     1     2     0     1     1     1     1
     1     1     2     0     0     1     2     1     1     1
     1     0     1     1     1     1     0     1     2     2
     2     1     2     2     1     0     1     2     1     1
     1     0     0     1     1     1     0     0     1     1
     2     1     1     2     2     1     1     1     1     0
     1     1     1     1     1     0     1     0     2     2
     1     1     1     0     2     1     1     0     2     1
     0     1     1     1     1     0     1     1     1     1
>> det(K) =  -4
>> svd(K)' =   1.0423e+01   3.1581e+00   2.8614e+00   2.2472e+00   1.9109e+00
               1.0674e+00   8.6656e-01   6.5080e-01   4.2707e-01   3.8469e-02
>> %%% Determine the maxvol 2x2 sub matrices
```

```
>> I0=[];for k1=1:9; for k2=k1+1:10; I0=[I0;k1 k2];end;end;
>> size(I0) =    45     2
>> Kd=zeros(45,45);for k1=1:45;for k2=1:45;Kd(k1,k2)=det(K(I0(k1,:),I0(k2,:)));end;
>> max(max(abs(Kd))) =     4
>> Er=[];for k1=1:45;for k2=1:45; if abs(Kd(k1,k2))==4;Er=[Er; k1 k2];end;end;end =
     3    41
     3    42
     4    18
     4    27
     6    18
     6    27
    10    19
    10    32
    11    16
    11    17
    18    41
    18    42
    19    18
    19    27
    21    18
    21    19
    21    27
    21    32
    23    19
    23    32
    27     9
```

```
    27    30
    27    35
    33    43
    33    44
    34    25
    34    29
    34    33
    34    43
    40     9
    40    30
    40    35
    41    25
    41    29  ==>  there are 34 2x2 maxvol sub matrices
>>
>> %%% for instance
>> I=I0(23,:);J=I0(19,:);
>> Kapp=K(:,J)*inv(K(I,J))*K(I,:);
>> K-Kapp =
[ -1/4, -1/4, 0, 0, 0, -5/4,  0,  1/4, -3/4, -1/4]
[    0,    1, 0, 1, 0,   -1,  0,    1,   -1,    0]
[    0,    0, 0, 0, 0,    0,  0,    0,    0,    0]
[  1/4, -3/4, 0, 1, 0,  1/4, -1,  3/4,  3/4,  5/4]
[  3/4, -1/4, 0, 2, 0, -5/4, -1,  5/4, -3/4, -1/4]
[  3/4, -1/4, 0, 1, 0,  3/4,  0,  1/4,  1/4,  3/4]
[    1,    0, 0, 2, 0,    0,  0,    1,   -1,   -1]
[  1/4,  1/4, 0, 1, 0, -3/4,  0, -1/4,  3/4,  5/4]
```

```
[    0,    0, 0, 0, 0,    0,  0,    0,    0,    0]
[ -3/4,  1/4, 0, 1, 0, -3/4,  0,  3/4, -1/4,  1/4]
>> I0(23,:) =     3     9
>> I0(19,:) =     3     5
>> max(max(abs(sym(K-Kapp)))) =  2
>> sv(1,3)*3 =   8.5842e+00
>> %%% Cross-approximation algorithm
>> I=[8 10];
>> J=maxvol(K(:,I),.01) =    5     8
>> I=maxvol(K(:,J),.01) =    9     5
>> J=maxvol(K(:,I),.01) =    4     2
>> I=maxvol(K(:,J),.01) =    5     2
>> J=maxvol(K(:,I),.01) =    2     3
>> I=maxvol(K(:,J),.01) =    2     1
>> J=maxvol(K(:,I),.01) =    2     5
>> I=maxvol(K(:,J),.01) =    2     3
>> J=maxvol(K(:,I),.01) =    2     1
>> I=maxvol(K(:,J),.01) =    2     5
>> J=maxvol(K(:,I),.01) =    2     3
>> I=maxvol(K(:,J),.01) =    2     1
>> Kapp1=sym(K(:,J)*inv(K(I,J))*K(I,:));
>> K-Kapp1 =
[    0, 0, 0,    0,  0, 0,  0,    0,   0,    0]
[    0, 0, 0,    0,  0, 0,  0,    0,   0,    0]
[    0, 0, 0,    0, -1, 1,  0,    0,   0,    0]
[  2/3, 0, 0,  4/3,  1, 1, -1,  2/3, 5/3,  5/3]
```

```
[    1, 0, 0,    2,  0, 0, -1,    1,   0,    0]
[    1, 0, 0,    1,  1, 1,  0,    0,   1,    1]
[  4/3, 0, 0,  5/3,  1, 1,  0,  1/3, 1/3, -2/3]
[  1/3, 0, 0,  2/3,  0, 0,  0, -2/3, 4/3,  4/3]
[  1/3, 0, 0, -1/3,  1, 1,  0, -2/3, 4/3,  1/3]
[ -2/3, 0, 0,  2/3,  0, 0,  0,  1/3, 1/3,  1/3]
>> max(max(abs(ans))) = 2
>> Kapp-Kapp =
[  1/4,  1/4, 0,    0,  0, 5/4, 0,  -1/4,   3/4,  1/4]
[    0,   -1, 0,   -1,  0,   1, 0,    -1,     1,    0]
[    0,    0, 0,    0, -1,   1, 0,     0,     0,    0]
[ 5/12,  3/4, 0,  1/3,  1, 3/4, 0, -1/12, 11/12, 5/12]
[  1/4,  1/4, 0,    0,  0, 5/4, 0,  -1/4,   3/4,  1/4]
[  1/4,  1/4, 0,    0,  1, 1/4, 0,  -1/4,   3/4,  1/4]
[  1/3,    0, 0, -1/3,  1,   1, 0,  -2/3,   4/3,  1/3]
[ 1/12, -1/4, 0, -1/3,  0, 3/4, 0, -5/12,  7/12, 1/12]
[  1/3,    0, 0, -1/3,  1,   1, 0,  -2/3,   4/3,  1/3]
[ 1/12, -1/4, 0, -1/3,  0, 3/4, 0, -5/12,  7/12, 1/12]
>> det(K(I,J)) =     3
```

Summary. The CUR factorization of rank k a matrix $A \in \mathbb{R}^{n \times m}$ is defined as:

$$\boxed{A_{\text{cur}} = A(:, J) \cdot [A(I, J)]^{-1} \cdot A(I, :)}$$

where I and J k column and row indices respectively. The problem consists in appropriately determining these indices. Here are some ways of choosing I and J.

1. Exhaustive search for the maximum volume $k \times k$ submatrix $A(I, J)$ of A. The complexity of this direct approach is $\frac{n!}{(n-k)!k!} \cdot \frac{m!}{(m-k)!k!}$, which is prohibitive for even moderate n, m.
2. Cross-Approximation algorithm. Pick an initial choice of k columns, say $I_0 = [1 \;\; 2 \;\; \cdots \;\; k]$. Find the rows indexed by J_1 of these k rows that maximize the volume of $A(J_1, I_0)$. Then find the columns indexed by I_1 which maximize the volume $A(J_1, I_1)$. And so on, until convergence.
3. By means of the SVD. Let $[U, S, V] = svd(A)$. Find k rows of $U(:, 1:k)$ indexed by I which maximize the volume of $U(I, 1:k)$. Then find k rows of $V(:, 1:k)$ indexed by J such that the volume of $V(J, 1:k)$ is maximized.
4. As in the previous case but the choice of rows indexed by I and of the columns indexed by J is done by means of the DEIM (Discrete-Empirical Interpolation Method) procedure (which as discussed involves an oblique projection).
5. Cross-Approximation combined with DEIM.

References.

S.A. Goreinov and E.E. Tyrtyshnikov, The Maximal-Volume Concept in Approximation by Low-Rank Matrices, Contemporary Mathematics, Volume 280: 47-51, (2001).

S.A. Goreinov, E.E. Tyrtyshnikov, and N.L. Zamarashkin, A Theory of Pseudoskeleton Approximations, Linear Algebra and Its Applications, vol 261: 1-21 (1997).

S. A. Goreinov, I. V. Oseledets, D. V. Savostyanov, E. E. Tyrtyshnikov, N. L. Zamarashkin, How to find a good submatrix, Institute for Computational Mathematics Hong Kong Baptist University, ICM Research Report 08-10, 2008.

D.C. Sorensen and M. Embree, A DEIM induced CUR factorization, SIAM J. Scientific Computing, Vol. 38, No. 3, pp. A1454-A1482 (2016).

P. Drineas, M. W. Mahoney, and S. Muthukrishnan, Relative-error CUR matrix decompositions, SIAM J. Matrix Anal. Appl., 30: 844-881 (2008).

The google matrix and the PageRank

A matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$ is called **positive** (**non-negative**) if $\mathbf{M}_{i,j} > 0$, ($\mathbf{M}_{i,j} \geq 0$) for all i and j.

A non-negative matrix square $\mathbf{M}$ is called **primitive** if there is a positive integer k such that all entries of $\mathbf{M}^k$ are positive. It is called **irreducible**, if for any indices i, j, there is a positive integer $k = k(i,j)$ such that $\mathbf{M}^k_{i,j} > 0$.

Proposition. If $\mathbf{M}$ is irreducible then $\mathbf{I} + \mathbf{M}$ is primitive.

Theorem. Oscar Perron (1907).

Given a positive matrix $\mathbf{M}$ the following hold.

- $\mathbf{M}$ has a positive eigenvalue $\lambda > 0$.

  This eigenvalue has geometric and algebraic multiplicity equal to one.
- For all other eigenvalues μ of $\mathbf{M}$ there holds $\lambda > |\mu|$.
- The left and right eigenvectors corresponding to λ can be chosen

  with positive entries.

Theorem. Ferdinand Georg Frobenius (1912).

Perron's result holds as well, for irreducible matrices $\mathbf{M}$.

Example. Consider

$$\mathbf{M}=\begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \Rightarrow \mathbf{M}^2=\begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 2 & 1 & 2 & 1 & 2 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 2 & 2 & 2 & 3 & 3 & 4 \\ 1 & 1 & 0 & 0 & 0 & 2 \end{bmatrix} \Rightarrow$$

$$\mathbf{M}^4=\begin{bmatrix} 8 & 10 & 7 & 12 & 10 & 17 \\ 4 & 6 & 4 & 5 & 5 & 9 \\ 6 & 9 & 6 & 9 & 8 & 14 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 14 & 18 & 12 & 20 & 17 & 30 \\ 3 & 4 & 2 & 2 & 3 & 7 \end{bmatrix} \Rightarrow \mathbf{M}^{10}=\begin{bmatrix} 2186 & 2929 & 1894 & 2934 & 2638 & 4823 \\ 1150 & 1542 & 997 & 1541 & 1388 & 2538 \\ 1779 & 2385 & 1542 & 2385 & 2147 & 3926 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 3788 & 5076 & 3282 & 5081 & 4571 & 8358 \\ 744 & 997 & 644 & 992 & 897 & 1642 \end{bmatrix}$$

Conclusion. The matrix $\mathbf{M}$ above is non-negative, but **neither** irreducible **nor** primitive.

The EVD of $\mathbf{M}$ is: $\mathbf{M} = \mathbf{T}\mathbf{\Lambda}\mathbf{T}^{-1}$:

$$\mathbf{T} = \begin{array}{|c|c|c|c|c|c|} \hline 0.4446 & 0.3015 & -0.0019 - 0.4129i & -0.0019 + 0.4129i & 0.5773 & -0.5773 \\ \hline 0.2339 & -0.6030 & -0.4032 - 0.1785i & -0.4032 + 0.1785i & 0 & 0 \\ \hline 0.3619 & -0.6030 & 0.6141 & 0.6141 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0.7705 & 0.3015 & -0.3807 + 0.1118i & -0.3807 - 0.1118i & 0.5773 & -0.5773 \\ \hline 0.1512 & 0.3015 & 0.2128 + 0.2344i & 0.2128 - 0.2344i & -0.5773 & 0.5773 \\ \hline \end{array}$$

$$\text{diag}\,\mathbf{\Lambda} = [2.5468, \quad -1, \quad -0.2734 + 0.5638i, \quad -0.2734 - 0.5638i, \quad 1, \quad 1]$$

The set of left eigenvectors (obtained from the EVD of $\mathbf{M}^T$) are

0	−0.2927	−0.6070	−0.6070	0.4850	0
0	−0.3923	−0.2609 − 0.3112i	−0.2609 + 0.3112i	−0.4850	0
0	−0.2536	0.0808 + 0.2802i	0.0808 − 0.2802i	−0.4850	0
1	−0.3923	−0.2609 − 0.3112i	−0.2609 + 0.3112i	0.2425	1
0	−0.3532	0.4269 − 0.0310i	0.4269 + 0.0310i	0	0
0	−0.6459	−0.1800 − 0.0310i	−0.1800 + 0.0310i	0.4850	0

corresponding to the following ordering of the eigenvalues:

$$1, \quad 2.5468, \quad -0.27341 + 0.56382i, \quad -0.27341 - 0.56382i, \quad -1, \quad 1.$$

Remark. **(a)** the largest in magnitude eigenvalue has multiplicity one and the entries of the right/left eigenvectors have the same sign.

(b) The eigenvalue 1 has algebraic multiplicity equal to 2 but geometric multiplicity equal to 1. Thus (up to a scalar) the right eigenvector and generalized eigenvector are

$$[1,\ \ 0,\ \ 0,\ \ 0,\ \ 1,\ \ -1]^T,\ \ [1,\ \ 1,\ \ -1,\ \ -2,\ \ 1,\ \ 0,\ \ -1]^T,$$

while the corresponding left eigenvector and generalized eigenvector are:

$$[0,\ \ 0,\ \ 0,\ \ 1,\ \ 0,\ \ 0],\ \ [1,\ \ 3,\ \ -1,\ \ 0,\ \ -2,\ \ 5]$$

Example. Consider the non-negative matrix

$$\mathbf{M}=\begin{bmatrix}0&1&0&0&0&0\\0&1&1&0&0&0\\0&1&1&0&0&1\\0&0&1&0&0&1\\1&1&1&1&0&0\\1&1&1&0&1&0\end{bmatrix},\mathbf{M}^2=\begin{bmatrix}0&1&1&0&0&0\\0&2&2&0&0&1\\1&3&3&0&1&1\\1&2&2&0&1&1\\0&3&3&0&0&2\\1&4&3&1&0&1\end{bmatrix},\mathbf{M}^3=\begin{bmatrix}0&2&2&0&0&1\\1&5&5&0&1&2\\2&9&8&1&1&3\\2&7&6&1&1&2\\2&8&8&0&2&3\\1&9&9&0&1&4\end{bmatrix},$$

$$\mathbf{M}^4=\begin{bmatrix}1&5&5&0&1&2\\3&14&13&1&2&5\\4&23&22&1&3&9\\3&18&17&1&2&7\\5&23&21&2&3&8\\5&24&23&1&4&9\end{bmatrix},\ \mathbf{M}^5=\begin{bmatrix}3&14&13&1&2&5\\7&37&35&2&5&14\\12&61&58&3&9&23\\9&47&45&2&7&18\\11&60&57&3&8&23\\13&65&61&4&9&24\end{bmatrix}.$$

Since $\mathbf{M}^5 > 0$ is positive, $\mathbf{M}$ is **primitive**. Therefore the Perron-Frobenius results apply.

The characteristic polynomial of $\mathbf{M}$ is $s^3(s^3-2s^2-s-2)$, which means that the zero eigenvalue has algebraic multiplicity 3.

As it turns out the geometric multiplicity of this eigenvalue is 1. Hence the matrix contains a 3×3 Jordan block of the form $\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$. The EVD is $\mathbf{M} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$, is

$$\text{diag}\left(\mathbf{\Lambda}\right) = \begin{bmatrix} 2.659 + 0i \\ -0.32948 + 0.80225i \\ -0.32948 - 0.80225i \\ 6.4572e-06 + 0i \\ -3.2286e-06 + 5.5921e-06i \\ -3.2286e-06 - 5.5921e-06i \end{bmatrix} \quad \text{and}$$

0.1121	$0.1822 + 0.1762i$	$0.1822 - 0.1762i$	-0.5773	0.5773	0.5773
0.2981	$-0.2014 + 0.0881i$	$-0.2014 - 0.0881i$	$-3.7 \cdot 10^{-6}$	$(-1.8 + 3.2i) \cdot 10^{-6}$	$(-1.8 - 3.228i) \cdot 10^{-6}$
0.4946	$0.1971 - 0.2788i$	$0.1971 + 0.2780i$	$3.7 \cdot 10^{-6}$	$(1.8 - 3.2i) \cdot 10^{-6}$	$(1.864 + 3.228i) \cdot 10^{-6}$
0.3825	$0.0148 - 0.4551i$	$0.0148 + 0.4551i$	0.5773	$-0.5773 - 3.2 \cdot 10^{-6}i$	$-0.5773 + 3.228 \cdot 10^{-6}i$
0.4842	-0.5852	-0.5852	0.5773	$-0.5773 + 2.0 \cdot 10^{-11}i$	$-0.5773 - 2.0 \cdot 10^{-11}i$
0.5224	$0.1630 + 0.4407i$	$0.1630 - 0.4407i$	$4.8 \cdot 10^{-11}$	$(2.4 + 4.1i) \cdot 10^{-11}$	$(2.4 - 4.1i) \cdot 10^{-11}$

while the left eigenvector matrix (with the same ordering of eigenvalues) is

0.1343	0.5188	0.5188	0.5773	-0.5773	-0.5773
0.6903	$0.0939 - 0.3097i$	$0.0939 + 0.3097i$	$5.6 \cdot 10^{-6}$	$(2.8 + 4.9i) \cdot 10^{-6}i$	$(2.8 - 4.9i) \cdot 10^{-6}i$
0.6535	$-0.2242 + 0.0709i$	$-0.2242 - 0.0709i$	-0.5773	$0.5773 - 2.4 \cdot 10^{-6}i$	$0.5773 + 2.4 \cdot 10^{-6}i$
0.0367	$0.3182 - 0.3807i$	$0.3182 + 0.3807i$	0.5773	$-0.5773 + 2.4 \cdot 10^{-6}i$	$-0.5773 - 2.4 \cdot 10^{-6}i$
0.0976	$0.2006 + 0.3807i$	$0.2006 - 0.3807i$	$-2.8 \cdot 10^{-6}$	$(-1.4 - 2.4i) \cdot 10^{-6}$	$(-1.4 + 2.4i) \cdot 10^{-6}i$
0.2596	$-0.3715 + 0.0354i$	$-0.3715 - 0.0354i$	$1.3 \cdot 10^{-11}$	$(6.9 - 1.2i) \cdot 10^{-11}$	$(6.9 + 1.2i) \cdot 10^{-11}i$

Markov chains in a nutshell

A non-negative matrix $\mathbf{M}$ is **(column) stochastic**, if the elements of each **column** sum to 1.

- Then the row vector $\mathbb{I}^T$ all of whose entries equal 1, is a left eigenvector with eigenvalue 1.
- If $\mathbf{M}$ is irreducible, 1 is the maximal eigenvalue.
- If $\mathbf{M}$ is primitive, then from the general theory follows that

$$\mathbf{M}^k = \begin{bmatrix} \pi_1 & \pi_1 & \cdots & \pi_1 \\ \pi_2 & \pi_2 & \cdots & \pi_2 \\ \vdots & \vdots & \cdots & \vdots \\ \pi_n & \pi_n & \cdots & \pi_n \end{bmatrix},$$

for k large enough, where

$$\pi := [\pi_1, \ \pi_2, \ \cdots, \ \pi_n]^T,$$

is the unique vector whose entries sum to one and satisfies

$$\mathbf{M}\pi = \pi.$$

Connectivity of the 6-node graph.

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 1 \\ 0 & 0 & \frac{1}{3} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix}, \quad \mathbf{A}_0 = \begin{bmatrix} 0 & \frac{1}{6} & \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{6} & \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{6} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{6} & 0 & 0 & \frac{1}{2} & 1 \\ 0 & \frac{1}{6} & \frac{1}{3} & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{6} & 0 & \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix}.$$

```
>> [diag(dA) diag(dA0)] =
           0 +           0i           1 +           0i
           1 +           0i     0.67787 +           0i
     0.40825 +           0i    -0.099542 +          0i
    -0.40825 +           0i     -0.41166 +          0i
        -0.5 +           0i         -0.5 + 1.2092e-08i
        -0.5 +           0i         -0.5 - 1.2092e-08i

>> [uA(:,2) uA0(:,1)] =
  -8.9023e-16   3.3379e-16
  -4.2773e-16  -5.4448e-17
  -7.2745e-16  -2.0314e-16
     -0.74278      0.74278
     -0.37139      0.37139
     -0.55709      0.55709
```

$$\mathbf{A}_0^{10} = \begin{array}{|c|c|c|c|c|c|}\hline 0.00711 & 0.00407 & 0.00540 & 0 & 0 & 0 \\ \hline 0.01221 & 0.00708 & 0.00950 & 0 & 0 & 0 \\ \hline 0.00811 & 0.00475 & 0.00642 & 0 & 0 & 0 \\ \hline 0.43248 & 0.43652 & 0.43050 & 0.44824 & 0.44727 & 0.4375 \\ \hline 0.21721 & 0.21999 & 0.22197 & 0.21875 & 0.21973 & 0.22852 \\ \hline 0.32287 & 0.32757 & 0.32619 & 0.33301 & 0.33301 & 0.33398 \\ \hline \end{array},$$

$$\mathbf{A}_0^{100} = \begin{array}{|c|c|c|c|c|c|}\hline 4.4e-18 & 2.5e-18 & 3.4e-18 & 0 & 0 & 0 \\ \hline 7.7e-18 & 4.5e-18 & 6.0e-18 & 0 & 0 & 0 \\ \hline 5.2e-18 & 3.0e-18 & 4.0e-18 & 0 & 0 & 0 \\ \hline 0.44444 & 0.44444 & 0.44444 & 0.44444 & 0.44444 & 0.44444 \\ \hline 0.22222 & 0.22222 & 0.22222 & 0.22222 & 0.22222 & 0.22222 \\ \hline 0.33333 & 0.33333 & 0.33333 & 0.33333 & 0.33333 & 0.33333 \\ \hline \end{array},$$

$$\mathbf{A}^{100} = \begin{array}{|c|c|c|c|c|c|}\hline 1.2e-39 & 0 & 0 & 0 & 0 & 0 \\ \hline 1.2e-39 & 0 & 1.2e-39 & 0 & 0 & 0 \\ \hline 0 & 0 & 1.2e-39 & 0 & 0 & 0 \\ \hline 0.088889 & 0 & 0.17778 & 0.44444 & 0.44444 & 0.44444 \\ \hline 0.044444 & 0 & 0.08888 & 0.22222 & 0.22222 & 0.22222 \\ \hline 0.066667 & 0 & 0.13333 & 0.33333 & 0.33333 & 0.33333 \\ \hline \end{array}.$$

$$\left.\begin{array}{r} \mathbf{A}_1 = \alpha \cdot \mathbf{A}_0 + (1-\alpha) \cdot \frac{1}{6} \cdot \text{ones}\,(6,6) \\ \text{where} \quad \alpha = 0.9 \end{array}\right\} = \begin{bmatrix} \frac{1}{60} & \frac{1}{6} & \frac{19}{60} & \frac{1}{60} & \frac{1}{60} & \frac{1}{60} \\ \frac{7}{15} & \frac{1}{6} & \frac{19}{60} & \frac{1}{60} & \frac{1}{60} & \frac{1}{60} \\ \frac{7}{15} & \frac{1}{6} & \frac{1}{60} & \frac{1}{60} & \frac{1}{60} & \frac{1}{60} \\ \frac{1}{60} & \frac{1}{6} & \frac{1}{60} & \frac{1}{60} & \frac{7}{15} & \frac{11}{12} \\ \frac{1}{60} & \frac{1}{6} & \frac{19}{60} & \frac{7}{15} & \frac{1}{60} & \frac{1}{60} \\ \frac{1}{60} & \frac{1}{6} & \frac{1}{60} & \frac{7}{15} & \frac{7}{15} & \frac{1}{60} \end{bmatrix}.$$

$\lambda(\mathbf{A})$	$\lambda(\mathbf{A}_0)$	$\lambda(\mathbf{A}_1)$	$\mathbf{u}_{\mathbf{A}_1}(:,1)$
0	1	1	0.07147
1	0.67787	0.61009	0.10363
0.40825	−0.09954	−0.08958	0.07971
−0.40825	−0.41166	−0.37050	0.72041
−0.5	$-0.5 + 1.2e-08i$	$-0.45 + 1.2e-08i$	0.39566
−0.5	$-0.5 - 1.2e-08i$	$-0.45 - 1.2e-08i$	0.54979

The eigenvalues of the Google matrix

Consider a graph connectivity matrix $\mathbf{C} \in \mathbb{R}^{n \times n}$ and the associated Google matrix $\mathbf{G}$:

$$\mathbf{G} = \alpha \mathbf{C} + (1 - \alpha) \frac{1}{n} \mathbb{I} \cdot \mathbb{I}^T$$

where $0 \leq \alpha \leq 1$ and $\mathbb{I}^T = [1, \ 1, \ \cdots, \ 1] \in \mathbb{R}^{1 \times n}$.

Recall: $\mathbf{C}$ is a column stochastic matrix, i.e. the entries of each column sum to 1. Thus the biggest (in magnitude) eigenvalue of $\mathbf{C}$ is 1 and $\mathbb{I}^T$ is the associated left eigenvector.

Let the matrix of all left/right eigenvectors be: $\mathbf{W}, \mathbf{V} \in \mathbb{C}^{n \times n}$. Assume further that the first row/column of $\mathbf{V}$, $\mathbf{W}$ is the left/right eigenvector of $\mathbf{C}$ associated with the eigenvalue 1:

$$\mathbf{W} = \left[\begin{array}{c} \mathbb{I}^T \\ \hline \mathbf{X} \end{array} \right], \quad \mathbf{V} = \left[\begin{array}{c|c} \boldsymbol{\pi} & \mathbf{Y} \end{array} \right] \quad \text{where} \quad \mathbf{WV} = \mathbf{I}_n.$$

Moreover $\boldsymbol{\pi}$ is normalized so that the sum of its entries is one, i.e. it is the **page rank**. Thus

$$\mathbb{I}^T \boldsymbol{\pi} = 1, \quad \mathbb{I}^T \mathbf{Y} = \mathbf{0}^T, \quad \mathbf{X}\boldsymbol{\pi} = \mathbf{0}, \quad \mathbf{X}\,\mathbf{Y} = \mathbf{I}_{n-1} \qquad \Rightarrow \qquad \mathbf{WCV} = \left[\begin{array}{c|c} 1 & \mathbf{0}^T \\ \hline \mathbf{0} & \mathbf{XCY} \end{array}\right].$$

Consequently, the eigenvalues of $\mathbf{C}$ (in descending magnitude) are $1, \lambda_2, \lambda_3, \cdots, \lambda_n$, and those of $\mathbf{XCY} \in \mathbb{C}^{(n-1)\times(n-1)}$, are $\lambda_2, \cdots, \lambda_n$.

Let us now consider $\frac{1}{n}\mathbf{W}\,(\mathbb{I}\cdot\mathbb{I})\,\mathbf{V}$:

$$\frac{1}{n}\mathbf{W}\left(\mathbb{I}\cdot\mathbb{I}^T\right)\mathbf{V} = \frac{1}{n}\mathbf{W}\,\mathbb{I}\,\left[1, \quad \mathbf{0}^T\right] = \frac{1}{n}\left[\begin{array}{c} n \\ \hline \mathbf{X}\,\mathbb{I} \end{array}\right]\left[1, \quad \mathbf{0}^T\right] = \left[\begin{array}{c|c} 1 & \mathbf{0}^T \\ \hline \frac{1}{n}\mathbf{X}\,\mathbb{I} & \mathbf{0}_{n-1} \end{array}\right].$$

Thus

$$\begin{aligned} \mathbf{WGV} &= \alpha\,\mathbf{WCV} + (1-\alpha)\frac{1}{n}\mathbf{W}\,\mathbb{I}\cdot\mathbb{I}^T\,\mathbf{V} \\ &= \alpha\left[\begin{array}{c|c} 1 & \mathbf{0}^T \\ \hline \mathbf{0} & \mathbf{XCY} \end{array}\right] + (1-\alpha)\left[\begin{array}{c|c} 1 & \mathbf{0}^T \\ \hline \frac{1}{n}\mathbf{X}\,\mathbb{I} & \mathbf{0}_{n-1} \end{array}\right] = \left[\begin{array}{c|c} 1 & \mathbf{0}^T \\ \hline \frac{1-\alpha}{n}\mathbf{X}\,\mathbb{I} & \alpha\,\mathbf{XCY} \end{array}\right] \end{aligned}$$

Since the transformed matrix is block diagonal its eigenvalues are 1 and those of the second block on the diagonal, namely $\alpha\mathbf{XCY}$, in other words $\alpha\lambda_2, \alpha\lambda_3, \cdots, \alpha\lambda_n$.

Conclusion. The parameter α in the Google matrix **G** serves two purposes:

(a) It makes **G** a **positive** matrix (**C** is very sparse, i.e. it has lots of zeros), and

(b) It enhances the **convergence** of the power iteration method by increasing the gap between the first and the second eigenvalues.

(c) Finally, the **page rank** becomes: $\frac{1-\alpha}{n}(\mathbf{I}_6 - \alpha\mathbf{C})^{-1}\mathbb{I}$.

References.

1. Cleve Moler, The Worlds Largest Matrix Computation, Cleve's Corner Collection, The MathWorks Technical Articles (2002).

2. Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd, The PageRank Citation Ranking: Bringing Order to the Web, Technical Report, Stanford InfoLab (1999).

Outline

The power iteration

The simple vector iteration (power method)

Given is the diagonalizable matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$. Let $\mathbf{v}^{(0)}$ be any vector with $\|\mathbf{v}^{(0)}\| = 1$. Repeat the following steps:

$$\mathbf{x} = \mathbf{A}\mathbf{v}^{(k)} \quad \text{and} \quad \mathbf{v}^{(k+1)} = \frac{\mathbf{x}}{\|\mathbf{x}\|}.$$

Proposition. Assume that $\mathbf{A}$ has a simple largest eigenvalue λ_1 and let $\mathbf{v}_1$ be the corresponding eigenvector. Let $\mathbf{v}^{(0)}$ be any vector that has a non-zero component in the direction of $\mathbf{v}_1$. The simple iteration described above converges towards the dominant eigenvector and the angle between the k^{th} iterate $\mathbf{v}^{(k)}$ and $\mathbf{v}_1$, is of the order $\mathcal{O}\left(\frac{|\lambda_2|^k}{|\lambda_1|^k}\right)$.

Remarks.

(a) The smaller the ratio $\frac{|\lambda_2|}{|\lambda_1|}$, the faster the convergence.

(b) The result holds even if the matrix is not diagonalizable but $\lambda_1 \neq \lambda_2$.

(c) If the initial vector $\mathbf{v}^{(0)}$ does not have a component in the direction of $\mathbf{v}_1$, convergence is towards $\mathbf{v}_2$, assuming that $|\lambda_2| \neq |\lambda_3|$.

(d) The algorithm does not converge if $|\lambda_1| = |\lambda_2|$, but $\lambda_1 \neq \lambda_2$. Example: $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$. ■

The basic Krylov iteration. Given a real $n \times n$ matrix $\mathbf{A}$ and an n-vector $\mathbf{b}$, let $\mathbf{v}_1 = \frac{\mathbf{b}}{\|\mathbf{b}\|}$. At the k^{th} step we have

$$\mathbf{A}\mathbf{V}_k = \mathbf{V}_k\mathbf{H}_k + \mathbf{f}_k\mathbf{e}_k^*,\ \mathbf{V}_k \in \mathbb{R}^{n\times k},\ \mathbf{H}_k \in \mathbb{R}^{k\times k},\ \mathbf{f}_k \in \mathbb{R}^n,\ \mathbf{e}_k \in \mathbb{R}^k,$$

where $\mathbf{e}_k$ is the k^{th} canonical unit vector in $\mathbb{R}^k$, $\mathbf{V}_k = [\mathbf{v}_1\ \cdots\ \mathbf{v}_k]$ consists of k column vectors which are orthonormal, $\mathbf{V}_k^*\mathbf{V}_k = \mathbf{I}_k$, and $\mathbf{A}$ projected onto the subspace spanned by the columns of $\mathbf{V}_k$ is $\mathbf{H}_k = \mathbf{V}_k^*\mathbf{A}\mathbf{V}_k$; these conditions imply that $\mathbf{v}_{j+1} = \frac{\mathbf{f}_j}{\|\mathbf{f}_j\|}$, $j = 1, \cdots, n-1$.

Two algorithms fall under this umbrella, namely the **Lanczos algorithm** and the **Arnoldi algorithm**.

For arbitrary $\mathbf{A}$, the Krylov iteration is known as the *Arnoldi iteration*; in this case $\mathbf{H}_k$ is *upper Hessenberg*. For symmetric $\mathbf{A} = \mathbf{A}^*$, it is known as the symmetric or one-sided *Lanczos iteration*, in which case $\mathbf{H}_k$ is tridiagonal and symmetric. A variant involving two staring vectors can be applied to non-symmetric matrices $\mathbf{A}$ and is known as the two-sided or non-symmetric Lanczos iteration. In this case the projected matrix $\mathbf{H}_k$ is tridiagonal (but not symmetric).

Three uses of the Krylov iteration. The iteration described above has three main uses.

(a) *Iterative solution of* $\mathbf{Ax} = \mathbf{b}$. In this case we seek to approximate the solution $\mathbf{x}$ in an iterative fashion. The Krylov methods are based on the fact that successive approximants belong the subspaces $\mathcal{K}_i$ mentioned above. Both the Arnoldi and the one-sided Lanczos algorithms, construct iteratively orthonormal bases for these subspaces.

(b) *Iterative approximation of the eigenvalues of* $\mathbf{A}$. In this case $\mathbf{b}$ is not apriori fixed. The goal is to use the eigenvalues of the projected matrix $\mathbf{H}_k$ as approximants of the dominant eigenvalues of $\mathbf{A}$. The most simple-minded approach to the approximation of eigenvalues is the *power method*, where given $\mathbf{b}$, successive terms $\mathbf{A}^{k-1}b$ are computed. To overcome the slow convergence of this method, *Krylov methods* are used, where at the k^{th} step one makes use of the information contained in the whole sequence $\mathbf{b}$, $\mathbf{Ab}$, $\cdots$, $\mathbf{A}^{k-1}\mathbf{b}$.

(c) *Approximation of linear systems by moment matching*.

Krylov methods have their origins in eigenvalue **computations** and eigenvalue **estimations**. ■

Krylov methods

Given $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{b} \in \mathbb{R}^n$, let $\mathcal{R}_k(\mathbf{A}, \mathbf{b}) \in \mathbb{R}^{n\times k}$ be the k$^{\text{th}}$ controllability or Krylov matrix.

Problem.

Devise a process which is iterative and at the k^{th} step gives

$$\mathbf{A}\mathbf{V}_k = \mathbf{V}_k\mathbf{H}_k + \mathbf{R}_k, \quad \mathbf{V}_k,\ \mathbf{R}_k \in \mathbb{R}^{n\times k}, \quad \mathbf{H}_k \in \mathbb{R}^{k\times k}, \quad k = 1, 2, \cdots, n.$$

These quantities have to satisfy the following conditions at each step.

1. The columns of $\mathbf{V}_k$ are orthonormal: $\mathbf{V}_k^*\mathbf{V}_k = \mathbf{I}_k,\ k = 1, 2, \cdots, n.$
2. The residual $\mathbf{R}_k$ is orthogonal to the columns of $\mathbf{V}_k$, that is, it satisfies the **Galerkin condition**: $\mathbf{V}_k^*\mathbf{R}_k = \mathbf{0},\ k = 1, 2, \cdots, n.$
3. $\text{span col}\,\mathbf{V}_k = \text{span col}\,\mathcal{R}_k(\mathbf{A}, \mathbf{b}),\ k = 1, 2, \cdots, n.$

This problem leads to the **Lanczos and Arnoldi procedures**.

The Lanczos method

Let $\mathbf{A} = \mathbf{A}^T$ together with a sequence of vectors $\mathbf{v}_1, \cdots, \mathbf{v}_k$ that form an orthonormal basis for the reachability (Krylov) subspace $\mathcal{R}_k(\mathbf{A}, \mathbf{v}_1)$.

We now need to orthogonalize $\mathbf{A}\mathbf{v}_k$ with respect to the $\mathbf{v}_i$, $i = 1, \cdots, k$. This can be done by applying the Gram–Schmidt procedure to the new direction, namely $\mathbf{A}\mathbf{v}_k$. The component $\mathbf{r}_k$ of $\mathbf{A}\mathbf{v}_k$ orthogonal to the span of $\mathcal{R}_k(\mathbf{A}, \mathbf{v}_1)$, is given by

$$\mathbf{r}_k = \mathbf{A}\mathbf{v}_k - \sum_{i=1}^{k} (\mathbf{v}_i^* \mathbf{A}\mathbf{v}_k)\, \mathbf{v}_i = \mathbf{A}\mathbf{v}_k - \mathbf{V}_k \left[\mathbf{V}_k^* \mathbf{A}\mathbf{v}_k\right] = \left[\mathbf{I} - \mathbf{V}_k \mathbf{V}_k^*\right] \mathbf{A}\mathbf{v}_k$$

Thus the new vector in the sequence is $\mathbf{v}_{k+1} = \frac{\mathbf{r}_k}{\|\mathbf{r}_k\|}$. Therefore

$$\mathbf{A}\mathbf{v}_k = \sum_{i=1}^{k+1} \alpha_{i,k} \mathbf{v}_i \quad \text{where} \quad \alpha_{i,\ell} = \mathbf{v}_i^* \mathbf{A}\mathbf{v}_\ell \in \mathbb{R}.$$

The above equations can also be written compactly as follows:

$$\mathbf{A}\mathbf{V}_k = \mathbf{V}_k \mathbf{H}_k + \mathbf{r}_k \mathbf{e}_k^* \quad \text{where} \quad \mathbf{H}_k = \mathbf{V}_k^* \mathbf{A}\mathbf{V}_k.$$

Thus, $\mathbf{H}_k$, is **tridiagonal**; for simplicity let $\alpha_i = \alpha_{i,i}$ and $\beta_{i+1} = \alpha_{i,i+1}$; then

$$\mathbf{H}_k = \begin{bmatrix} \alpha_1 & \beta_2 & & & & \\ \beta_2 & \alpha_2 & \beta_3 & & & \\ & \beta_3 & \alpha_3 & & & \\ & & \ddots & \ddots & \ddots & \\ & & & & \alpha_{k-1} & \beta_k \\ & & & & \beta_k & \alpha_k \end{bmatrix}.$$

Thus the vectors in the Lanczos procedure satisfy a **three term recurrence** relationship:

$$\mathbf{A}\mathbf{v}_i = \beta_{i+1}\mathbf{v}_{i+1} + \alpha_i\mathbf{v}_i + \beta_i\mathbf{v}_{i-1}, \ i = 1, 2, \cdots, k-1.$$

Eigenvalue estimation.

- If $\mathbf{r}_k = 0$, Lanczos terminates, in which case if $(\lambda, \mathbf{x})$ is an eigenpair of $\mathbf{H}_k$, $(\lambda, \mathbf{V}_k\mathbf{x})$ is an eigenpair of $\mathbf{A}$ (since $\mathbf{H}_k\mathbf{x} = \lambda\mathbf{x}$ implies $\mathbf{A}\mathbf{V}_k\mathbf{x} = \mathbf{V}_k\mathbf{H}_k\mathbf{x} = \lambda\mathbf{V}_k\mathbf{x}$).

- Otherwise, $\mathbf{A}$ has an eigenvalue in the interval $[\lambda + \mu, \lambda - \mu]$, where $\mu = \|\mathbf{A}\mathbf{x} - \lambda\mathbf{x}\|$, and $\rho(\mathbf{V}_k\mathbf{x}) = \frac{\mathbf{x}^*\mathbf{V}_k^*\mathbf{A}\mathbf{V}_k\mathbf{x}}{\mathbf{x}^*\mathbf{x}} = \lambda$, is the so-called **Rayleigh quotient**. Moreover, by interlacing, we also have $\lambda_k(\mathbf{A}) \leq \lambda \leq \lambda_1(\mathbf{A})$. ■

The Arnoldi method

The procedure described above can also be applied to non-symmetric matrices $\mathbf{A}$. The resulting process is known as the *Arnoldi method*. The difference is that the projected matrix $\mathbf{H}_k$ loses its symmetry and becomes a *Hessenberg matrix*; in this case we denote the entries by $h_{i,j} = \alpha_{i,j}$:

$$\mathbf{H}_k = \begin{bmatrix} h_{1,1} & h_{1,2} & h_{1,3} & \cdots & h_{1,k-1} & h_{1,k} \\ h_{2,1} & h_{2,2} & h_{2,3} & \cdots & h_{2,k-1} & h_{2,k} \\ & h_{3,2} & h_{3,3} & & h_{3,k-1} & h_{3,k} \\ & & & \ddots & \vdots & \vdots \\ & & & & h_{k-1,k-1} & h_{k-1,k} \\ & & & & h_{k,k-1} & h_{k,k} \end{bmatrix}.$$

A key consequence of this lack of tridiagonal structure is that *long recurrences* are now needed to construct $\mathbf{v}_{k+1}$. This is in contrast to the Lanczos procedure where only three-term recurrences are necessary.

Theorem.

1. $\mathbf{H}_k$ is obtained by projecting $\mathbf{A}$ onto the span of the columns of $\mathbf{V}_k$: $\mathbf{H}_k = \mathbf{V}_k^* \mathbf{A} \mathbf{V}_k$.
2. The remainder $\mathbf{R}_k$ has rank one and can be written as $\mathbf{R}_k = \mathbf{r}_k \mathbf{e}_k^*$, where $\mathbf{e}_k$ is the k^{th} unit vector; thus $\mathbf{r}_k \perp \mathcal{R}_k$.
3. This further implies that $\mathbf{v}_{k+1} = \frac{\mathbf{r}_k}{\|\mathbf{r}_k\|}$, where $\mathbf{v}_{k+1}$ is the $(k+1)^{\text{st}}$ column of $\mathbf{V}$. Consequently, $\mathbf{H}_k$ is an *upper Hessenberg* matrix.
4. Let $\mathbf{p}_k(\lambda) = \det(\lambda \mathbf{I}_k - \mathbf{H}_k)$, be the characteristic polynomial of $\mathbf{H}_k$. This monic polynomial is the solution of the following minimization problem:

$$\mathbf{p}_k = \arg\min \|\mathbf{p}(\mathbf{A})\mathbf{b}\|_2,$$

   where the minimum is taken over all *monic* polynomials $\mathbf{p}$ of degree k. Since $\mathbf{p}_k(\mathbf{A})\mathbf{b} = \mathbf{A}^k \mathbf{b} + \mathcal{R}_k \cdot \underline{\mathbf{p}}$, where $\underline{\mathbf{p}}_{i+1}$ is the coefficient of λ^i of the polynomial $\mathbf{p}_k$, it also follows that the coefficients of $\mathbf{p}_k$ provide the least squares fit between $\mathbf{A}^k \mathbf{b}$ and the columns of $\mathcal{R}_k$.
5. There holds

$$\mathbf{r}_k = \frac{1}{\|\mathbf{p}_{k-1}(\mathbf{A})\mathbf{b}\|} \mathbf{p}_k(\mathbf{A})\mathbf{b}, \quad \mathbf{H}_{k,k-1} = \frac{\|\mathbf{p}_k(\mathbf{A})\mathbf{b}\|}{\|\mathbf{p}_{k-1}(\mathbf{A})\mathbf{b}\|}.$$

The above result is based on the following *fundamental* lemma.

Lemma.

Let $\mathbf{AV} = \mathbf{VH} + \mathbf{f}\mathbf{e}_k^*$, with $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{H} \in \mathbb{R}^{k\times k}$ upper Hessenberg, $\mathbf{V} \in \mathbb{R}^{n\times k}$, $\mathbf{V}^*\mathbf{V} = \mathbf{I}_k$, and $\mathbf{V}\mathbf{e}_1 = \mathbf{v}_1$. There holds

$$\mathbf{A}^j\mathbf{v}_1 = \mathbf{V}\mathbf{H}^j\mathbf{e}_1 \quad \text{for} \quad 0 \le j < k,$$

and in addition, for any polynomial ϕ of degree less than k

$$\phi(\mathbf{A})\mathbf{v}_1 = \mathbf{V}\phi(\mathbf{H})\mathbf{e}_1.$$

For $j = k$ we have

$$\mathbf{A}^k\mathbf{v}_1 = \mathbf{V}\mathbf{H}^k\mathbf{e}_1 + \nu\mathbf{f}, \quad \nu = \mathbf{\Pi}_{i=1}^{k-1} h_{i+1,i} \in \mathbb{R},$$

that is, $\nu = \mathbf{e}_k^*\mathbf{H}^k\mathbf{e}_1$, is the product of the entries of the subdiagonal of $\mathbf{H}$. Furthermore, for any polynomial ϕ of degree k there holds

$$\phi(\mathbf{A})\mathbf{v}_1 = \mathbf{V}\phi(\mathbf{H})\mathbf{e}_1 + \nu\alpha_k\mathbf{f},$$

where α_k is the coefficient of the highest power ξ^k, of $\phi(\xi)$.

The Arnoldi algorithm: recursive implementation

Given: the pair $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{b} \in \mathbb{R}^n$, **find**: $\mathbf{V} \in \mathbb{R}^{n\times k}$, $\mathbf{f} \in \mathbb{R}^n$, and $\mathbf{H} \in \mathbb{R}^{k\times k}$, such that

$$\mathbf{AV} = \mathbf{VH} + \mathbf{fe}_k^*, \text{ where}$$
$$\mathbf{H} = \mathbf{V}^*\mathbf{AV},\ \mathbf{V}^*\mathbf{V} = \mathbf{I}_k,\ \mathbf{V}^*\mathbf{f} = \mathbf{0}$$

where $\mathbf{H}$ is in *upper Hessenberg* form (as before $\mathbf{e}_k$ denotes the k^{th} unit vector in $\mathbb{R}^n$).

The Arnoldi algorithm

1. $\mathbf{v}_1 = \frac{\mathbf{b}}{\|\mathbf{b}\|}$, $\mathbf{w} = \mathbf{A}\mathbf{v}_1$; $\alpha_1 = \mathbf{v}_1^*\mathbf{w}$
   $\mathbf{f}_1 = \mathbf{w} - \mathbf{v}_1\alpha_1$; $\mathbf{V}_1 = (\mathbf{v}_1)$; $\mathbf{H}_1 = (\alpha_1)$
2. For $j = 1, 2, \cdots, k-1$
   $\beta_j = \| \mathbf{f}_j \|$, $\mathbf{v}_{j+1} = \frac{\mathbf{f}_j}{\beta_j}$
   $\mathbf{V}_{j+1} = (\mathbf{V}_j \ \ \mathbf{v}_{j+1})$, $\hat{\mathbf{H}}_j = \begin{pmatrix} \mathbf{H}_j \\ \beta_j \mathbf{e}_j^* \end{pmatrix}$
   $\mathbf{w} = \mathbf{A}\mathbf{v}_{j+1}$, $\mathbf{h} = \mathbf{V}_{j+1}^*\mathbf{w}$, $\mathbf{f}_{j+1} = \mathbf{w} - \mathbf{V}_{j+1}\mathbf{h}$
   $\mathbf{H}_{j+1} = \begin{pmatrix} \hat{\mathbf{H}}_j & \mathbf{h} \end{pmatrix}$

Algorithm: The k–step *Arnoldi* factorization.

Data: $\mathbf{A} \in \mathbb{R}^{n\times n}$, starting vector $\mathbf{v} \in \mathbb{R}^n$.

$\mathbf{v}_1 = \mathbf{v}/\|\mathbf{v}\|$;

$\mathbf{w} = \mathbf{A}\mathbf{v}_1$; $\alpha_1 = \mathbf{v}_1^*\mathbf{w}$;

$\mathbf{f}_1 \leftarrow \mathbf{w} - \alpha_1\mathbf{v}$;

$\mathbf{V}_1 \leftarrow (\mathbf{v}_1)$; $\mathbf{H}_1 \leftarrow (\alpha_1)$;

for $j = 1, 2, \cdots, k-1$,

$\beta_j = \|\mathbf{f}_j\|$; $\mathbf{v}_{j+1} \leftarrow \mathbf{f}_j/\beta_j$;

$\mathbf{V}_{j+1} \leftarrow (\mathbf{V}_j,\ \mathbf{v}_{j+1})$;

$\hat{\mathbf{H}}_j \leftarrow \begin{bmatrix} \mathbf{H}_j \\ \beta_j \mathbf{e}_j^* \end{bmatrix}$

$\mathbf{w} \leftarrow \mathbf{A}\mathbf{v}_{j+1}$;

$\mathbf{h} \leftarrow \mathbf{V}_{j+1}^*\mathbf{w}$; $\mathbf{f}_{j+1} \leftarrow \mathbf{w} - \mathbf{V}_{j+1}\mathbf{h}$

$\mathbf{H}_{j+1} \leftarrow (\hat{\mathbf{H}}_j,\ \mathbf{h})$;

end

Algorithm: The k–step *two-sided Lanczos* process.

Data: $\mathbf{A} \in \mathbb{R}^{n\times n}$, starting vectors $\mathbf{v}, \mathbf{w} \in \mathbb{R}^n$.

$\mathbf{v}_1 = \mathbf{v}/\|\mathbf{v}\|$; $\mathbf{w}_1 = \mathbf{w}/\|\mathbf{w}\|$;

$\mathbf{f} = \mathbf{A}\mathbf{v}_1$; $\mathbf{g} = \mathbf{A}^*\mathbf{w}_1$;

$\alpha_1 = \mathbf{w}_1^*\mathbf{f}$;

$\mathbf{f}_1 \leftarrow \mathbf{f} - \alpha_1\mathbf{v}_1$; $\mathbf{g}_1 \leftarrow \mathbf{g} - \alpha_1\mathbf{w}_1$;

for $j = 1, 2, \cdots, k-1$,

$\beta_j = \sqrt{|\mathbf{g}_j^*\mathbf{f}_j|}$; $\gamma_j = \mathrm{sign}\,(\mathbf{g}_j^*\mathbf{f}_j)\beta_j$;

$\mathbf{v}_{j+1} \leftarrow \mathbf{f}_j/\beta_j$; $\mathbf{w}_{j+1} \leftarrow \mathbf{g}_j/\gamma_j$;

$\mathbf{f} \leftarrow \mathbf{A}\mathbf{v}_{j+1} - \gamma_j\mathbf{v}_j$; $\mathbf{g} \leftarrow \mathbf{A}^*\mathbf{w}_{j+1} - \beta_j\mathbf{w}_j$;

$\alpha_{j+1} \leftarrow \mathbf{w}_{j+1}^*\mathbf{f}$;

$\mathbf{f}_{j+1} \leftarrow \mathbf{f} - \alpha_{j+1}\mathbf{v}_{j+1}$; $\mathbf{g}_{j+1} \leftarrow \mathbf{g} - \alpha_{j+1}\mathbf{w}_{j+1}$

end

The Arnoldi and the two-sided Lanczos algorithms

Rational Krylov methods

In order to accelerate convergence of the Krylov methods, one can apply a *shift-invert* strategy as for the single vector iteration. Therein $\mathbf{A}$ is replaced by $(\mathbf{A}-\lambda\mathbf{I})^{-1}$, where λ is the shift which is close to the eigenvalue of interest. This leads to the family of rational Krylov methods. A further improvement can be obtained by using *several shifts* $\lambda_1, \cdots, \lambda_k$.

The rational Arnoldi algorithm

1. Choose shift μ
   Solve $(\mathbf{A}-\mu\mathbf{I})\mathbf{v}_1 = \mathbf{b}$, normalize $\mathbf{v}_1 \leftarrow \frac{\mathbf{v}_1}{\|\mathbf{v}_1\|}$;
   Solve $(\mathbf{A}-\mu\mathbf{I})\mathbf{w} = \mathbf{v}_1$; $\alpha_1 = \mathbf{v}_1^*\mathbf{w}$
   $\mathbf{f}_1 = \mathbf{w} - \mathbf{v}_1\alpha_1$; $\mathbf{V}_1 = (\mathbf{v}_1)$; $\mathbf{H}_1 = (\alpha_1)$
2. For $j = 1, 2, \cdots, k-1$
   $\beta_j = \| \mathbf{f}_j \|$, $\mathbf{v}_{j+1} = \frac{\mathbf{f}_j}{\beta_j}$
   $\mathbf{V}_{j+1} = (\mathbf{V}_j \;\; \mathbf{v}_{j+1})$, $\hat{\mathbf{H}}_j = \begin{pmatrix} \mathbf{H}_j \\ \beta_j \mathbf{e}_j^* \end{pmatrix}$
   Solve $(\mathbf{A}-\mu\mathbf{I})\mathbf{w} = \mathbf{v}_{j+1}$, $\mathbf{h} = \mathbf{V}_{j+1}^*\mathbf{w}$, $\mathbf{f}_{j+1} = \mathbf{w} - \mathbf{V}_{j+1}\mathbf{h}$
   $\mathbf{H}_{j+1} = \begin{pmatrix} \hat{\mathbf{H}}_j & \mathbf{h} \end{pmatrix}$

Example. Consider the following symmetric matrix:

$$\mathbf{A} = \begin{bmatrix} 2 & 1 & 2 & 1 \\ 1 & 2 & 0 & 1 \\ 2 & 0 & 2 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

The symmetric Lanczos (Arnoldi) procedure for steps $k = 1, 2, 3, 4$, yields:

$$\mathbf{V}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \qquad \mathbf{H}_1 = [2], \qquad \mathbf{R}_1 = \begin{bmatrix} 0 \\ 1 \\ 2 \\ 1 \end{bmatrix},$$

$$\mathbf{V}_2 = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{\sqrt{6}} \\ 0 & \frac{2}{\sqrt{6}} \\ 0 & \frac{1}{\sqrt{6}} \end{bmatrix}, \qquad \mathbf{H}_2 = \begin{bmatrix} 2 & \sqrt{6} \\ \sqrt{6} & \frac{8}{3} \end{bmatrix}, \qquad \mathbf{R}_2 = \begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{\sqrt{54}} \\ 0 & \frac{-1}{\sqrt{54}} \\ 0 & \frac{1}{\sqrt{54}} \end{bmatrix},$$

$$\mathbf{V}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{3}} \\ 0 & \frac{2}{\sqrt{6}} & \frac{-1}{\sqrt{3}} \\ 0 & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{3}} \end{bmatrix}, \qquad \mathbf{H}_3 = \begin{bmatrix} 2 & \sqrt{6} & 0 \\ \sqrt{6} & \frac{8}{3} & \frac{1}{\sqrt{18}} \\ 0 & \frac{1}{\sqrt{18}} & \frac{4}{3} \end{bmatrix}, \qquad \mathbf{R}_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \frac{\sqrt{3}}{2} \\ 0 & 0 & 0 \\ 0 & 0 & \frac{-\sqrt{3}}{2} \end{bmatrix},$$

$$\mathbf{V}_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} \\ 0 & \frac{2}{\sqrt{6}} & \frac{-1}{\sqrt{3}} & 0 \\ 0 & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{3}} & \frac{-1}{\sqrt{2}} \end{bmatrix}, \quad \mathbf{H}_4 = \begin{bmatrix} 2 & \sqrt{6} & 0 & 0 \\ \sqrt{6} & \frac{8}{3} & \frac{1}{\sqrt{18}} & 0 \\ 0 & \frac{1}{\sqrt{18}} & \frac{4}{3} & \frac{\sqrt{3}}{\sqrt{2}} \\ 0 & 0 & \frac{\sqrt{3}}{\sqrt{2}} & 0 \end{bmatrix}, \quad \mathbf{R}_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

The eigenvalues of the successive approximants $\mathbf{H}_1$, $\mathbf{H}_2$, $\mathbf{H}_3$, $\mathbf{H}_4$ are:

$$2, \begin{pmatrix} -0.1387 \\ 4.8054 \end{pmatrix}, \begin{pmatrix} -0.1550 \\ 1.3405 \\ 4.8145 \end{pmatrix}, \begin{pmatrix} -0.7399 \\ -0.1362 \\ 2.0607 \\ 4.8153 \end{pmatrix}.$$

Example.

$$\mathbf{A} = \frac{1}{4}\begin{bmatrix} 23 & -15 & -3 & 3 \\ 3 & -11 & 1 & -1 \\ 3 & 5 & 1 & -1 \\ 7 & 1 & -19 & 3 \end{bmatrix}, \ \mathbf{b} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

The Arnoldi procedure $\mathbf{A}\mathbf{V}_k = \mathbf{V}_k\mathbf{H}_k + \mathbf{f}_k\mathbf{e}_k^*$, yields the following matrices for $k = 1, 2, 3, 4$:

$$\mathbf{V}_1 = \frac{1}{2}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \qquad \mathbf{H}_1 = [0], \qquad \mathbf{f}_1 = \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}$$

$$\mathbf{V}_2 = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \\ 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad \mathbf{H}_2 = \begin{bmatrix} 0 & 2 \\ 2 & 2 \end{bmatrix}, \quad \mathbf{f}_2 = \begin{bmatrix} 2 \\ 2 \\ -2 \\ -2 \end{bmatrix}$$

$$\mathbf{V}_3 = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & -1 \end{bmatrix}, \quad \mathbf{H}_3 = \begin{bmatrix} 0 & 2 & 2 \\ 2 & 2 & 0 \\ 0 & 4 & -2 \end{bmatrix}, \quad \mathbf{f}_3 = \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \end{bmatrix}$$

$$\mathbf{V}_4 = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}, \quad \mathbf{H}_4 = \begin{bmatrix} 0 & 2 & 2 & 5 \\ 2 & 2 & 0 & 0 \\ 0 & 4 & -2 & 2 \\ 0 & 0 & 2 & 4 \end{bmatrix}, \quad \mathbf{f}_4 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

The eigenvalues of the successive approximants $\mathbf{H}_1$, $\mathbf{H}_2$, $\mathbf{H}_3$, $\mathbf{H}_4$ are:

$$0, \quad \begin{pmatrix} 3.2361 \\ -1.2361 \end{pmatrix}, \quad \begin{pmatrix} 3.7866 \\ -1.8933 + 1.6594i \\ -1.8933 - 1.6594i \end{pmatrix}, \quad \begin{pmatrix} 5.4641 \\ 2 \\ -1.4641 \\ -2 \end{pmatrix}.$$