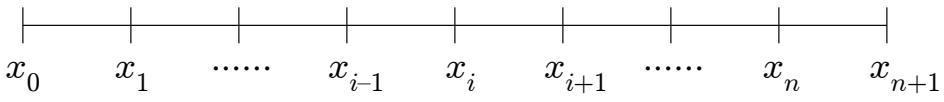
2-point BVP

$-y'' + d(x)y \;=\; f(x)\;,\quad 0 \le x \le 1\;,\quad y(0) = \alpha\;,\; y(1) = \beta$

finite-difference scheme

$h = \dfrac{1}{n+1}\;,\; x_i = ih\;,\; i = 0, 1, \ldots, n+1\;:$ mesh points



$y_i \;=\; y(x_i)\;,\; d_i \;=\; d(x_i)\;,\; f_i \;=\; f(x_i)$

$D_+ y_i \;=\; \dfrac{y_{i+1} - y_i}{h}\;,\; D_- y_i \;=\; \dfrac{y_i - y_{i-1}}{h}\;:$ forward/backward difference operators

$$D_+ D_- y_i \;=\; D_+ \left(D_- y_i\right) \;=\; D_+ \left(\frac{y_i - y_{i-1}}{h}\right) \;=\; \frac{1}{h}\left(D_+ y_i - D_+ y_{i-1}\right)$$

$$= \frac{1}{h}\left(\frac{y_{i+1} - y_i}{h} - \left(\frac{y_i - y_{i-1}}{h}\right)\right) \;=\; \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}$$

$y_{i+1} \;=\; y(x_{i+1}) \;=\; y(x_i + h)$

$$y_{i+1} \;=\; y_i \;+\; h y_i' \;+\; \tfrac{h^2}{2} y_i'' \;+\; \tfrac{h^3}{3!} y_i''' \;+\; \tfrac{h^4}{4!} y_i^{(4)} \;+\; \tfrac{h^5}{5!} y_i^{(5)} \;+\; O(h^6)$$

$$y_{i-1} \;=\; y_i \;-\; h y_i' \;+\; \tfrac{h^2}{2} y_i'' \;-\; \tfrac{h^3}{3!} y_i''' \;+\; \tfrac{h^4}{4!} y_i^{(4)} \;-\; \tfrac{h^5}{5!} y_i^{(5)} \;+\; O(h^6)$$

$$D_+ D_- y_i \;=\; \underbrace{\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}}_{\substack{\text{discrete} \\ \text{approximation}}} \;=\; \underbrace{y_i''}_{\substack{\uparrow \\ \text{exact} \\ \text{value}}} \;+\; \underbrace{\frac{h^2}{12} y_i^{(4)} \;+\; O(h^4)}_{\substack{\text{discretization} \\ \text{error}}}$$

$u_i\;:$ numerical solution $\;,\; u_i \approx y_i \;\;,\; u_0 = \alpha\;,\; u_{n+1} = \beta$

$$-\left(\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}\right) + d_i u_i \;=\; f_i\;,\; i \;=\; 1, \ldots, n\;:\text{ finite-difference scheme}$$

$$\frac{1}{h^2}\left(-u_{i+1} + \left(2 + d_i h^2\right) u_i - u_{i-1}\right) \;=\; f_i$$

$$i = 1 \;\Rightarrow\; \frac{1}{h^2}\left(-u_2 + \left(2 + d_1 h^2\right) u_1 - \alpha\right) \;=\; f_1$$

$$i = n \;\Rightarrow\; \frac{1}{h^2}\left(-\beta + \left(2 + d_n h^2\right) u_n - u_{n-1}\right) \;=\; f_n$$

$$\frac{1}{h^2}\begin{pmatrix} 2+d_1h^2 & -1 & & & & & \\ -1 & 2+d_2h^2 & -1 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & -1 & 2+d_{n-1}h^2 & -1 \\ & & & & & -1 & 2+d_nh^2 \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ \vdots \\ \vdots \\ u_{n-1} \\ u_n \end{pmatrix} = \begin{pmatrix} f_1+\alpha/h^2 \\ f_2 \\ \vdots \\ \vdots \\ \vdots \\ f_{n-1} \\ f_n+\beta/h^2 \end{pmatrix}$$

$A_h u_h = f_h$, A_h : tridiagonal , symmetric

<u>questions</u>

1. Is A_h invertible?

2. How can u_h be computed?

3. Does $u_h \to y_h$ as $h \to 0$, i.e. does the scheme converge?

<u>LU factorization for a tridiagonal system</u>

$$\begin{pmatrix} a_1 & c_1 & & & \\ b_2 & a_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & c_{n-1} \\ & & & b_n & a_n \end{pmatrix} = \begin{pmatrix} 1 & & & & \\ \beta_2 & 1 & & & \\ & \ddots & \ddots & & \\ & & \ddots & \ddots & \\ & & & \beta_n & 1 \end{pmatrix}\begin{pmatrix} \alpha_1 & c_1 & & & \\ & \alpha_2 & c_2 & & \\ & & \ddots & \ddots & \\ & & & \ddots & c_{n-1} \\ & & & & \alpha_n \end{pmatrix}$$

$A = LU$, L : unit lower triangular , U : upper triangular

consider $Ax = f$

procedure : find L, U , solve $Lz = f$, solve $Ux = z$

check : $Ax = LUx = Lz = f$ <u>ok</u>

<u>find L, U</u>

$a_1 = \alpha_1$

$b_k = \beta_k \alpha_{k-1} \;\Rightarrow\; \beta_k = \dfrac{b_k}{\alpha_{k-1}}$

$a_k = \beta_k c_{k-1} + \alpha_k \;\Rightarrow\; \alpha_k = a_k - \beta_k c_{k-1}$ for $k = 2 : n$

solve $Lz = f$: forward elimination

$z_1 = f_1$

$\beta_k z_{k-1} + z_k = f_k \;\Rightarrow\; z_k = f_k - \beta_k z_{k-1}$ for $k = 2 : n$

solve $Ux = z$: back substitution

$\alpha_n x_n = z_n \;\Rightarrow\; x_n = \dfrac{z_n}{\alpha_n}$

$\alpha_k x_k + c_k x_{k+1} = z_k \;\Rightarrow\; x_k = \dfrac{z_k - c_k x_{k+1}}{\alpha_k}$ for $k = n-1 : 1$

note

memory $= O(n)$ if vectors are used instead of full matrices

operation count $= O(n)$

---

1. matrix-vector multiplication

A : $m \times n$ matrix ($\mathbb{R}$ or $\mathbb{C}$) , x : n-vector $\Rightarrow$ $Ax = b$: m-vector

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

$$\sum_{j=1}^{n} a_{ij} x_j \;=\; b_i \;,\; i = 1 : m$$

$$\sum_{j=1}^{n} x_j a_j \;=\; b \;,\; \text{where } a_j = j\text{th column of } A$$

recall : If A is invertible, then $x \;=\; A^{-1} b$.

theorem : Let $A \in \mathbb{C}^{n \times n}$. Then the following conditions are equivalent.

1. A is invertible, i.e. there exists A^{-1} st $AA^{-1} = I$
2. $\det A \;\neq\; 0$
3. The columns of A are linearly independent.
4. $\text{range}\, A \;=\; \{b : b = Ax \text{ for some } x\} \;= \mathbb{C}^n$
5. $\text{rank}\, A \;=\; \dim \text{range}\, A \;=\; n$
6. $\text{null}\, A \;=\; \text{null space of } A \;=\; \{x : Ax = 0\} \;=\; \{0\}$
7. the e-values of A are nonzero

2. orthogonality

over $\mathbb{R}$	over $\mathbb{C}$
$x^T y = \sum_{i=1}^{n} x_i y_i$: inner product	$x^* y = \sum_{i=1}^{n} \overline{x}_i y_i$: inner product
$x^T y = 0$: orthogonal	$x^* y = 0$: orthogonal
$A^T = A$: symmetric	$A^* = A$: hermitian or self-adjoint
$A^T = A^{-1}$: orthogonal	$A^* = A^{-1}$: unitary

note

1. $x^* x \geq 0$, $x^* x = 0 \Leftrightarrow x = 0$

2. If $A \in \mathbb{C}^{n \times n}$ is hermitian, then its e-values $\lambda_1, \ldots, \lambda_n$ are real and the corresponding e-vectors $u_1, \ldots, u_n$ may be chosen to form an orthonormal basis for $\mathbb{C}^n$, i.e. $u_i^* u_j = \delta_{ij}$ and $\text{span}(u_1, \ldots, u_n) = \mathbb{C}^n$. In this case, $U = [u_1 \ldots u_n]$ is unitary.

$$A u_j = \lambda_j u_j \Rightarrow A\,U = U D \;,\; D = \text{diag}(\lambda_1, \ldots, \lambda_n)$$

$$\Rightarrow A = U D\, U^* \;:\; \text{spectral factorization}$$

If $A \in \mathbb{R}^{n \times n}$ is symmetric, then $A = Q D Q^T$, where Q is orthogonal.

spectral method for solving $Ax = b$ when A is hermitian

find D , U

expand $b = \alpha_1 u_1 + \cdots + \alpha_n u_n \;\;,\;\; \alpha_i = u_i^* b$

then $x = \dfrac{\alpha_1}{\lambda_1} u_1 + \cdots + \dfrac{\alpha_n}{\lambda_n} u_n \Rightarrow Ax = b$ ok

3. If $A \in \mathbb{C}^{m \times n}$, then $A^* A \in \mathbb{C}^{n \times n}$ is hermitian with e-values $\lambda \geq 0$.

proof

$(AB)^* = B^* A^* \Rightarrow (A^* A)^* = A^* A$

$$A^* A x = \lambda x \,,\, x \neq 0 \Rightarrow x^* A^* A x = \lambda x^* x \Rightarrow \lambda = \frac{(Ax)^*(Ax)}{x^* x} \geq 0 \quad \text{ok}$$

3. norms

A vector norm is a function $x \to ||x||$ satisfying the following properties.

1. $||x|| \geq 0$, $||x|| = 0 \Leftrightarrow x = 0$

2. $||\alpha x|| = |\alpha| \cdot ||x||$

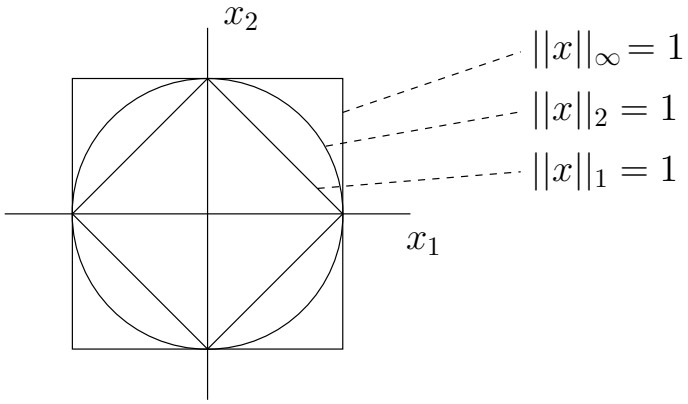
3. $||x + y|| \leq ||x|| + ||y||$

example

$$||x||_1 = \sum_{i=1}^{n} |x_i|$$

$$||x||_2 = (x^* x)^{1/2} = \left(\sum_{i=1}^{n} |x_i|^2 \right)^{1/2}$$

$$||x||_\infty = \max\{|x_i| : i = 1, \ldots, n\}$$



note : $||x||_\infty \leq ||x||_2 \leq ||x||_1$

matrix norm

1. $||A|| \geq 0$, $||A|| = 0 \Leftrightarrow A = 0$

2. $||\alpha A|| = |\alpha| \cdot ||A||$

3. $||A + B|| \leq ||A|| + ||B||$

definition : Given a vector norm $||x||$, the induced matrix norm is $||A|| = \sup_{x \neq 0} \dfrac{||Ax||}{||x||}$.

proof : hw

note

1. sup = supremum = least upper bound

   max = maximum

example : sup[0,1] = 1 , max[0,1] = 1

sup[0,1) = 1 , max[0,1) does not exist

2. $\left.\begin{array}{l} ||Ax|| \leq ||A|| \cdot ||x|| \\ ||AB|| \leq ||A|| \cdot ||B|| \end{array}\right\}$ for any induced matrix norm , proof : omit

theorem

1. $||A||_1 = \sup_{x\neq 0} \dfrac{||Ax||_1}{||x||_1} = \max_j \sum_i |a_{ij}|$: max column sum

2. $||A||_\infty = \sup_{x\neq 0} \dfrac{||Ax||_\infty}{||x||_\infty} = \max_i \sum_j |a_{ij}|$: max row sum

example

$A = \begin{pmatrix} 1 & 2 \\ 0 & 2 \end{pmatrix} \Rightarrow ||A||_1 = 4 \ , \ ||A||_\infty = 3$

proof

1. $||Ax||_1 = \sum_i |(Ax)_i| = \sum_i \Big| \sum_j a_{ij}x_j \Big| \leq \sum_i \sum_j |a_{ij}||x_j| = \sum_j \sum_i |a_{ij}||x_j|$

$$\leq \max_j \sum_i |a_{ij}| \cdot \sum_j |x_j| = \max_j \sum_i |a_{ij}| \cdot ||x||_1$$

$\Rightarrow \dfrac{||Ax||_1}{||x||_1} \leq \max_j \sum_i |a_{ij}|$ for all $x \neq 0$

note : $\max_j \sum_i |a_{ij}| = \sum_i |a_{ik}|$ for some index k

Consider $e_k = (0, \ldots, 0, \underset{\substack{\uparrow \\ k\text{th spot}}}{1}, 0, \ldots, 0)^T$.

Then $||e_k||_1 = 1 \ , \ ||Ae_k||_1 = \sum_i |a_{ik}|$.

$\Rightarrow \sup_{x\neq 0} \dfrac{||Ax||_1}{||x||_1} \leq \max_j \sum_i |a_{ij}| = \sum_i |a_{ik}| = \dfrac{||Ae_k||_1}{||e_k||_1} \leq \sup_{x\neq 0} \dfrac{||Ax||_1}{||x||_1}$ ok

2. hw

note

$||A|| = \sup_{x\neq 0} \dfrac{||Ax||}{||x||} = \sup_{||x||=1} ||Ax||$

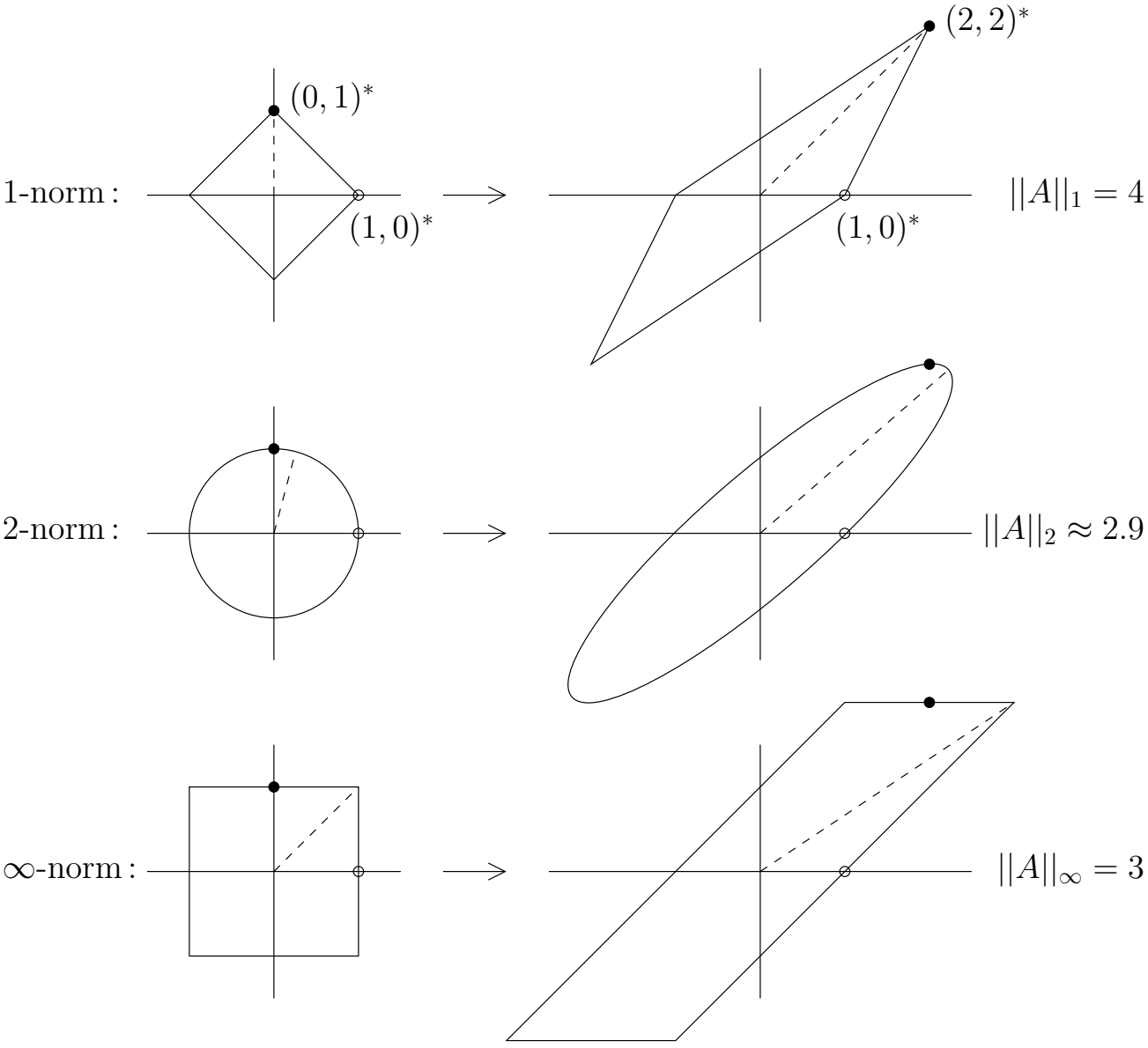
1. $||A||$ is the maximum amplification factor for vectors on the unit sphere

2. There exists a vector x such that $||x|| = 1$ and $||Ax|| = ||A||$.

Figure 3.1 from Trefethen and Bau (page 20)

$$A = \begin{pmatrix} 1 & 2 \\ 0 & 2 \end{pmatrix}$$

The left side shows the unit sphere $||x|| = 1$ in $\mathbb{R}^2$ for the 1-, 2-, and ∞-norms, and the right side shows the image of the unit sphere under the map $x \to Ax$.



note

1. Dashed lines mark unit vectors that are amplified the most in each norm.

2. We will discuss the matrix 2-norm next.

definition

spectrum : $\mathrm{sp}(A) = \{\lambda \in \mathbb{C} : \lambda \text{ is an e-value of } A\}$

spectral radius : $\rho(A) = \max\{|\lambda| : \lambda \in \mathrm{sp}(A)\}$

Rayleigh quotient : $R_A(x) = \dfrac{x^*Ax}{x^*x}$

theorem

1. If A is hermitian, then $\sup_{x\neq 0} R_A(x) =$ max e-value of A.

2. For any matrix A, $||A||_2 = \rho(A^*A)^{1/2}$.

3. If A is hermitian, then $||A||_2 = \rho(A)$.

proof

1. The e-values of A are real and hence they can be ordered, $\lambda_1 \geq \cdots \geq \lambda_n$, and the e-vectors, $u_1, \ldots, u_n$, form an orthonormal basis.

$x = \alpha_1 u_1 + \cdots + \alpha_n u_n \;\Rightarrow\; x^*x = |\alpha_1|^2 + \cdots + |\alpha_n|^2 \;,\; Ax = \alpha_1\lambda_1 u_1 + \cdots + \alpha_n\lambda_n u_n$

$x^*Ax = \lambda_1|\alpha_1|^2 + \cdots + \lambda_n|\alpha_n|^2$

$$R_A(x) = \frac{x^*A\,x}{x^*x} = \frac{\lambda_1|\alpha_1|^2 + \cdots + \lambda_n|\alpha_n|^2}{|\alpha_1|^2 + \cdots + |\alpha_n|^2} \leq \lambda_1 \;,\; R_A(u_1) = \frac{u_1^*A\,u_1}{u_1^*u_1} = \lambda_1 \quad \underline{\text{ok}}$$

$$2.\; ||A||_2 = \sup_{x\neq 0}\frac{||Ax||_2}{||x||_2} = \sup_{x\neq 0}\frac{((Ax)^*(Ax))^{1/2}}{(x^*x)^{1/2}} = \sup_{x\neq 0}\Big(\frac{x^*A^*Ax}{x^*x}\Big)^{1/2}$$

$$= \sup_{x\neq 0} R_{A^*A}(x)^{1/2} = (\text{max e-value of } A^*A)^{1/2} = \rho(A^*A)^{1/2} \quad \underline{\text{ok}}$$

$$3.\; ||A||_2 = \rho(A^*A)^{1/2} = \rho(A^2)^{1/2} = \rho(A) \quad \underline{\text{ok}}$$

example : $A = \begin{pmatrix} 1 & 2 \\ 0 & 2 \end{pmatrix}$, $||A||_2 = \rho(A^*A)^{1/2} = \sqrt{\frac{9+\sqrt{65}}{2}} = 2.9208$

$$A^*A = \begin{pmatrix} 1 & 0 \\ 2 & 2 \end{pmatrix}\begin{pmatrix} 1 & 2 \\ 0 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 2 & 8 \end{pmatrix} \Rightarrow \lambda = \frac{9 \pm \sqrt{65}}{2} \;,\; \text{check } \ldots$$

problem : find x st $\dfrac{||Ax||_2}{||x||_2} = ||A||_2$

$$Ae_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \Rightarrow \frac{||Ae_1||_2}{||e_1||_2} = 1 \;,\; Ae_2 = \begin{pmatrix} 2 \\ 2 \end{pmatrix} \Rightarrow \frac{||Ae_2||_2}{||e_2||_2} = \sqrt{8} = 2.8284 \;,\; \text{hw2}$$

note

1. $\rho(A)$ does not define a matrix norm.

2. $\rho(A) \leq ||A||$ for any induced matrix norm.

3. U : unitary $\Rightarrow$ $||Ux||_2 = ||x||_2$, $||U||_2 = 1$, $||UA||_2 = ||A||_2 = ||AU||_2$

proof

1. $A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$ $\Rightarrow$ $\rho(A) = 0$ but $A \neq 0$

2. , 3. : hw1, hw2

---

4 , 5. singular value decomposition

theorem

$A \in \mathbb{C}^{m \times n}$ $\Rightarrow$ $A = U\Sigma V^*$, where $U \in \mathbb{C}^{m \times m}$ and $V \in \mathbb{C}^{n \times n}$ are unitary, $\Sigma \in \mathbb{R}^{m \times n}$, $\Sigma \approx \text{diag}(\sigma_1, \ldots, \sigma_p), p = \min(m, n), \sigma_1 \geq \cdots \geq \sigma_r > 0 \, , r \leq p \, , \sigma_{r+1} = 0 \, , \ldots$
$\ldots, \sigma_p = 0$

$m < n$

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix} = \begin{pmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{pmatrix} \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \end{pmatrix} \begin{pmatrix} \overline{v_{11}} & \overline{v_{21}} & \overline{v_{31}} \\ \overline{v_{12}} & \overline{v_{22}} & \overline{v_{32}} \\ \overline{v_{13}} & \overline{v_{23}} & \overline{v_{33}} \end{pmatrix}$$

$m > n$

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{pmatrix} = \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ u_{21} & u_{22} & u_{23} \\ u_{31} & u_{32} & u_{33} \end{pmatrix} \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \overline{v_{11}} & \overline{v_{21}} \\ \overline{v_{12}} & \overline{v_{22}} \end{pmatrix}$$

σ_i : s-values

$U = [u_1 \ldots u_m]$, $u_i \in \mathbb{C}^m$: left s-vectors

$V = [v_1 \ldots v_n]$, $v_j \in \mathbb{C}^n$: right s-vectors

example

$$\begin{pmatrix} 0 & 0 \\ 1 & -1 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \sqrt{2} & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \quad , \quad \text{check} \ldots$$

$m = 3$, $n = 2$, $p = 2$, $r = 1$

derivation

$A = U\Sigma V^* \Rightarrow A^*A = V\Sigma^* U^* U\Sigma V^* = V D V^*$, $D = \Sigma^*\Sigma$, $AV = U\Sigma$

$$A^*A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 1 & -1 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}$$

$$\Rightarrow \Sigma = \begin{pmatrix} \sqrt{2} & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} , \ V = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}$$

$$AV = \begin{pmatrix} 0 & 0 \\ 1 & -1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ \sqrt{2} & 0 \\ 0 & 0 \end{pmatrix} = U\Sigma$$

$$\Rightarrow u_1 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \text{, choose } u_2 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} , \ u_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \Rightarrow U = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{ok}$$

ten notes about svd

1. $\sigma_i(A) = \lambda_i(A^*A)^{1/2} \Rightarrow$ the s-values are unique, but there is some freedom in choosing the s-vectors

- -

2. $||A||_2 = \sigma_1$

proof : $||A||_2 = \rho(A^*A)^{1/2} = \sigma_1$ ok

- -

3. A : hermitian $\Rightarrow \sigma_i(A) = |\lambda_i(A)|$

proof : $A = UDU^* = U|D|\text{sign}(D)U^* = U\Sigma V^* \Rightarrow \Sigma = |D|$ ok

- -

4. $A \in \mathbb{C}^{n\times n} \Rightarrow |\det A| = \sigma_1 \cdots \sigma_n$

proof

$\det A = \det(U\Sigma V^*) = \det U \cdot \det \Sigma \cdot \det V^*$, $|\det U| = |\det V^*| = 1$ ok

- -

5. $A \in \mathbb{C}^{m\times n}$, $AV = U\Sigma \Rightarrow Av_j = \begin{cases} \sigma_j u_j \ , \ j = 1:r \\ 0 \ , \ j = r+1:n \end{cases}$

$\Rightarrow$ range $A = \text{span}(u_1, \ldots, u_r)$, null $A = \text{span}(v_{r+1}, \ldots, v_n)$

$\Rightarrow$ rank $A = r$, dim null $A = n - r \Rightarrow$ rank A + dim null $A = n$

digression : forward difference approximation of a derivative

$$D_+f(x) = \frac{f(x+h)-f(x)}{h} = f'(x) + \tfrac{1}{2}f''(x)h + \cdots = f'(x) + O(h)$$

For example, if $f(x) = e^x, x = 1$, then $f'(1) = e = 2.71828\ldots$ is the exact value.

h	$D_+f(1)$	$\lvert D_+f(1) - f'(1)\rvert$	$\lvert D_+f(1) - f'(1)\rvert / h$
0.1	2.8588	0.1406	1.4056
0.05	2.7874	0.0691	1.3821
0.025	2.7525	0.0343	1.3705
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
0	e	0	$\frac{0}{0} = \frac{1}{2}f''(1) = \frac{e}{2} = 1.3591$

note

If error $\approx ch^p$, then p is the order of accuracy of the approximation.

$\Rightarrow$ $\log(\text{error}) \approx \log(ch^p) = \log c + p \log h$

$\Rightarrow$ $p =$ slope of the data on a log-log plot

We see that $p = 1$ for large h (expected) and $p = -1$ for small h (unexpected).

question : why does the error increase for small h?

1. The computed value has two sources of error; truncation error is due to replacing the exact derivative $f'(x)$ by the finite-difference approximation $D_+f(x)$, and roundoff error is due to using finite precision arithmetic.

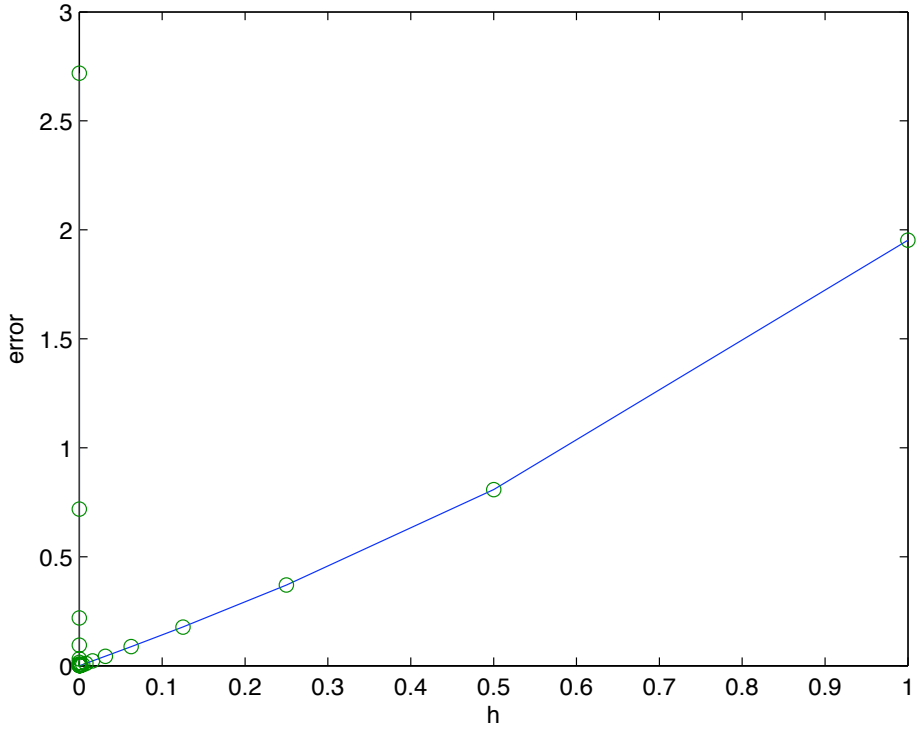
2. The truncation error is $O(h)$ and the roundoff error is $O(\epsilon/h)$, where $\epsilon \approx 10^{-15}$ in Matlab.

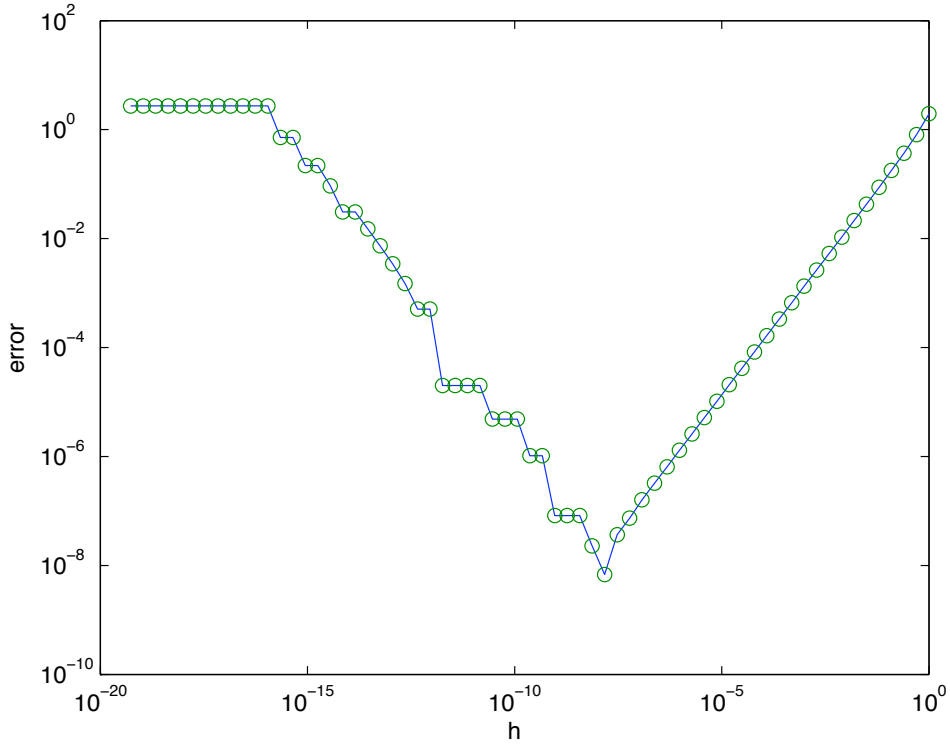
3. The total error is $O(h) + O(\epsilon/h)$, hence for large h the truncation error dominates the roundoff error, but for small h the roundoff error dominates the truncation error.

note

$D_0f(x) = \dfrac{f(x+h)-f(x-h)}{2h}$: centered difference approximation , hw2

```
% Matlab
   exact_value = exp(1);
   for j=1:65
     h(j) = 1/2^(j-1);
     computed_value = (exp(1+h(j)) - exp(1))/h(j);
     error(j) = abs(computed_value - exact_value);
   end
   plot(h,error,h,error,'o'); xlabel('h'); ylabel('error')
   loglog(h,error,h,error,'o'); ...
```





6. change of basis

$A \in \mathbb{C}^{m\times n}$, $A = U\Sigma V^*$, $x \in \mathbb{C}^n \Rightarrow y = Ax \in \mathbb{C}^m$

$U = [u_1 \dots u_m]$, $u_i \in \mathbb{C}^m$, $V = [v_1 \dots v_n]$, $v_j \in \mathbb{C}^n$

$\Rightarrow \quad x = \alpha_1 v_1 + \cdots + \alpha_n v_n \quad , \quad y = \beta_1 u_1 + \cdots + \beta_m u_m$

define $x' = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix} = \begin{pmatrix} v_1^* x \\ \vdots \\ v_n^* x \end{pmatrix} = V^* x \quad , \quad y' = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_m \end{pmatrix} = \begin{pmatrix} u_1^* y \\ \vdots \\ u_m^* y \end{pmatrix} = U^* y$

$y = Ax \Rightarrow Uy' = AVx' \Rightarrow y' = U^*AVx' \Rightarrow y' = \Sigma x'$

Hence A reduces to Σ when the domain is expressed in the basis $\{v_1, \dots, v_n\}$ and the range is expressed in the basis $\{u_1, \dots, u_m\}$.

- -

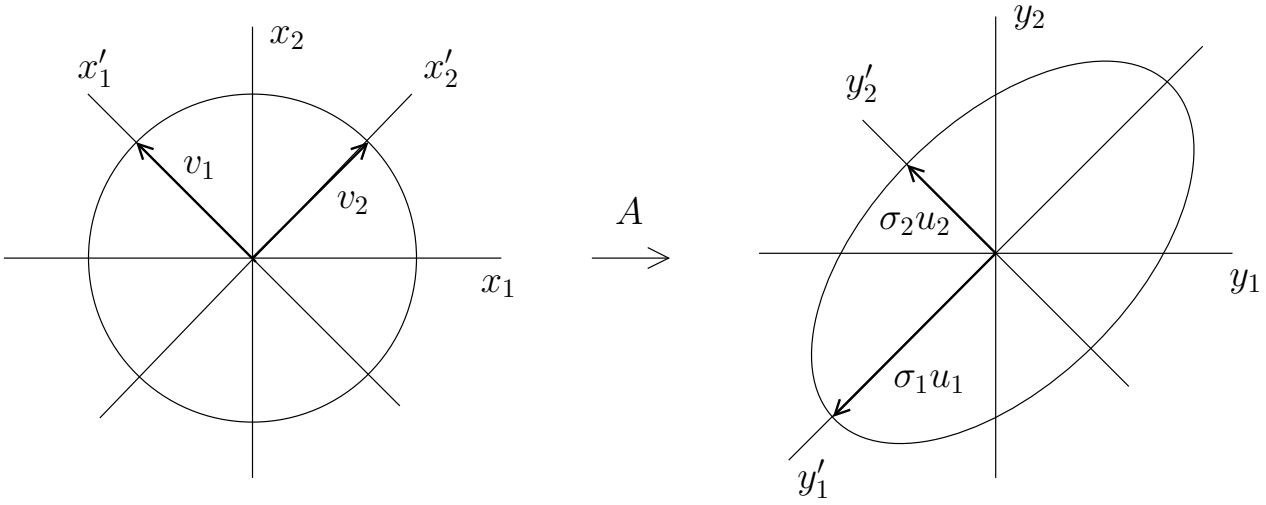
7. geometric interpretation of svd

$A \in \mathbb{R}^{m\times n}$ maps the unit sphere in $\mathbb{R}^n$ into an ellipsoid in $\mathbb{R}^m$ with principal axes $\sigma_1 u_1, \dots, \sigma_r u_r$.

$A = U\Sigma V^T$

$x \to Ux$ preserves lengths and angles, i.e. $||Ux||_2 = ||x||_2$, $(Ux)^*(Uy) = x^*y$

$x \to \Sigma x$ stretches a sphere into an ellipsoid (of possibly lower dimension)



$Av_1 = \sigma_1 u_1$, $Av_2 = \sigma_2 u_2$

$x' = V^*x$, $y' = U^*y$, $y' = \Sigma x' \Rightarrow y_1' = \sigma_1 x_1'$, $y_2' = \sigma_2 x_2'$

$(x_1')^2 + (x_2')^2 = 1 \Rightarrow \left(\frac{y_1'}{\sigma_1}\right)^2 + \left(\frac{y_2'}{\sigma_2}\right)^2 = 1$ ok

8. comparison of svd and Jordan form

svd : $A = U\Sigma V^*$, Jordan form : $A = BJB^{-1}$

a) svd applies to square and non-square matrices; Jordan applies only to square

b) svd uses 2 orthonormal bases to achieve diagonal form;

Jordan uses 1 general basis to achieve bidiagonal form

c) Jordan is used to compute A^k , e^A; svd is used as we will see soon

- -

9. reduced/compressed svd

$$\begin{pmatrix} 0 & 0 \\ 1 & -1 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \sqrt{2} & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \quad : \ (m \times m)(m \times n)(n \times n)$$

$$= \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \sqrt{2} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \quad : \ (m \times p)(p \times p)(p \times n)$$

$$= \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} (\sqrt{2}) \left(\tfrac{1}{\sqrt{2}} \quad \tfrac{-1}{\sqrt{2}} \right) \quad : \ (m \times r)(r \times r)(r \times n)$$

- -

10. best approximation

<u>theorem</u>

Consider $A \in \mathbb{C}^{m \times n}$, $A = U\Sigma V^*$, rank $A = r$.

a) $A = \sigma_1 u_1 v_1^* + \cdots + \sigma_r u_r v_r^*$: alternative form of svd

b) Let $A_k = \sigma_1 u_1 v_1^* + \cdots + \sigma_k u_k v_k^*$, $k = 1 : r-1$.

Then $||A - A_k||_2 = \sigma_{k+1} = \inf\{||A - B||_2 : B \in \mathbb{C}^{m \times n}, \text{rank}\, B = k\}$,

i.e. A_k is the best rank-k approximation of A in the 2-norm.

<u>note</u>

1. $u \in \mathbb{C}^m$, $v \in \mathbb{C}^n \Rightarrow uv^* \in \mathbb{C}^{m \times n}$, range(uv^*) = span(u) , rank$(uv^*) = 1$

2. inf = infimum = greatest lower bound $\approx$ minimum

proof

a) $\Sigma = \sigma_1 e_1 e_1^* + \cdots + \sigma_r e_r e_r^*$, where $e_i \in \mathbb{C}^m$, $e_i^* \in \mathbb{C}^n$

$A = U(\sigma_1 e_1 e_1^* + \cdots + \sigma_r e_r e_r^*)V^* = \sigma_1 (Ue_1)(Ve_1)^* + \cdots + \sigma_r (Ue_r)(Ve_r)^*$

$= \sigma_1 u_1 v_1^* + \cdots + \sigma_r u_r v_r^*$ ok

b) $A_k = \sigma_1 u_1 v_1^* + \cdots + \sigma_k u_k v_k^*$, $B \in \mathbb{C}^{m \times n}$ st $\operatorname{rank} B = k$

$\operatorname{rank} A_k = k \Rightarrow \inf_B ||A - B||_2 \leq ||A - A_k||_2$

$A - A_k = \sigma_{k+1} u_{k+1} v_{k+1}^* + \cdots + \sigma_r u_r v_r^* \Rightarrow ||A - A_k||_2 = \sigma_{k+1}$

$\Rightarrow \inf_B ||A - B||_2 \leq \sigma_{k+1}$, need to show equality

$\operatorname{null} B \subset \mathbb{C}^n$, $\dim \operatorname{null} B = n - \operatorname{rank} B = n - k$

$\operatorname{span}(v_1, \ldots, v_{k+1}) \subset \mathbb{C}^n$, $\dim \operatorname{span}(v_1, \ldots, v_{k+1}) = k + 1$

$\Rightarrow$ there exists $x \neq 0$ st $x \in \operatorname{null} B \cap \operatorname{span}(v_1, \ldots, v_{k+1})$

$\Rightarrow Bx = 0$, $x = \alpha_1 v_1 + \cdots + \alpha_{k+1} v_{k+1}$

$\Rightarrow Ax = \alpha_1 \sigma_1 u_1 + \cdots + \alpha_{k+1} \sigma_{k+1} u_{k+1}$

$$||A - B||_2 \geq \frac{||(A-B)x||_2}{||x||_2} = \frac{||Ax||_2}{||x||_2}$$

$$= \frac{(|\alpha_1|^2 \sigma_1^2 + \cdots + |\alpha_{k+1}|^2 \sigma_{k+1}^2)^{1/2}}{(|\alpha_1|^2 + \cdots + |\alpha_{k+1}|^2)^{1/2}} \geq \sigma_{k+1}$$

$\Rightarrow \inf_B ||A - B||_2 \geq \sigma_{k+1} \geq \inf_B ||A - B||_2$ ok

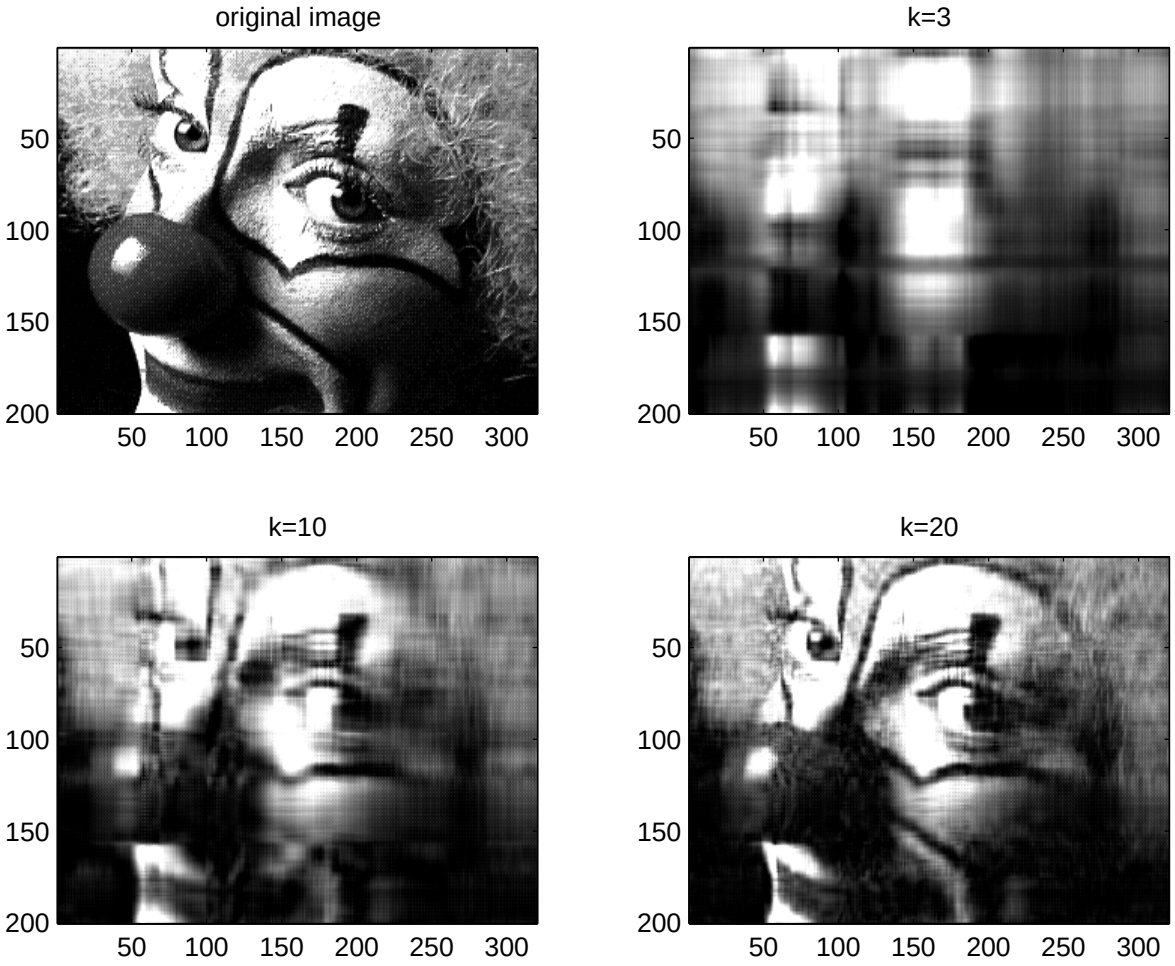
application : image compression

different names for SVD

PCA : principal component analysis

POD : proper orthogonal decomposition

```
% application of the SVD to image compression
% p. 114, "Applied Numerical Linear Algebra", J. Demmel (SIAM)
load clown.mat;
[U,S,V]=svd(X);
% X is a matrix of pixels of dimension 200 by 320 = 64000
colormap('gray');
subplot(2,2,1);
image(X); title('original image');
k=3; subplot(2,2,2);
image(U(:,1:k)*S(1:k,1:k)*V(:,1:k)'); title('k=3');
```



	relative error	compression ratio
k	σ_{k+1}/σ_1	$520k/64000$
3	0.155	0.024
10	0.077	0.081
20	0.040	0.163

6. projectors

definition : A projector is a matrix $P \in \mathbb{C}^{m\times m}$ st $P^2 = P$.

example

$$P = \begin{pmatrix} 0 & -1 \\ 0 & 1 \end{pmatrix}$$

$$P^2 = \begin{pmatrix} 0 & -1 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 0 & -1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & -1 \\ 0 & 1 \end{pmatrix} = P$$

note

1. $\text{range}\, P = \{v : Pv = v\}$
2. $I-P$ is also a projector
3. $\text{range}\, P = \text{null}(I-P)$
4. $\text{range}(I-P) = \text{null}\, P$
5. $\text{null}\, P \cap \text{null}(I-P) = \{0\}$
6. $\mathbb{C}^m = \text{range}\, P + \text{null}\, P$, $\text{range}\, P \cap \text{null}\, P = \{0\}$

proof

1. $v \in \text{range}\, P \Rightarrow v = Px \Rightarrow Pv = P^2x = Px = v$
2. $(I-P)^2 = I - 2P + P^2 = I - 2P + P = I-P$
3. a) $v \in \text{range}\, P \Rightarrow v = Px$

$$\Rightarrow (I-P)v = (I-P)Px = (P-P^2)x = 0 \Rightarrow v \in \text{null}(I-P)$$

b) $v \in \text{null}(I-P) \Rightarrow (I-P)v = 0$

$$\Rightarrow v = Pv \Rightarrow v \in \text{range}\, P$$

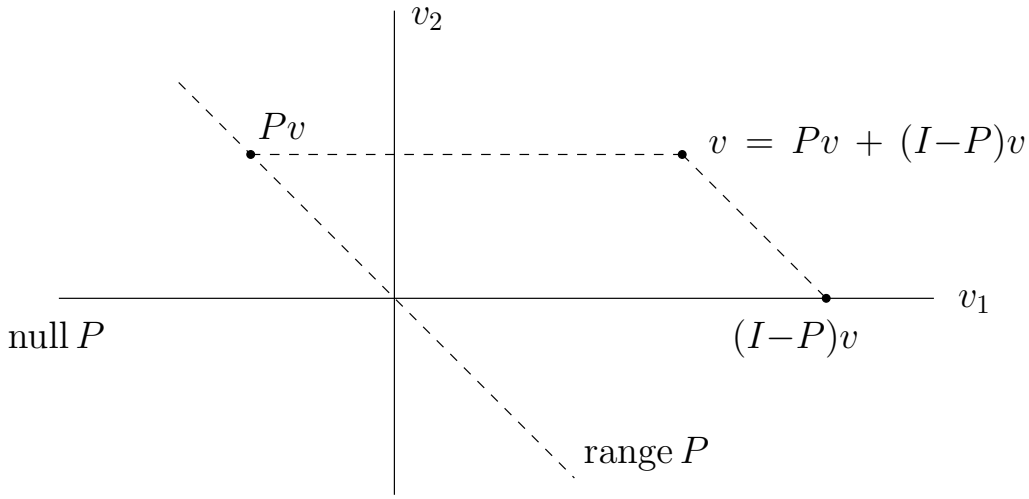
4. apply (3) with P replaced by $I-P$
5. $v \in \text{null}\, P \cap \text{null}(I-P) \Rightarrow Pv = 0$, $(I-P)v = 0 \Rightarrow v = 0$
6. a) $v = Pv + (I-P)v \in \text{range}\, P + \text{range}(I-P) \Rightarrow \mathbb{C}^m = \text{range}\, P + \text{null}\, P$

b) $\text{range}\, P \cap \text{null}\, P = \text{null}(I-P) \cap \text{null}\, P = \{0\}$

example

$P = \begin{pmatrix} 0 & -1 \\ 0 & 1 \end{pmatrix}$

$Pv = \begin{pmatrix} 0 & -1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} -v_2 \\ v_2 \end{pmatrix} = v_2 \begin{pmatrix} -1 \\ 1 \end{pmatrix} \Rightarrow \operatorname{range} P = \operatorname{span}\left\{ \begin{pmatrix} -1 \\ 1 \end{pmatrix} \right\}$



$\operatorname{null} P = \operatorname{span}\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\}$

We say that P projects $\mathbb{C}^m$ onto range P along null P.

$I-P = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & -1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$

$(I-P)^2 = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} = I-P$

$(I-P)v = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} v_1 + v_2 \\ 0 \end{pmatrix} = (v_1 + v_2) \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

$\Rightarrow \quad \operatorname{range}(I-P) = \operatorname{span}\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\} = \operatorname{null} P$

$\operatorname{null}(I-P) = \operatorname{span}\left\{ \begin{pmatrix} -1 \\ 1 \end{pmatrix} \right\} = \operatorname{range} P$

definition : An orthogonal projector is a projector P st range $P \perp$ null P.

example : $\begin{pmatrix} 0 & -1 \\ 0 & 1 \end{pmatrix}$ is not an orthogonal projector

claim: Let P be a projector. Then P is an orthogonal projector $\Leftrightarrow P^*=P$.

proof

$\Leftarrow$) Let $Px \in \text{range}\, P$, $(I-P)y \in \text{range}(I-P) = \text{null}\, P$.

Then $(Px)^*(I-P)y = x^*P^*(I-P)y = x^*P(I-P)y = 0 \Rightarrow \text{range}\, P \perp \text{null}\, P$.

$\Rightarrow$) Let $\{q_1, \dots, q_r\}$ be an orthonormal basis for range P and

$\{q_{r+1}, \dots, q_m\}$ ..... " ..... null P.

$\Rightarrow \{q_1, \dots, q_m\}$ is an orthonormal basis for $\mathbb{C}^m$ st $Pq_j = \begin{cases} q_j \ , \ j = 1:r \\ 0 \ , \ j = r+1:m \end{cases}$

Let $Q = [q_1 \cdots q_m]$: unitary. Then $PQ = QD$, where $D = \text{diag}(\underbrace{1, \dots, 1}_{r}, \underbrace{0, \dots, 0}_{m-r})$.

$\Rightarrow P = QDQ^*$: spectral factorization $\Rightarrow P^*=P$ ok

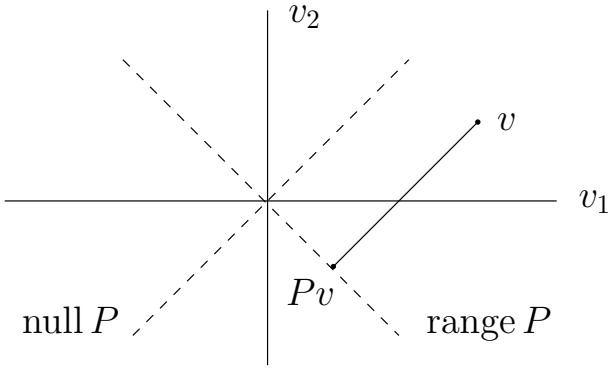
note

$P = Q(e_1e_1^* + \cdots + e_re_r^*)Q^* = q_1q_1^* + \cdots + q_rq_r^* = \hat{Q}\hat{Q}^*$, where $\hat{Q} = [q_1 \dots q_r]$

proof

$\hat{Q}^*q_j = \begin{cases} e_j \ , \ j = 1:r \\ 0 \ , \ j = r+1:m \end{cases} \Rightarrow \hat{Q}\hat{Q}^*q_j = \begin{cases} q_j \ , \ j = 1:r \\ 0 \ , \ j = r+1:m \end{cases} \Rightarrow P = \hat{Q}\hat{Q}^*$ ok

example : find the orthogonal projector onto span $\left\{\begin{pmatrix} 1 \\ -1 \end{pmatrix}\right\}$



range $P = \text{span}(q_1)$, $q_1 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix} \Rightarrow P = q_1q_1^* = \frac{1}{2}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$

check : $P^2 = q_1q_1^*q_1q_1^* = P$, $P^*=P$

$Pv = \frac{1}{2}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}\begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \frac{v_1 - v_2}{2}\begin{pmatrix} 1 \\ -1 \end{pmatrix} \Rightarrow \text{range}\, P = \text{span}\left\{\begin{pmatrix} 1 \\ -1 \end{pmatrix}\right\}$

$Pv = 0 \Rightarrow v_1 = v_2 \Rightarrow \text{null}\, P = \text{span}\left\{\begin{pmatrix} 1 \\ 1 \end{pmatrix}\right\} \perp \text{range}\, P$ ok

theorem

Let $A \in \mathbb{C}^{m\times n}$, $m \geq n$, $\operatorname{rank} A = n$.

1. A^*A is invertible

2. $P = A(A^*A)^{-1}A^*$ is the orthogonal projector onto range A

example

$A = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \Rightarrow A^*A = 2$

$P = A(A^*A)^{-1}A^* = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \cdot \frac{1}{2} \cdot \begin{pmatrix} 1 & -1 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$ ok

proof

1. hw3

2. derivation

Let P be the orthogonal projector onto range A and let $v \in \mathbb{C}^n$; we will derive an expression for Pv.

$\operatorname{range} P = \operatorname{range} A \Rightarrow Pv = Ax$

$A = [a_1 \cdots a_n]$, $\operatorname{range} A = \operatorname{span}\{a_1, \ldots, a_n\}$

$\operatorname{range} P \perp \operatorname{null} P \Rightarrow a_j \perp (I-P)v$, $j = 1:n$, because $\operatorname{null} P = \operatorname{range}(I-P)$

$\Rightarrow a_j^*(I-P)v = 0$, $j = 1:n$

$\Rightarrow A^*(I-P)v = 0$

$\Rightarrow A^*(v - Pv) = A^*(v - Ax) = 0$

$\Rightarrow A^*Ax = A^*v$

$\Rightarrow x = (A^*A)^{-1}A^*v$

$\Rightarrow Pv = Ax = A(A^*A)^{-1}A^*v$

need to check : $P^2 = P$, $P^* = P$, $\operatorname{range} P = \operatorname{range} A$: hw3

note

1. If $a_1, \ldots, a_n$ are orthonormal, then $P = A(A^*A)^{-1}A^* = AA^* = \hat{Q}\hat{Q}^*$ as before.

2. $P = A(A^*A)^{-1}A^*$ is used in solving least squares problems. (later)

7, 8, 9. QR factorization

problem : Given $A \in \mathbb{C}^{m\times n}$, $m \geq n$, $\operatorname{rank} A = n$, find an orthonormal basis for $\operatorname{range} A = \operatorname{span}\{a_1, \ldots, a_n\}$.

note : The SVD gives one solution, but here we consider an alternative.

Gram-Schmidt orthogonalization

$$q_1 = \frac{a_1}{||a_1||_2} = \frac{a_1}{r_{11}} \Rightarrow a_1 = r_{11}q_1 \ , \ ||q_1||_2 = 1$$

$$q_2 = \frac{a_2 - q_1^* a_2 q_1}{||a_2 - q_1^* a_2 q_1||_2} = \frac{a_2 - r_{12}q_1}{r_{22}} \Rightarrow a_2 = r_{12}q_1 + r_{22}q_2$$
$$q_1^* q_2 = 0 \ , \ ||q_2||_2 = 1$$

$$q_3 = \frac{a_3 - q_1^* a_3 q_1 - q_2^* a_3 q_2}{||a_3 - q_1^* a_3 q_1 - q_2^* a_3 q_2||_2} = \frac{a_3 - r_{13}q_1 - r_{23}q_2}{r_{33}}$$
$$\Rightarrow a_3 = r_{13}q_1 + r_{23}q_2 + r_{33}q_3$$
$$q_1^* q_3 = q_2^* q_3 = 0 \ , \ ||q_3||_2 = 1$$

algorithm : classical GS

for $j = 1 : n$

  $v_j = a_j$

  for $i = 1 : j-1$

    $r_{ij} = q_i^* a_j$

    $v_j = v_j - r_{ij} q_i$

  $r_{jj} = ||v_j||_2$

  $q_j = v_j / r_{jj}$

operation count

$$\sum_{j=1}^{n} \sum_{i=1}^{j-1} (m + (m-1) + m + m) \sim 4m \sum_{j=1}^{n} (j-1) \sim 4m \cdot \frac{n^2}{2} = 2mn^2 \text{ flops}$$

note

1. $r_{jj} \neq 0$ because $\{a_1, \ldots, a_n\}$ are assumed to be linearly independent

2. $$[a_1 \cdots a_n] = [q_1 \cdots q_n] \begin{pmatrix} r_{11} & r_{12} & r_{13} & \cdots & r_{1n} \\ & r_{22} & r_{23} & & \cdot \\ & & r_{33} & & \cdot \\ & & & \ddots & \cdot \\ & & & & r_{nn} \end{pmatrix}$$

$A = \hat{Q}\hat{R}$: reduced

$\hat{Q} : m \times n$, orthonormal columns

$\hat{R} : n \times n$, upper triangular , positive diagonal elements

$A = QR$: full

$Q : m \times m$, unitary - add orthonormal columns to $\hat{Q}$

$R : m \times n$, ut , pde - add zero rows to $\hat{R}$

3. $\{q_1, \ldots, q_j\}$ is an orthonormal basis for $\text{span}(a_1, \ldots, a_j)$ for $j = 1 : n$

4. $P = \hat{Q}\hat{Q}^*$ is the orthogonal projector onto range A : hw3

example

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

$$r_{11} = ||a_1||_2 = \sqrt{2} \quad , \quad q_1 = \frac{a_1}{r_{11}} = \tfrac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}$$

$$r_{12} = q_1^* a_2 = \tfrac{1}{\sqrt{2}} \quad , \quad v_2 = a_2 - r_{12} q_1 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} - \tfrac{1}{\sqrt{2}} \cdot \tfrac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1/2 \\ 1/2 \\ 1 \end{pmatrix}$$

$$r_{22} = ||v_2||_2 = \sqrt{\tfrac{3}{2}} \quad , \quad q_2 = \frac{v_2}{r_{22}} = \sqrt{\tfrac{2}{3}} \begin{pmatrix} 1/2 \\ 1/2 \\ 1 \end{pmatrix} = \begin{pmatrix} 1/\sqrt{6} \\ 1/\sqrt{6} \\ 2/\sqrt{6} \end{pmatrix}$$

$$\hat{Q}\hat{R} = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{6} \\ -1/\sqrt{2} & 1/\sqrt{6} \\ 0 & 2/\sqrt{6} \end{pmatrix} \begin{pmatrix} \sqrt{2} & 1/\sqrt{2} \\ 0 & 3/\sqrt{6} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ -1 & 0 \\ 0 & 1 \end{pmatrix} = A$$

To obtain a full QR factorization we need q_3 st $q_3 \perp q_1$, $q_3 \perp q_2$, $||q_3||_2 = 1$.

$$\text{choose } q_3 = \tfrac{1}{\sqrt{3}} \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix}$$

$$QR = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{6} & 1/\sqrt{3} \\ -1/\sqrt{2} & 1/\sqrt{6} & 1/\sqrt{3} \\ 0 & 2/\sqrt{6} & -1/\sqrt{3} \end{pmatrix} \begin{pmatrix} \sqrt{2} & 1/\sqrt{2} \\ 0 & 3/\sqrt{6} \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ -1 & 0 \\ 0 & 1 \end{pmatrix} = A$$

example

$A = \begin{pmatrix} 0.70000 & 0.70711 \\ 0.70001 & 0.70711 \end{pmatrix}$, note : $\frac{1}{\sqrt{2}} = 0.70710678$

Apply classical GS to compute QR using 5-digit arithmetic.

$r_{11} = (0.70000^2 + 0.70001^2)^{1/2}$

$= (0.49000 + 0.4900140001)^{1/2}$: exact

$= (0.49000 + 0.49001)^{1/2}$: rounded

$= 0.98995$

$q_1 = \dfrac{a_1}{r_{11}} = \begin{pmatrix} 0.70711 \\ 0.70712 \end{pmatrix}$, $r_{12} = q_1^* a_2 = 1.00000$

$v_2 = a_2 - r_{12} q_1 = \begin{pmatrix} 0.70711 \\ 0.70711 \end{pmatrix} - \begin{pmatrix} 0.70711 \\ 0.70712 \end{pmatrix} = \begin{pmatrix} 0.00000 \\ -0.00001 \end{pmatrix}$

$r_{22} = ||v_2||_2 = 0.00001$, $q_2 = \dfrac{v_2}{r_{22}} = \begin{pmatrix} 0.00000 \\ -1.00000 \end{pmatrix}$

$\tilde{Q} = \begin{pmatrix} 0.70711 & 0.00000 \\ 0.70712 & -1.00000 \end{pmatrix}$, $\tilde{R} = \begin{pmatrix} 0.98995 & 1.00000 \\ 0 & 0.00001 \end{pmatrix}$: computed factors

$\tilde{Q}\tilde{R} = \begin{pmatrix} 0.7000035445 & 0.70711 \\ 0.7000134440 & 0.70711 \end{pmatrix} = A + \delta A$, where $||\delta A||_\infty < 10^{-5}$

but $q_1^* q_2 \sim -0.7$, so $\tilde{Q}$ is far from orthogonal

$\Rightarrow$ classical GS is not backward stable (more later)

recall

$$[a_1 \cdots a_n] = [q_1 \cdots q_n] \begin{pmatrix} r_{11} & r_{12} & r_{13} & \cdots & r_{1n} \\ & r_{22} & r_{23} & & \cdot \\ & & r_{33} & & \cdot \\ & & & \ddots & \cdot \\ & & & & r_{nn} \end{pmatrix}$$

classical GS computes successive columns of R

modified GS $\cdots\cdots\cdots$ " $\cdots\cdots\cdots$ rows of R

algorithm : modified GS

for $i = 1 : n$

$v_i = a_i$

for $i = 1 : n$

$r_{ii} = ||v_i||_2$

$q_i = v_i / r_{ii}$

for $j = i + 1 : n$

$r_{ij} = q_i^* v_j$

$v_j = v_j - r_{ij} q_i$

note

1. At step j in cGS, the components of $\{q_1, \ldots, q_{j-1}\}$ are removed from a_j.
At step i in mGS, the components of q_i are removed from $\{a_{i+1}, \ldots, a_n\}$.
The algorithms are mathematically equivalent, i.e. they produce the same $\hat{Q}, \hat{R}$ in exact arithmetic, but the operations are different.

example : $a^2 - b^2 = (a + b)(a - b)$

2. The operation count for mGS is $2mn^2$. (same as cGS : hw3)

3. mGS can be expressed in matrix form. (so can cGS : hw3)

$$[a_1\ a_2\ a_3] \begin{pmatrix} \frac{1}{r_{11}} & \frac{-r_{12}}{r_{11}} & \frac{-r_{13}}{r_{11}} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{r_{22}} & \frac{-r_{23}}{r_{22}} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{r_{33}} \end{pmatrix} = [q_1\ q_2\ q_3]$$

$$\Rightarrow\ [a_1\ a_2\ a_3] = [q_1\ q_2\ q_3] \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & r_{33} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & r_{22} & r_{23} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$= [q_1\ q_2\ q_3] \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ 0 & r_{22} & r_{23} \\ 0 & 0 & r_{33} \end{pmatrix} \quad \text{ok}$$

4. R_1, R_2 : ut , pde $\Rightarrow$ $R_1 R_2, R_1^{-1}$: ut , pde

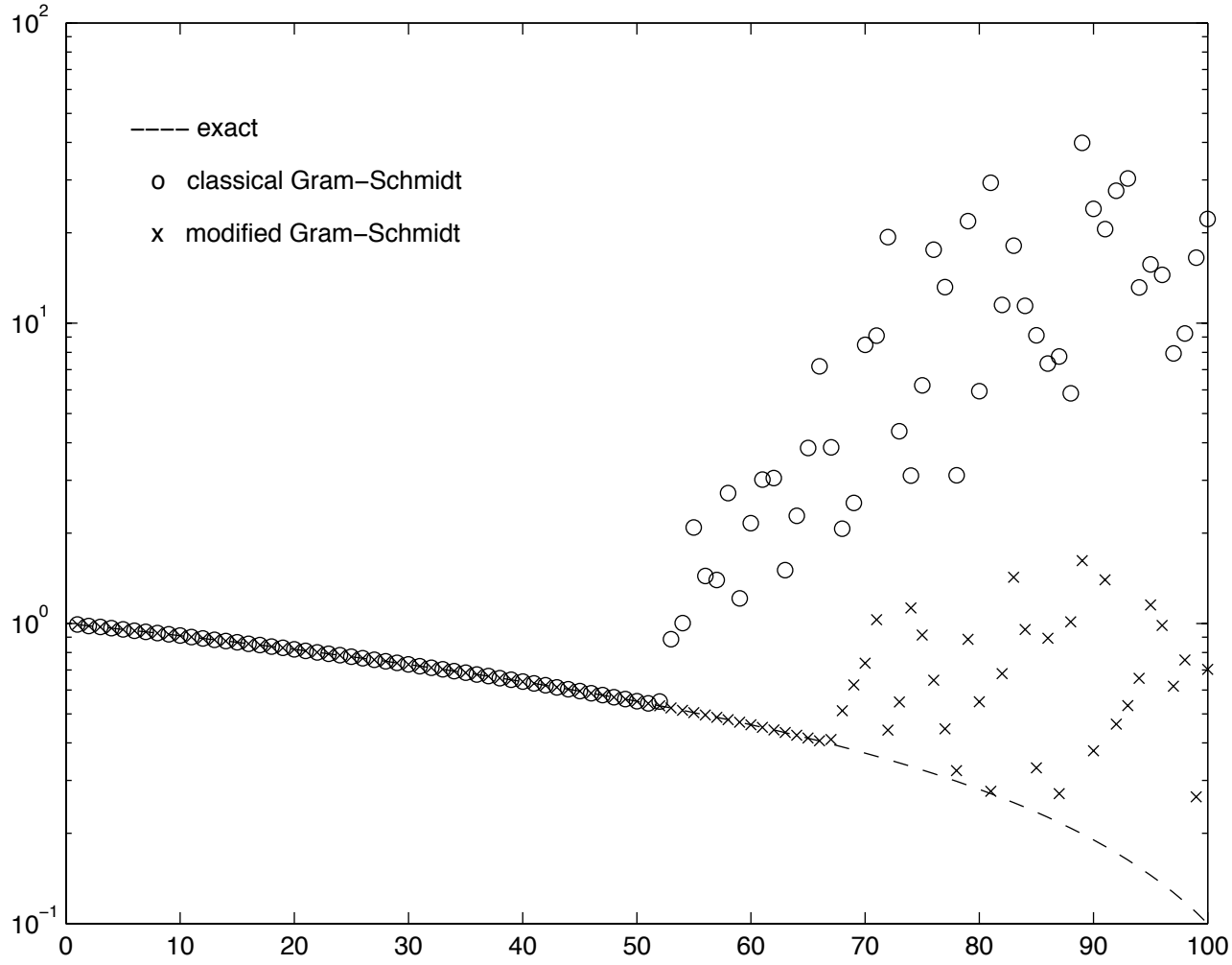
proof : omit (but more later in the context of LU factorization)

summary of GS : $A \underbrace{R_1 R_2 \cdots R_n}_{\hat{R}^{-1}} = \hat{Q} \Rightarrow A = \hat{Q}\hat{R}$

```
% comparison between classical and modified Gram-Schmidt
% A variant of Experiment 2 on page 65 in Trefethen/Bau.
clf; clear;
n = 100; [Q,R] = qr(randn(n));
x = 1:n; r = 1.0 - 0.9 * x/n; for i = 1:n; R(i,i) = r(i); end;
A = Q*R; tic; [QC,RC] = cgs(A); toc; tic; [QM,RM] = mgs(A); toc;
semilogy( x , diag(RC) , 'o' ); hold on;
semilogy( x , diag(RM) , 'x' ); hold on;
semilogy( x , r , '--' ); hold on;
```

---

typical output: `elapsed_time = 0.140803 s; elapsed time = 0.129855 s`



note

1. Modified GS is slightly more accurate than classical GS.
2. Both methods require approximately the same cpu time.
3. Matlab uses Householder's method for QR factorization. (later)

10. Householder's method for QR factorization

$A \in \mathbb{R}^{m\times n}$, $m \geq n$, $\operatorname{rank} A = n$

GS is triangular orthogonalization

Householder is orthogonal triangularization

$$\underbrace{Q_n \cdots Q_2 Q_1}_{Q^{-1}} A = R \Rightarrow A = QR : \text{ full}$$

example

$$\begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \end{pmatrix} \rightarrow \begin{pmatrix} * & * & * \\ 0 & * & * \\ 0 & * & * \\ 0 & * & * \\ 0 & * & * \end{pmatrix} \rightarrow \begin{pmatrix} * & * & * \\ 0 & * & * \\ 0 & 0 & * \\ 0 & 0 & * \\ 0 & 0 & * \end{pmatrix} \rightarrow \begin{pmatrix} * & * & * \\ 0 & * & * \\ 0 & 0 & * \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

A $\quad$ $Q_1 A$ $\quad$ $Q_2 Q_1 A$ $\quad$ $Q_3 Q_2 Q_1 A$

A_0 $\quad$ A_1 $\quad$ A_2 $\quad$ A_3

step k

$$A_k = Q_k A_{k-1} \ , \ Q_k \ : \begin{cases} \text{orthogonal} \\ \text{introduces zeros below the diagonal in column } k \\ \text{rows and columns } 1 : k-1 \text{ are unchanged} \end{cases}$$

$$Q_k A_{k-1} = \begin{pmatrix} I_{k-1} & 0 \\ 0 & H \end{pmatrix} \begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ 0 & HA_{22} \end{pmatrix} = A_k$$

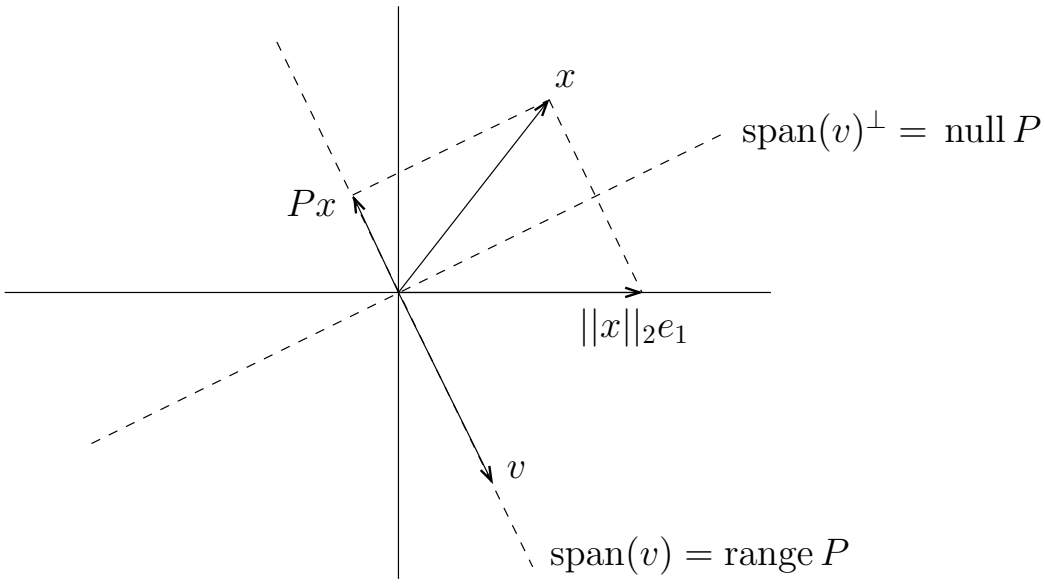
$$H : \begin{cases} \text{orthogonal} \\ Hx = \pm||x||_2 \, e_1 \text{ , } x = \text{1st column of } A_{22} \text{ , e.g. } x = \begin{pmatrix} * \\ * \\ * \\ * \end{pmatrix} \text{ , } Hx = \begin{pmatrix} \pm||x||_2 \\ 0 \\ 0 \\ 0 \end{pmatrix} \end{cases}$$

We will choose H to be a Householder reflector.

theorem : Given x, let $v = ||x||_2 e_1 - x$.

Define $H = I - 2P$, $P = \dfrac{vv^*}{||v||_2^2}$: orthogonal projector onto span(v).

1. $Px = -\dfrac{1}{2}v$
2. $Hx = ||x||_2 e_1$
3. H is hermitian , orthogonal
4. $\left.\begin{array}{l} Hv = -v \\ Hw = w \text{ for all } w \perp v \end{array}\right\}$: H reflects vectors across span$(v)^\perp$



proof

1. $Px = \dfrac{vv^*x}{||v||_2^2} = -\dfrac{1}{2}v$

$v^*x = (||x||_2 e_1 - x)^* x = ||x||_2 x_1 - ||x||_2^2 = ||x||_2(x_1 - ||x||_2)$

$||v||_2^2 = v^*v = (||x||_2 e_1 - x)^*(||x||_2 e_1 - x)$

$= ||x||_2^2 - 2||x||_2 x_1 + ||x||_2^2 = 2||x||_2(||x||_2 - x_1) = -2||x||_2(x_1 - ||x||_2) = -2v^*x$

2. $Hx = (I-2P)x = x - 2Px = x - 2 \cdot -\dfrac{1}{2}v = x + v = ||x||_2 e_1$
3. $H^* = (I-2P)^* = I - 2P = H$

$H^*H = (I-2P)^2 = I - 4P + 4P^2 = I$

4. $Hv = (I-2P)v = v - 2Pv = v - 2v = -v$

let $w \perp v$, then $Hw = (I-2P)w = w - 2Pw = w$ ok

note

$HA_{22} = (I - 2P)A_{22} = A_{22} - 2\dfrac{vv^*}{||v||_2^2}A_{22}$: we can avoid forming H explicitly

algorithm : Householder's QR factorization

for $k = 1 : n$

$x = A_{k:m,k}$

$v_k = ||x||_2 e_1 - x$

$v_k = v_k / ||v_k||_2$

for $j = k : n$

$A_{k:m,j} = A_{k:m,j} - 2v_k(v_k^* A_{k:m,j})$

note

1. input : A , output : $R, v_1, \ldots, v_n$, i.e. Q is obtained implicitly

2. For numerical stability we can replace $v = ||x||_2 e_1 - x$ by $v = -||x||_2 e_1 - x$ and the resulting R may have some negative diagonal elements.

operation count

v_k has length $\ell = m - k + 1$

inner loop : $2\ell - 1 + 1 + \ell + \ell = 4\ell$

total : $\displaystyle\sum_{k=1}^{n}\sum_{j=k}^{n} 4(m-k+1) = 4m\sum_{k=1}^{n}\sum_{j=k}^{n} 1 - 4\sum_{k=1}^{n}\sum_{j=k}^{n} k + 4\sum_{k=1}^{n}\sum_{j=k}^{n} 1$

$\sim 4m \cdot \dfrac{n^2}{2} - 4\displaystyle\sum_{k=1}^{n} k(n-k) \sim 2mn^2 - 4\left(n \cdot \frac{n^2}{2} - \frac{n^3}{3}\right) = 2mn^2 - \frac{2}{3}n^3$ flops

This is less than GS ($2mn^2$).

application

Let $m = n$ and consider $Ax = b$.

factor $A = QR$, then $Ax = b \Rightarrow QRx = b \Rightarrow Rx = Q^*b$: triangular

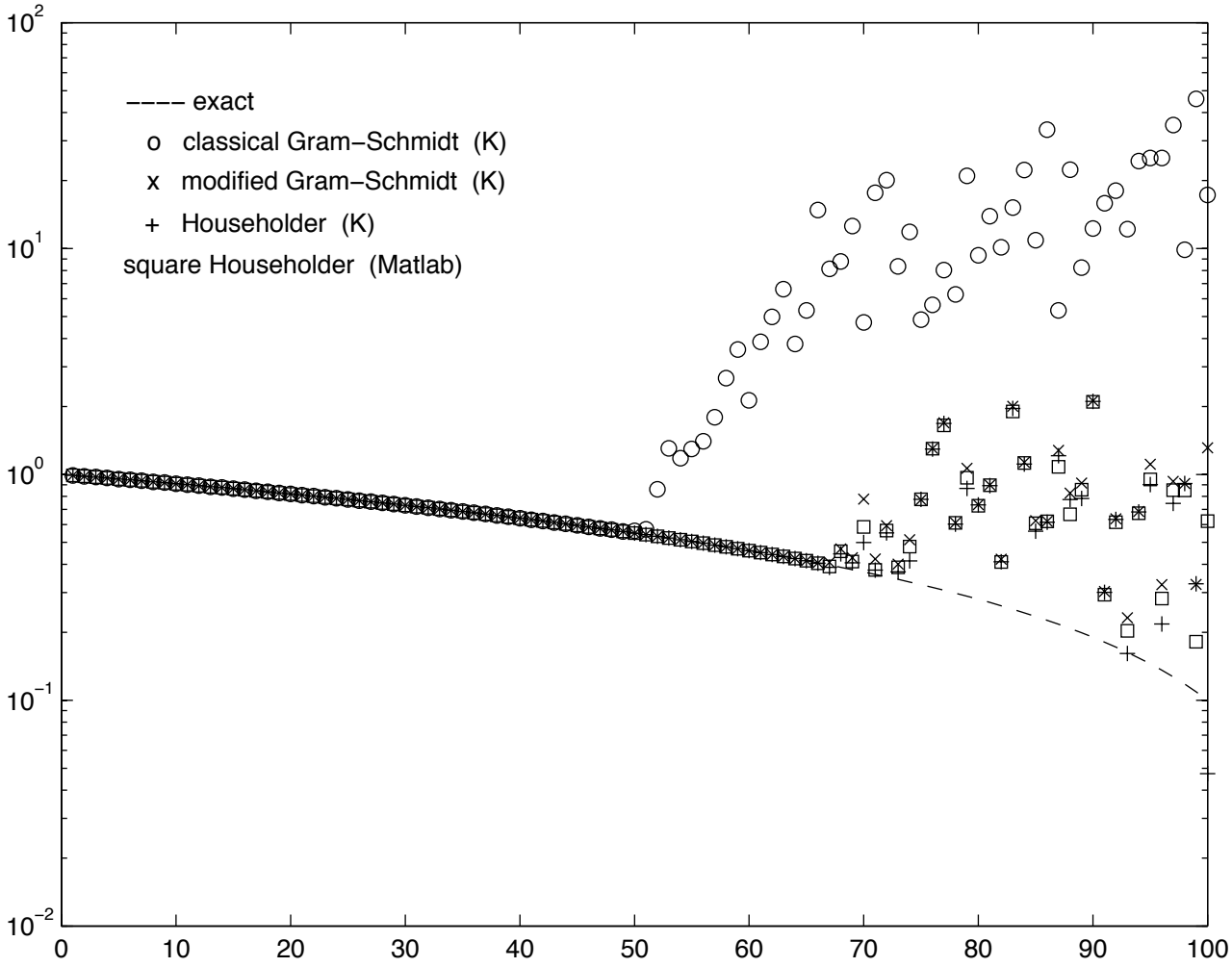
using Householder : $Q_m \cdots Q_1 A = R \Rightarrow Q = Q_1 \cdots Q_m \Rightarrow Q^* = Q_m \cdots Q_1$

Q^*b is computed implicitly : $b_{k:m} = b_{k:m} - 2v_k(v_k^* b_{k:m})$

operation count is $\dfrac{4}{3}m^3$ flops

example

Comparison between Gram-Schmidt and Householder for computing the QR factorization; a variant of Experiment 2 on page 65 in Trefethen/Bau. The computed values of R_{jj} are plotted for several methods and implementations.



	elapsed_time
classical GS (K)	0.144016
modified GS (K)	0.137808
Householder (K)	0.073608
Householder (Matlab)	0.001323

1. The modified Gram-Schmidt and Householder methods have similar accuracy.
2. My Householder code is twice as fast as my Gram-Schmidt code.
3. Matlab's Householder code is 56 times faster than my Householder code.

11. least squares problems

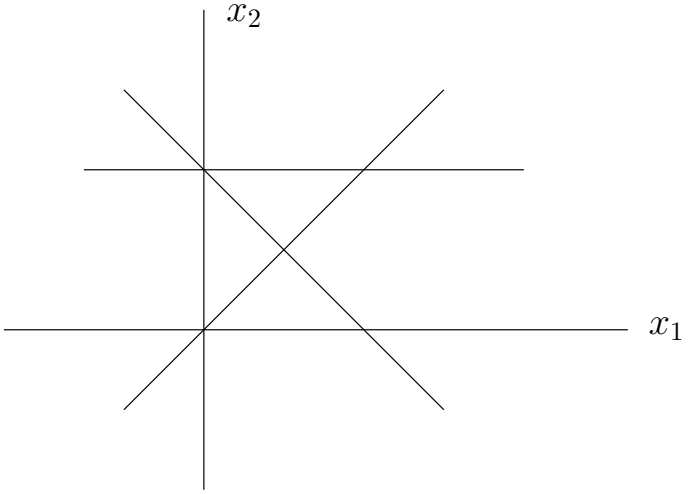
problem : Given $A \in \mathbb{C}^{m\times n},\ b \in \mathbb{C}^m$, find $x \in \mathbb{C}^n$ st $Ax = b$.

$$\left.\begin{array}{l} a_{11}x_1 + \cdots + a_{1n}x_n = b_1 \\ \cdots \\ a_{m1}x_1 + \cdots + a_{mn}x_n = b_m \end{array}\right\} : m \text{ equations , } n \text{ variables}$$

If $m > n$, the linear system $Ax = b$ is overdetermined.

example

$$A = \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ 0 & 1 \end{pmatrix},\ b = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \Rightarrow \left.\begin{array}{rcl} x_1 + x_2 &=& 1 \\ x_1 - x_2 &=& 0 \\ x_2 &=& 1 \end{array}\right\} : \text{ no solution}$$



note

$Ax = b$ has a solution $\Leftrightarrow b \in \text{range}A \Leftrightarrow b = x_1a_1 + \cdots + x_na_n$

definition : $\hat{x}$ is a least squares solution of $Ax = b$ if $||b - A\hat{x}||_2 = \min_x ||b - Ax||_2$, i.e. $\hat{x}$ minimizes the 2-norm of the residual $r = b - Ax$.

example : A , b as above

$$x = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \Rightarrow r = b - Ax = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix} \Rightarrow ||r||_2 = \sqrt{2}$$

$$x = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \Rightarrow r = b - Ax = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \Rightarrow ||r||_2 = 1$$

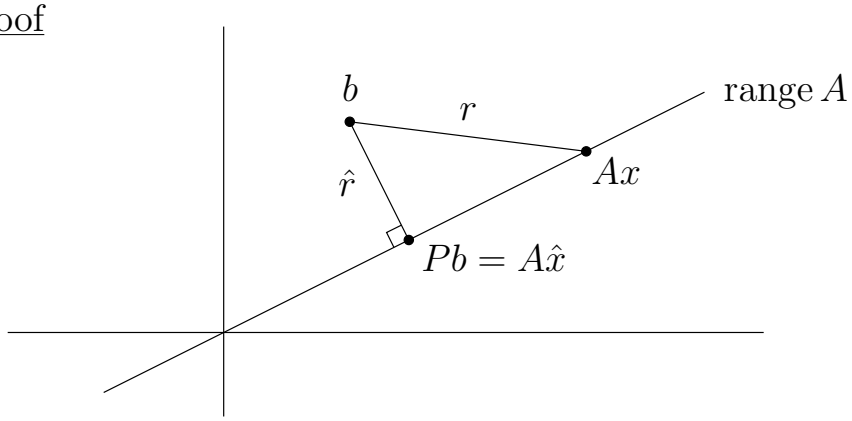
theorem

Let $A \in \mathbb{C}^{m\times n}$, $m \geq n$, $\text{rank}A = n$, $b \in \mathbb{C}^m$.
Then $\hat{x}$ is a least squares solution of $Ax = b \Leftrightarrow A^*A\hat{x} = A^*b$. : normal equations

note

1. A^*A is invertible
2. $P = A(A^*A)^{-1}A^*$ is the orthogonal projector onto range A } hw3

proof



$r = b - Ax = b - Pb + Pb - Ax \Rightarrow ||r||_2^2 = ||b - Pb||_2^2 + ||Pb - Ax||_2^2$

$\hat{x}$ is a ls solution of $Ax = b \Leftrightarrow Pb = A\hat{x} \Leftrightarrow$ normal equations (need to show)

$Pb = A\hat{x} \Rightarrow A^*Pb = A^*A(A^*A)^{-1}A^*b = A^*A\hat{x} \Rightarrow A^*b = A^*A\hat{x}$

$A^*A\hat{x} = A^*b \Rightarrow \hat{x} = (A^*A)^{-1}A^*b \Rightarrow A\hat{x} = A(A^*A)^{-1}A^*b = Pb$ ok

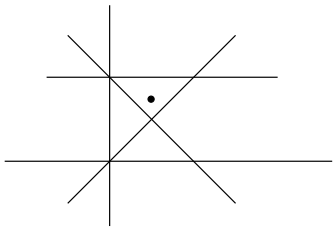
note

1. The ls solution is unique.
2. $\hat{r} = b - A\hat{x} = b - Pb = (I - P)b \perp \text{range}\, A$

example : A , b as above

$$A^*A = \begin{pmatrix} 1 & 1 & 0 \\ 1 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 3 \end{pmatrix} , \; A^*b = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

$$\hat{x} = (A^*A)^{-1}A^*b = \begin{pmatrix} 1/2 \\ 2/3 \end{pmatrix}$$



$$\hat{r} = b - A\hat{x} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} - \begin{pmatrix} 7/6 \\ -1/6 \\ 2/3 \end{pmatrix} = \begin{pmatrix} -1/6 \\ 1/6 \\ 1/3 \end{pmatrix} \Rightarrow ||\hat{r}||_2 = \frac{1}{\sqrt{6}} = 0.4082$$

definition

$(A^*A)^{-1}A^* = A^+$: pseudoinverse of A , $A^+ \in \mathbb{C}^{n\times m}$, $\hat{x} = A^+b$

example

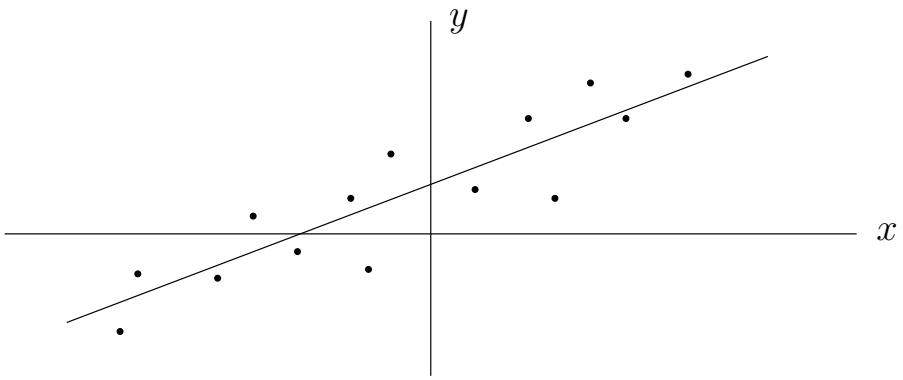
$$A = \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ 0 & 1 \end{pmatrix}$$

$$A^+ = (A^*A)^{-1}A^* = \begin{pmatrix} 1/2 & 1/2 & 0 \\ 1/3 & -1/3 & 1/3 \end{pmatrix} \ , \ \hat{x} = A^+b = \begin{pmatrix} 1/2 \\ 2/3 \end{pmatrix} \quad \text{ok}$$

note : $A^+A = I$, $AA^+ = P$

example : data fitting , regression

Suppose $(x_i, y_i), i = 1 : m$ are given data.



model : $y = \alpha x + \beta$, how to choose α, β ?

$$\left.\begin{array}{l} \alpha x_1 + \beta = y_1 \\ \vdots \\ \alpha x_m + \beta = y_m \end{array}\right\} : m \text{ equations , 2 variables}$$

$$A = \begin{pmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_m & 1 \end{pmatrix} \ , \ x = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \ , \ b = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} \ : \ Ax = b$$

$$||b - Ax||_2^2 = \sum_{i=1}^{m} (y_i - (\alpha x_i + \beta))^2$$

$$A^*Ax = A^*b \ , \ A^*A = \begin{pmatrix} \sum_{i=1}^{m} x_i^2 & \sum_{i=1}^{m} x_i \\ \sum_{i=1}^{m} x_i & m \end{pmatrix} \ , \ A^*b = \begin{pmatrix} \sum_{i=1}^{m} x_i y_i \\ \sum_{i=1}^{m} y_i \end{pmatrix}$$

note : There are 3 methods for solving ls problems.

1. normal equations

$A^*Ax = A^*b$

definition : $A \in \mathbb{C}^{m\times m}$ is positive definite if $x^*Ax > 0$ for all $x \neq 0$

note

$A \in \mathbb{C}^{m\times n}$, rank$A = n \Rightarrow A^*A \in \mathbb{C}^{n\times n}$, hermitian , pos def

proof : $x^*A^*Ax = ||Ax||_2^2 > 0$ for $x \neq 0$ ok

Cholesky factorization (more later)

$A^*A = LL^* = R^*R$, where $\begin{cases} L \in \mathbb{C}^{n\times n} \text{ , lower triangular} \\ R \in \mathbb{C}^{n\times n} \text{ , upper triangular} \end{cases}$

$A^*A\hat{x} = A^*b \Rightarrow R^*R\hat{x} = A^*b$, $\left.\begin{matrix} 1. \text{ solve } R^*y = A^*b \\ 2. \text{ solve } R\hat{x} = y \end{matrix}\right\}$: triangular

operation count

form A^*A, A^*b : mn^2 , Cholesky factorization : $\frac{1}{3}n^3$, solve : $O(n^2)$

2. QR factorization

$A = \hat{Q}\hat{R}$: reduced , $\hat{Q} \in \mathbb{C}^{m\times n}$, $\hat{R} \in \mathbb{C}^{n\times n}$

$A\hat{x} = Pb \Rightarrow \hat{Q}\hat{R}\hat{x} = \hat{Q}\hat{Q}^*b \Rightarrow \hat{R}\hat{x} = \hat{Q}^*b$: triangular

operation count : $\begin{cases} \text{GS} & : 2mn^2 \\ \text{H} & : 2mn^2 - \frac{2}{3}n^3 \end{cases}$

$m >> n \Rightarrow$ twice the work of forming the normal equations

3. SVD

$A = \hat{U}\hat{\Sigma}\hat{V}^*$: reduced , $\hat{U} \in \mathbb{C}^{m\times n}$, $\hat{\Sigma} \in \mathbb{R}^{n\times n}$, $\hat{V} \in \mathbb{C}^{n\times n}$

$A\hat{x} = Pb \Rightarrow \hat{U}\hat{\Sigma}\hat{V}^*\hat{x} = \hat{U}\hat{U}^*b \Rightarrow \hat{\Sigma}\hat{V}^*\hat{x} = \hat{U}^*b$: diagonal , unitary

operation count : later

note

1. $A^+ = (A^*A)^{-1}A^* = \hat{R}^{-1}\hat{Q}^* = \hat{V}\hat{\Sigma}^{-1}\hat{U}^*$

2. Aside from operation count, we must also consider stability.

12. condition number

example

$$\begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 32 \\ 23 \\ 33 \\ 31 \end{pmatrix} \; : \; Ax = b \;\; , \;\; A : \text{symmetric} \;\; , \;\; \det A = 1$$

$$\begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix} \begin{pmatrix} 9.2 \\ -12.6 \\ 4.5 \\ -1.1 \end{pmatrix} = \begin{pmatrix} 32.1 \\ 22.9 \\ 33.1 \\ 30.9 \end{pmatrix} \; : \; A(x + \delta x) = b + \delta b$$

$$\frac{||\delta x||_\infty}{||x||_\infty} = \frac{13.6}{1} \;\; , \;\; \frac{||\delta b||_\infty}{||b||_\infty} = \frac{0.1}{33} \;\sim\; 0.003$$

$$\frac{\text{relative change in solution}}{\text{relative change in data}} = \frac{||\delta x||_\infty / ||x||_\infty}{||\delta b||_\infty / ||b||_\infty} = \frac{13.6/1}{0.1/33} = 4488$$

note

$Ax = b \;\; , \;\; A(x + \delta x) = b + \delta b \;\; \Rightarrow \;\; A\delta x = \delta b$

$||\delta x|| \;=\; ||A^{-1}\delta b|| \;\leq\; ||A^{-1}|| \cdot ||\delta b|| \;\; , \;\; ||b|| \;=\; ||Ax|| \;\leq\; ||A|| \cdot ||x||$

$$\Rightarrow \quad \frac{||\delta x|| \,/\, ||x||}{||\delta b|| \,/\, ||b||} = \frac{||\delta x||}{||\delta b||} \cdot \frac{||b||}{||x||} \leq ||A^{-1}|| \cdot ||A|| \;:\; \text{max amplification factor}$$

equality holds when $||Ax|| = ||A|| \cdot ||x|| \;\; , \;\; ||A^{-1}\delta b|| = ||A^{-1}|| \cdot ||\delta b||$

example

$||Ax||_\infty = ||b||_\infty = 33 \;\; , \;\; ||A||_\infty = 33 \;\; , \;\; ||x||_\infty = 1$

$$A^{-1} = \begin{pmatrix} 25 & -41 & 10 & -6 \\ -41 & 68 & -17 & 10 \\ 10 & -17 & 5 & -3 \\ -6 & 10 & -3 & 2 \end{pmatrix}$$

$||A^{-1}\delta b||_\infty = ||\delta x||_\infty = 13.6 \;\; , \;\; ||A^{-1}||_\infty = 136 \;\; , \;\; ||\delta b||_\infty = 0.1$

$||A||_\infty \cdot ||A^{-1}||_\infty \;=\; 33 \cdot 136 \;=\; 4488 \quad$ ok

definition : $||A|| \cdot ||A^{-1}|| \;=\; \kappa(A) \;:\;$ condition number

$\kappa(A)$: small $\Rightarrow$ the problem of solving $Ax = b$ is well-conditioned

$\kappa(A)$: large $\Rightarrow$ $\cdots\cdots\cdots\cdots$ " $\cdots\cdots\cdots\cdots$ ill-conditioned

claim

1. $\kappa(A) \geq 1$ for any induced matrix norm

2. U : unitary $\Rightarrow$ $\kappa_2(U) = 1$, $\kappa_2(UA) = \kappa_2(AU) = \kappa_2(A)$

3. $\kappa_2(A) = \sigma_{\max}/\sigma_{\min}$

4. $A = A^* \Rightarrow \kappa_2(A) = |\lambda|_{\max}/|\lambda|_{\min}$

5. $Ax = b$, $A(x+\delta x) = b + \delta b \Rightarrow \dfrac{||\delta x|| \,/\, ||x||}{||\delta b|| \,/\, ||b||} \leq \kappa(A)$

6. $Ax = b$, $(A+\delta A)(x+\delta x) = b \Rightarrow \dfrac{||\delta x|| \,/\, ||x+\delta x||}{||\delta A|| \,/\, ||A||} \leq \kappa(A)$

proof : hw4

example

$-y'' = f$, $y(0) = y(1) = 0$

set $h = 1/N$, $x_i = ih$, $i = 0 : N$, note : previously $h = 1/(n+1)$

$-D_+D_-u_i = f_i$, $i = 1 : N-1$, $u_0 = u_N = 0$

$$\frac{1}{h^2}\begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-2} \\ u_{N-1} \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_{N-2} \\ f_{N-1} \end{pmatrix}$$

$\kappa_2(A_h) = |\lambda|_{\max}/|\lambda|_{\min} = ?$

continuous case

$-y'' = \lambda y$, $y(0) = y(1) = 0$

$\lambda_k = k^2\pi^2$, $k = 1, 2, \ldots$, $y_k(x) = \sin k\pi x$, k : wavenumber

discrete case

$A_h u_h = \lambda u_h \Leftrightarrow \dfrac{-u_{i+1} + 2u_i - u_{i-1}}{h^2} = \lambda u_i$, $u_0 = u_N = 0$

claim

$u_h = (u_{k,i})$: ith component of kth eigenvector

where $u_{k,i} = y_k(x_i) = \sin k\pi x_i = \sin ik\pi h$ and $i, k = 1 : N-1$

proof

$u_{k,0} = u_{k,N} = 0$ ok

$u_{k,i+1} = \sin(i+1)k\pi h = \sin ik\pi h \cos k\pi h + \cos ik\pi h \sin k\pi h$

$u_{k,i-1} = \sin(i-1)k\pi h = \sin ik\pi h \cos k\pi h - \cos ik\pi h \sin k\pi h$

$$\begin{aligned} -u_{k,i+1} + 2u_{k,i} - u_{k,i-1} &= -2\sin ik\pi h \cos k\pi h + 2\sin ik\pi h \\ &= 2(1-\cos k\pi h)\sin ik\pi h \\ &= 2(1-\cos k\pi h)u_{k,i} \end{aligned}$$

$$\Rightarrow \lambda_k = \frac{2}{h^2}(1-\cos k\pi h)\ ,\ k = 1 : N-1$$ ok

see picture

$$|\lambda|_{\max} = \lambda_{N-1} = \frac{2}{h^2}(1-\cos(N-1)\pi h) = \frac{2}{h^2}(1+\cos \pi h) \sim \frac{4}{h^2} \text{ as } h \to 0$$

$$|\lambda|_{\min} = \lambda_1 = \frac{2}{h^2}(1-\cos \pi h) \sim \pi^2$$

$$\kappa_2(A_h) = \frac{1+\cos \pi h}{1-\cos \pi h} \sim \frac{4}{\pi^2 h^2}$$

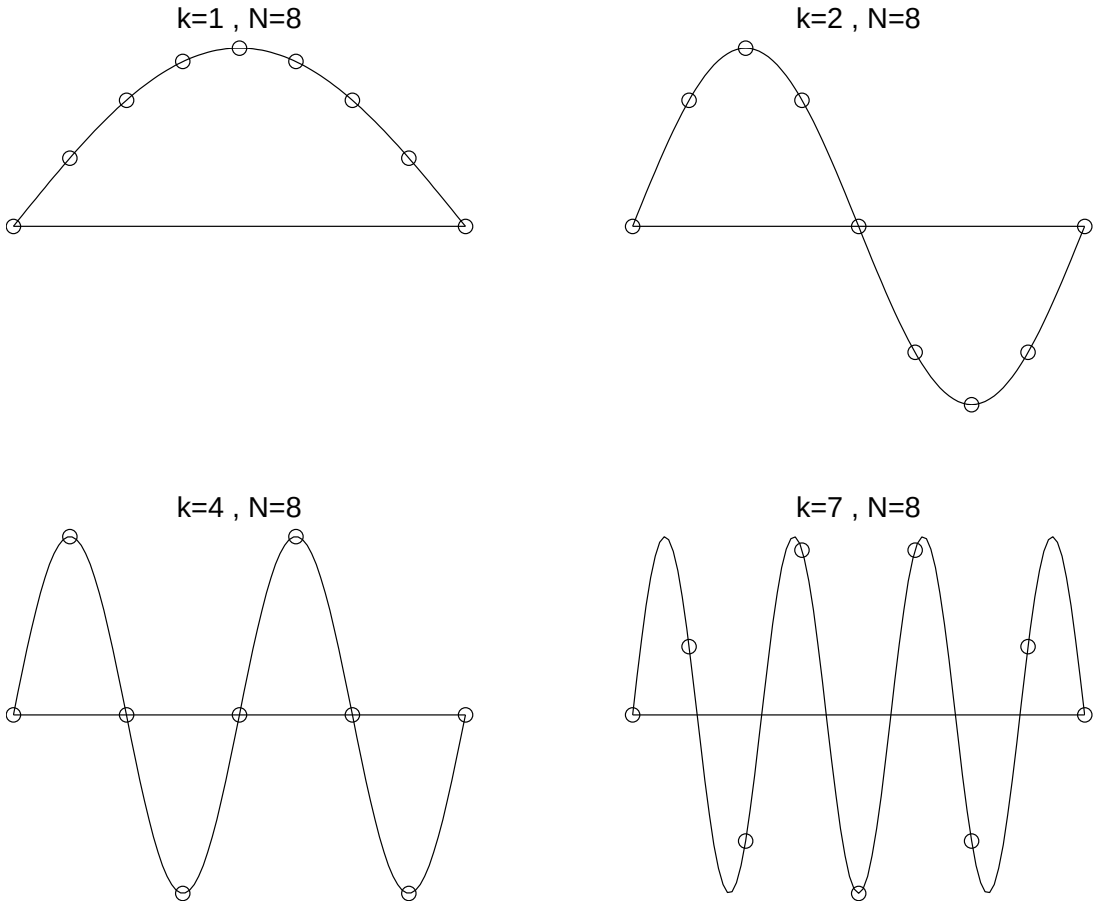
$\Rightarrow$ The problem of solving $A_h u_h = f_h$ becomes ill-conditioned as $h \to 0$.

continuous case : $-y'' = \lambda y \ , \ y(0) = y(1) = 0$

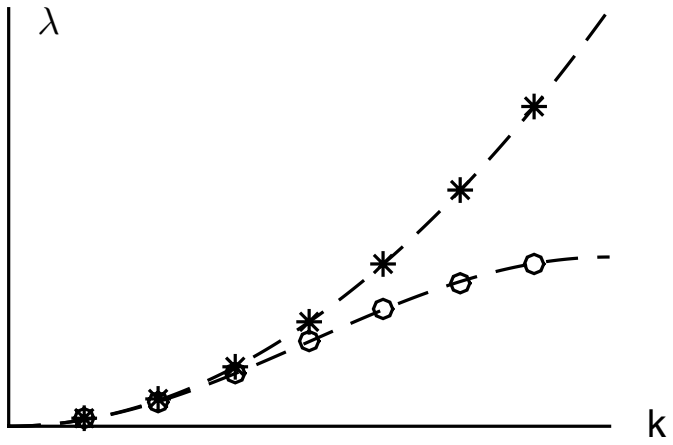
discrete case : $-D_+D_-u_i = \lambda u_i \ , \ u_0 = u_N = 0$

e-vectors : $y_k(x) = \sin k\pi x \ , \ k = 1, 2, \ldots$

$$u_{k,i} = y_k(x_i) = \sin ik\pi h \ , \ i, k = 1 : N-1$$



e-values : $\lambda_k = k^2\pi^2 \ , \ \lambda_k = \dfrac{2}{h^2}(1 - \cos k\pi h)$



13. floating point arithmetic

$x = \pm\left(\frac{m}{2^p}\right)\cdot 2^e$: base-2 floating point number

p : precision $= \begin{cases} 24 & \text{in IEEE single precision} \\ 53 & \text{" double "} \end{cases}$

m : mantissa , $1 \le m \le 2^p$

e : exponent , positive or negative integer

storage : $x =$ | $\pm$ | m | e | , 32 bits or 64 bits

$\epsilon_{\mathrm{m}} =$ machine epsilon $= \begin{cases} 2^{-24} \sim 10^{-8} \\ 2^{-53} \sim 10^{-16} \end{cases}$

$x \in \mathbb{R} \Rightarrow \mathrm{fl}(x)$: floating point representation of x

$+$, $-$, $\times$, $\div$: exact operations

$\oplus$, $\ominus$, $\otimes$, $\oslash$: floating point operations

IEEE arithmetic

1. $\mathrm{fl}(x) = x(1+\epsilon)$, where $|\epsilon| \le \epsilon_{\mathrm{m}}$

2. x, y : floating point numbers $\Rightarrow x \circledast y = (x * y)(1+\epsilon)$, where $|\epsilon| \le \epsilon_{\mathrm{m}}$

14, 15. stability

definition

A problem is defined by a mapping $x \to f(x)$, where x : data , $f(x)$: solution.

example

1. subtraction : $f(x_1, x_2) = x_1 - x_2$
2. solving $Ax = b$: $f(A, b) = A^{-1}b$

definition

$\kappa(x) = \sup_{\tilde{x} \approx x} \dfrac{||f(\tilde{x}) - f(x)||/||f(x)||}{||\tilde{x} - x||/||x||}$: condition number of f at x

note

This is consistent with our previous definition of $\kappa(A)$ for the problem of solving $Ax = b$.

definition

An algorithm for a problem f is defined by a mapping $x \to \tilde{f}(x)$, where $\tilde{f}(x)$ is an approximation to the exact solution $f(x)$ for given data x.

example

1. $\tilde{f}(x_1, x_2) = \text{fl}(x_1) \ominus \text{fl}(x_2)$
2. $\tilde{f}(A, b) =$ the result of applying Householder's method in IEEE arithmetic

definition : An algorithm $\tilde{f}$ for a problem f is called ...

1. ... accurate if $\dfrac{||\tilde{f}(x) - f(x)||}{||f(x)||} = O(\epsilon_{\text{m}})$,

   i.e. the error in the approximate solution is small.

2. ... stable if $\dfrac{||\tilde{x} - x||}{||x||} = O(\epsilon_{\text{m}}) \Rightarrow \dfrac{||\tilde{f}(\tilde{x}) - \tilde{f}(x)||}{||\tilde{f}(x)||} = O(\kappa(x) \cdot \epsilon_{\text{m}})$,

   i.e. a small change in the data yields a small but reasonable change in the approximate solution. (note : this differs from the textbook)

3. ... backward stable if for any x there exists $\tilde{x}$ st $\dfrac{||\tilde{x} - x||}{||x||} = O(\epsilon_{\text{m}})$ and $\tilde{f}(x) = f(\tilde{x})$,

   i.e. the approximate solution for the given data is the exact solution for some slightly different data.

example

1. IEEE operations are backward stable.

proof

$f(x_1, x_2) = x_1 - x_2$, $\tilde{f}(x_1, x_2) = \text{fl}(x_1) \ominus \text{fl}(x_2)$

$\text{fl}(x_1) = x_1(1+\epsilon_1)$, $\text{fl}(x_2) = x_2(1+\epsilon_2)$

$$\text{fl}(x_1) \ominus \text{fl}(x_2) = (\text{fl}(x_1) - \text{fl}(x_2))(1+\epsilon_3)$$
$$= x_1(1+\epsilon_1)(1+\epsilon_3) - x_2(1+\epsilon_2)(1+\epsilon_3) = \tilde{x}_1 - \tilde{x}_2$$

$\Rightarrow \tilde{f}(x_1, x_2) = f(\tilde{x}_1, \tilde{x}_2)$

$$\frac{|x_1 - \tilde{x}_1|}{|x_1|} = \frac{|x_1 - x_1(1+\epsilon_1)(1+\epsilon_3)|}{|x_1|} \leq |\epsilon_1| + |\epsilon_3| + |\epsilon_1\epsilon_3| \leq 3\,\epsilon_{\text{m}} \quad \text{ok}$$

2. The composition of two IEEE operations is backward stable,

i.e. $(\text{fl}(x_1) \circledast \text{fl}(x_2)) \circledast \text{fl}(x_3) = (\tilde{x}_1 * \tilde{x}_2) * \tilde{x}_3$, where $\dfrac{|\tilde{x}_i - x_i|}{|x_i|} = O(\epsilon_{\text{m}})$.

proof : hw5

3. Computing the QR factorization of a matrix A by the Gram-Schmidt method using IEEE arithmetic is not backward stable.

proof : we saw that the computed Q may not be orthogonal (page 23)

theorem

Let $x \to f(x)$ be a problem with condition number $\kappa(x)$ and let $x \to \tilde{f}(x)$ be a backward stable algorithm for f. Then $\dfrac{||\tilde{f}(x) - f(x)||}{||f(x)||} \leq \kappa(x) \cdot O(\epsilon_{\text{m}})$.

proof

$\tilde{f}$: backward stable $\Rightarrow \tilde{f}(x) = f(\tilde{x})$ for some $\tilde{x}$ st $\dfrac{||\tilde{x} - x||}{||x||} = O(\epsilon_{\text{m}})$

$$\kappa(x) = \sup_{\tilde{x} \approx x} \frac{||f(\tilde{x}) - f(x)||/||f(x)||}{||\tilde{x} - x||/||x||}$$

$$\Rightarrow \frac{||\tilde{f}(x) - f(x)||}{||f(x)||} = \frac{||f(\tilde{x}) - f(x)||}{||f(x)||} \cdot \frac{||\tilde{x} - x||/||x||}{||\tilde{x} - x||/||x||} \leq \kappa(x) \cdot O(\epsilon_{\text{m}}) \quad \text{ok}$$

note : This implies that a backward stable algorithm applied to a well-conditioned problem is accurate.

16. conditioning and stability for QR factorization

problem : given A, find Q, R st $A = QR$, where Q : orthogonal , R : ut, pde

Householder's method : $\underbrace{Q_n \cdots Q_1}_{Q^*} A = R \Rightarrow A = QR$

$Q_k = \begin{pmatrix} I_{k-1} & 0 \\ 0 & H \end{pmatrix}$, $H = I - 2P$, $P = \dfrac{v_k v_k^*}{||v_k||_2^2}$, R : explicit, Q : implicit

theorem

1. The condition number for the problem of QR factorization is the same as the condition number for the problem of solving $Ax = b$,

e.g. if $A = QR$ and $A + \delta A = (Q + \delta Q)R$, then $\dfrac{||\delta Q||_2/||Q||_2}{||\delta A||_2/||A||_2} \leq \kappa_2(A)$.

2. Householder's method with IEEE arithmetic is backward stable, i.e. if $\tilde{Q}, \tilde{R}$ are the result of applying the algorithm to a matrix A, then $\tilde{Q}$ is orthogonal, $\tilde{R}$ is ut, pde, and $\tilde{Q}\tilde{R} = A + \delta A$, where $||\delta A||/||A|| = O(\epsilon_{\mathrm{m}})$. proof : omit

example (page 114)

```
R1 = triu(randn(50)); [Q1,X] = qr(randn(50)); % Householder
norm(Q1'*Q1-eye(50))
   ans = 1.8034 e-15                          % O(ε_m)
A = Q1*R1; cond(A)
   ans = 1.3914e+16                           % κ(A) = O(ε_m^-1)
[Q2,R2] = qr(A); norm(Q2'*Q2-eye(50))
   ans = 2.0008 e-15                          % O(ε_m)
norm(Q2-Q1)/norm(Q1)
   ans = 0.0195                               % κ(A) · O(ε_m)
norm(R2-R1)/norm(R1)
   ans = 0.0032                               % κ(A) · O(ε_m)
norm(Q2*R2-A)/norm(A)
   ans = 8.5504 e-16                          % O(ε_m)
```

discussion : $A = Q_1 R_1$ and Q_2, R_2 are computed by $\mathtt{qr}(A)$; Q_1, Q_2 are orthogonal to $O(\epsilon_{\mathrm{m}})$ and R_1, R_2 are ut, pde, but Q_2, R_2 are poor approximations to Q_1, R_1 because `qr` allows roundoff errors of size $O(\epsilon_{\mathrm{m}})$ to be amplified by as much as $\kappa(A)$; nonetheless, `qr` is backward stable, i.e. Q_2, R_2 are the exact factors of a slightly different matrix.

question : what do you expect for Gram-Schmidt? hw5

17. back substitution is backward stable

This implies that Householder's method for solving $Ax = b$ is backward stable.
recall : $A = QR$, $y = Q^*b$, $Rx = y$

18. condition number of the ls problem

$A \in \mathbb{C}^{m\times n}$, $m \geq n$, $\operatorname{rank} A = n$
$Ax = b$, $||b - A\hat{x}||_2 = \min_x ||b - Ax||_2$

$f(A, b) = \hat{x}$, $\kappa(A, b)$ is a complicated function

19. stability of algorithms for ls problems

example : least squares polynomial approximation

signal : $f(t)$

measurements : $f(t_i)$, $t_i = i/(m-1)$, $i = 0 : m-1$

model : $p(t) = c_0 + c_1 t + c_2 t^2 + \cdots + c_{n-1} t^{n-1}$

$c_0 + c_1 t_0 + c_2 t_0^2 + \cdots + c_{n-1} t_0^{n-1} = f_0$

$c_0 + c_1 t_1 + c_2 t_1^2 + \cdots + c_{n-1} t_1^{n-1} = f_1$

$\vdots$

$c_0 + c_1 t_{m-1} + c_2 t_{m-1}^2 + \cdots + c_{n-1} t_{m-1}^{n-1} = f_{m-1}$

$$\begin{pmatrix} 1 & t_0 & t_0^2 & \cdots & t_0^{n-1} \\ 1 & t_1 & t_1^2 & \cdots & t_1^{n-1} \\ \vdots & & & & \\ 1 & t_{m-1} & t_{m-1}^2 & \cdots & t_{m-1}^{n-1} \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_{m-1} \end{pmatrix}$$

$Ax = b$, $A \in \mathbb{R}^{m\times n}$: Vandermonde matrix

code

```
m = 100; n = 15; t = (0:m-1)'/(m-1);
A = [ ];
for i=1:n; A = [A t.^(i-1)]; end
cond(A)
   ans = 2.2718e+10                          % κ(A) = ||A|| · ||A⁺||
b = exp(sin(4*t))/2006.787453080206;
x = A\b;                                     % x(15) = 1 + O(ε_m)
```

1. Householder

```
[Q,R] = qr(A); x = R\(Q'*b); x(15)
   ans = 1.00000031528723
```
% error $\sim 10^{-7} = 10^9 \cdot \epsilon_{\text{m}}$

Roundoff errors were amplified by a factor of 10^9; this is due to ill-conditioning; the algorithm is backward stable.

2. Gram-Schmidt

```
[Q,R] = mgs(A); x = R\(Q'*b); x(15)
   ans = 1.02926594532672
```
% error $\sim 10^{-2} = 10^{14} \cdot \epsilon_{\text{m}}$

Roundoff errors were amplified by a factor of 10^{14}; this is more than can be explained by ill-conditioning; in fact the algorithm is not backward stable.

3. normal equations

```
x = (A'*A)\(A'*b); x(15)
   ans = 0.39339069870283
```

The problem of solving $A^*Ax = A^*b$ is more ill-conditioned than the problem of solving $Ax = b$, because $\kappa_2(A^*A) = \kappa_2(A)^2$. Hence the algorithm is unstable; solving the ls problem using the normal equations increases the sensitivity of the computed solution to roundoff errors.

4. SVD

```
[U,S,V] = svd(A); x = V*(S\(U'*b)); x(15)
   ans = 0.99999998230471
```
% error $\sim 10^{-8}$

"SVD is the Cadillac of methods for solving ls problems." - W. Kahan (1982)

note

1. These results are from the textbook; yours may differ somewhat.

2. Difficulty arises here because $\kappa(A)$ is large; a better approach is to use a different polynomial basis leading to a small $\kappa(A)$, e.g. $p(t) = c_0 l_0(t) + c_1 l_1(t) + \cdots + c_{n-1} l_{n-1}(t)$, where $l_i(t)$: Legendre polynomials.

recall

$-y'' + d(x)y = f(x)\ ,\ y(0) = \alpha\ ,\ y(1) = \beta$

$D_+D_-y_i = y_i'' + \frac{h^2}{12}y_i^{(4)} + O(h^4)$

$-D_+D_-u_i + d_iu_i = f_i\ :$ 2nd order accurate $\ ,\ ||u_h - y_h||_\infty = O(h^2)$

4th order accuracy

$D_+D_-y_i = y_i'' + O(h^2)$ for any smooth function $y(x)$

$\Rightarrow\ D_+D_-y_i'' = y_i^{(4)} + O(h^2)$

$\Rightarrow\ D_+D_-\left(D_+D_-y_i + O(h^2)\right) = y_i^{(4)} + O(h^2)$

$\Rightarrow\ y_i^{(4)} = (D_+D_-)^2\, y_i + O(h^2)$

$\Rightarrow\ D_+D_-y_i = y_i'' + \frac{h^2}{12}\left((D_+D_-)^2\, y_i + O(h^2)\right) + O(h^4)$

$\Rightarrow\ y_i'' = D_+D_-y_i - \frac{h^2}{12}(D_+D_-)^2\, y_i + O(h^4)$

$= D_+D_-\left(1 - \frac{h^2}{12}D_+D_-\right)y_i + O(h^4)$

$-D_+D_-(1 - \frac{h^2}{12}D_+D_-)u_i + d_iu_i = f_i\ :$ 4th order scheme

$$(D_+D_-)^2\, u_i = D_+D_-\,(D_+D_-u_i) = D_+D_-\left(\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}\right)$$

$$= \frac{u_{i+2} - 2u_{i+1} + u_i - 2(u_{i+1} - 2u_i + u_{i-1}) + u_i - 2u_{i-1} + u_{i-2}}{h^4}$$

$$= \frac{u_{i+2} - 4u_{i+1} + 6u_i - 4u_{i-1} + u_{i-2}}{h^4}$$

$$y_i'' \approx D_+D_-u_i - \frac{h^2}{12}(D_+D_-)^2\, u_i$$

$$= \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} - \frac{h^2}{12}\left(\frac{u_{i+2} - 4u_{i+1} + 6u_i - 4u_{i-1} + u_{i-2}}{h^4}\right)$$

$$= \frac{1}{h^2}\left(-\frac{1}{12}u_{i+2} + \frac{4}{3}u_{i+1} - \frac{5}{2}u_i + \frac{4}{3}u_{i-1} - \frac{1}{12}u_{i-2}\right)\ :\ \text{5-point stencil}$$

The matrix is pentadiagonal and solving $A_hu_h = f_h$ requires more work than in the tridiagonal case.

$\underline{i = 1}$

1st equation relates u_3 , u_2 , u_1 , $u_0 = \alpha$, $u_{-1} = ?$

$\underline{\text{remedy}}$

1. extrapolate u_{-1} from u_0 , u_1 , u_2 , ...

2. use 2nd order scheme for 1st equation (u_2 , u_1 , $u_0 = \alpha$)

3. alternative

$$y_i'' = D_+D_-y_i - \frac{h^2}{12}y_i^{(4)} + O(h^4)$$

$$-y'' + dy = f \Rightarrow y'' = dy - f \Rightarrow y^{(4)} = (dy)'' - f''$$

$$y_i'' = D_+D_-y_i - \frac{h^2}{12}\left((dy)_i'' - f_i''\right) + O(h^4)$$

$$= D_+D_-y_i - \frac{h^2}{12}\left(D_+D_-(dy)_i - D_+D_-f_i + O(h^2)\right) + O(h^4)$$

$$= D_+D_-(1 - \frac{h^2}{12}d_i)y_i + \frac{h^2}{12}D_+D_-f_i + O(h^4)$$

$$-D_+D_-(1 - \frac{h^2}{12}d_i)u_i + d_iu_i = \left(1 + \frac{h^2}{12}D_+D_-\right)f_i \ : \ \text{4th order } \underline{\text{compact}} \text{ scheme}$$

$\underline{\text{note}}$

1. 3-point stencil , tridiagonal matrix

2. no problem for $i = 1$, n

3. 4th order accuracy, i.e. $||u_h - y_h||_\infty = O(h^4)$

20. Gaussian elimination

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3 \end{aligned} \quad \Rightarrow \quad \left(\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{array}\right)$$

step 1 : eliminate variable x_1 from eqs. 2 and 3 by elementary row operations

$$l_{21} = \frac{a_{21}}{a_{11}} \Rightarrow a_{22} \to a_{22} - l_{21}a_{12} \ , \ a_{23} \to a_{23} - l_{21}a_{13} \ , \ b_2 \to b_2 - l_{21}b_1$$

$$l_{31} = \frac{a_{31}}{a_{11}} \Rightarrow a_{32} \to a_{32} - l_{31}a_{12} \ , \ a_{33} \to a_{33} - l_{31}a_{13} \ , \ b_3 \to b_3 - l_{31}b_1$$

$$\left(\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ 0 & a_{22} & a_{23} & b_2 \\ 0 & a_{32} & a_{33} & b_3 \end{array}\right)$$ --- these elements have changed

step 2 : eliminate variable x_2 from eq. 3

$$l_{32} = \frac{a_{32}}{a_{22}} \Rightarrow a_{33} \to a_{33} - l_{32}a_{23} \ , \ b_3 \to b_3 - l_{32}b_2$$

$$\left(\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ 0 & a_{22} & a_{23} & b_2 \\ 0 & 0 & a_{33} & b_3 \end{array}\right)$$: upper triangular

matrix form

$$L_2L_1A = U \ , \ L_1 = \begin{pmatrix} 1 & 0 & 0 \\ -l_{21} & 1 & 0 \\ -l_{31} & 0 & 1 \end{pmatrix} \ , \ L_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -l_{32} & 1 \end{pmatrix}$$

general case : $A \in \mathbb{C}^{m\times m}$

$$L_{m-1}\cdots L_1A = U \ , \ L_k = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & -l_{k+1,k} & 1 & \\ & & \vdots & & \ddots \\ & & -l_{m,k} & & 1 \end{pmatrix}$$: unit lower triangular

note

$$(L_{m-1}\cdots L_1)^{-1} \;=\; L = \begin{pmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & \ddots & & \\ \vdots & \vdots & \ddots & \ddots & \\ l_{m1} & l_{m2} & \cdots & l_{m,m-1} & 1 \end{pmatrix} \Rightarrow A = LU$$

proof

$$(L_{m-1}\cdots L_1)^{-1} \;=\; L_1^{-1}\cdots L_{m-1}^{-1}$$

$$L_k \;=\; I - l_k e_k^* \;,\; l_k = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ l_{k+1,k} \\ \vdots \\ l_{m,k} \end{pmatrix} \;,\; e_k = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

$$(I - l_k e_k^*)(I + l_k e_k^*) \;=\; I \;-\; l_k e_k^* \;+\; l_k e_k^* \;-\; l_k \cancelto{0}{e_k^* l_k} e_k^* \;=\; I$$

$$\Rightarrow\; L_k^{-1} \;=\; I \;+\; l_k e_k^*$$

$$L_1^{-1} L_2^{-1} \;=\; (I + l_1 e_1^*)(I + l_2 e_2^*) \;=\; I \;+\; l_1 e_1^* \;+\; l_2 e_2^* \;+\; l_1 \cancelto{0}{e_1^* l_2} e_2^*$$

$$\Rightarrow\; L_1^{-1}\cdots L_{m-1}^{-1} \;=\; I \;+\; l_1 e_1^* \;+\; \cdots \;+\; l_{m-1} e_{m-1}^* \quad \text{ok}$$

note

1. The L, U factors can be stored in A.

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \rightarrow \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ l_{21} & u_{22} & u_{23} \\ l_{31} & l_{32} & u_{33} \end{pmatrix}$$

2. To solve $Ax = b$: factor $A = LU$, solve $Ly = b$, solve $Ux = y$, check ...

algorithm : LU factorization

for $k = 1 : m-1$

  for $j = k+1 : m$

    $a_{jk} = a_{jk} / a_{kk}$

    $a_{j,k+1:m} = a_{j,k+1:m} - a_{jk}\, a_{k,k+1:m}$

operation count

$a_{j,k+1:m}$ has length $m-k \Rightarrow$ inner loop $= 2(m-k)$ flops

total $= \sum_{k=1}^{m-1} \sum_{j=k+1}^{m} 2(m-k) \sim \sum_{k=1}^{m} 2(m-k)^2 \sim \frac{2}{3}m^3$: half of Householder QR factorization

example

$$\begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix} \rightarrow \begin{pmatrix} 2 & -1 & 0 \\ 0 & \frac{3}{2} & -1 \\ 0 & -1 & 2 \end{pmatrix} \rightarrow \begin{pmatrix} 2 & -1 & 0 \\ 0 & \frac{3}{2} & -1 \\ 0 & 0 & \frac{4}{3} \end{pmatrix}$$

$l_{21} = \frac{-1}{2}$ $\qquad$ $l_{32} = \frac{-1}{3/2} = -\frac{2}{3}$

$l_{31} = \frac{0}{2}$

$$\begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ 0 & -\frac{2}{3} & 1 \end{pmatrix} \begin{pmatrix} 2 & -1 & 0 \\ 0 & \frac{3}{2} & -1 \\ 0 & 0 & \frac{4}{3} \end{pmatrix}$$ ok

definition : A has bandwidth $2p+1$ if $a_{ij} = 0$ for $|i-j| > p$, e.g. $p = 1$: tridiagonal

$$\begin{pmatrix} a_{11} & \cdots & a_{1,p+1} & 0 & \cdots & 0 \\ \vdots & \ddots & & \ddots & \ddots & \vdots \\ a_{p+1,1} & & \ddots & & \ddots & 0 \\ 0 & \ddots & & \ddots & & a_{m-p,m} \\ \vdots & \ddots & \ddots & & \ddots & \vdots \\ 0 & \cdots & 0 & a_{m,m-p} & \cdots & a_{mm} \end{pmatrix}$$

note

1. LU factorization of a band matrix preserves the bandwidth (as above for $p = 1$), but sparsity within the band may be lost (more later)

2. for p fixed and $m \rightarrow \infty$, the operation count is $O(mp^2) << O(m^3)$

note

$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$ does not have an LU factorization

proof

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ l_{21} & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{pmatrix} \Rightarrow \left.\begin{array}{l} 0 = u_{11} \\ 1 = l_{21}u_{11} \end{array}\right\} \text{contradiction} \quad \text{ok}$$

theorem

Given A, assume that $\Delta_k = \begin{pmatrix} a_{11} & \cdots & a_{1k} \\ \vdots & & \vdots \\ a_{k1} & \cdots & a_{kk} \end{pmatrix}$ is invertible for $k = 1 : m$.

1. Then there exists L : ult , U : ut , such that $A = LU$.

2. The L, U factors are unique.

proof

1. $a_{11} = \Delta_1 \neq 0$, so we can perform step 1 of Gaussian elimination.

Assume steps $1 : k-1$ have been performed.

$$L_{k-1} \cdots L_1 A = \left(\begin{array}{ccccc:ccc} a_{11}^{(1)} & \cdots & \cdots & a_{1k}^{(1)} & & \cdots & \cdots & a_{1m}^{(1)} \\ & \ddots & & \vdots & & & & \vdots \\ & & \ddots & \vdots & & & & \vdots \\ & & & a_{kk}^{(k)} & & \cdots & \cdots & a_{km}^{(k)} \\ \hdashline & & & \vdots & & & & \vdots \\ & & & \vdots & & & & \vdots \\ & & & a_{mk}^{(k)} & & \cdots & \cdots & a_{mm}^{(k)} \end{array}\right) \quad : \text{ need } a_{kk}^{(k)} \neq 0 \text{ to proceed}$$

$$= \left(\begin{array}{ccccc:ccc} 1 & & & & & & & \\ * & \ddots & & & & & & \\ \vdots & \ddots & \ddots & & & & & \\ * & \cdots & * & 1 & & & & \\ \hdashline \vdots & & \vdots & 0 & \ddots & & & \\ \vdots & & \vdots & \vdots & \ddots & \ddots & & \\ * & \cdots & * & 0 & \cdots & 0 & 1 \end{array}\right) \left(\begin{array}{ccccc:ccc} a_{11} & \cdots & a_{1k} & & & \cdots & a_{1m} \\ \vdots & & \vdots & & & & \vdots \\ \vdots & & \vdots & & & & \vdots \\ a_{k1} & \cdots & a_{kk} & & & & \vdots \\ \hdashline \vdots & & & & & & \vdots \\ \vdots & & & & & & \vdots \\ a_{m1} & \cdots & \cdots & & & \cdots & a_{mm} \end{array}\right)$$

to see this, note :

$$L_2 L_1 = (I - l_2 e_2^*)(I - l_1 e_1^*) = I - l_1 e_1^* - l_2 e_2^* + l_2 e_2^* l_1 e_1^* = I - (l_1 - l_{21} l_2) e_1^* - l_2 e_2^*$$

$$\Rightarrow \begin{pmatrix} a_{11}^{(1)} & \cdots & a_{1k}^{(1)} \\ & \ddots & \vdots \\ & & a_{kk}^{(k)} \end{pmatrix} = \begin{pmatrix} 1 & & & \\ * & \ddots & & \\ \vdots & \ddots & \ddots & \\ * & \cdots & * & 1 \end{pmatrix} \begin{pmatrix} a_{11} & \cdots & a_{1k} \\ \vdots & & \vdots \\ \vdots & & \vdots \\ a_{k1} & \cdots & a_{kk} \end{pmatrix}$$

$\Rightarrow a_{11}^{(1)} \cdots a_{kk}^{(k)} = \det \Delta_k \neq 0$ ok

$a_{kk}^{(k)}$: pivot

2. $A = L_1 U_1 = L_2 U_2 \Rightarrow L_2^{-1} L_1 = U_2 U_1^{-1} = I \Rightarrow L_1 = L_2$, $U_1 = U_2$ ok

21. partial pivoting

theorem

Given A, there exists a permutation matrix P st $PA = LU$.

example

$A = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \Rightarrow P = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, $PA = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} = LU$ ok

proof

If $a_{kk}^{(k)} \neq 0$, proceed as before, i.e. eliminate variable k from eqs. $k+1 : m$.

If $a_{kk}^{(k)} = 0$, there are two possibilities.

1. If $a_{jk}^{(k)} = 0$ for $j = k+1 : m$, then variable k does not appear in eqs. $k : m$, so no elimination is needed ($L_k = I$). Proceed directly to step $k+1$.

2. Otherwise, choose i st $|a_{ik}^{(k)}| = \max\limits_{k \leq j \leq m} |a_{jk}^{(k)}|$, then interchange rows k and i, and proceed with elimination. This is called partial pivoting.

This yields $L_{m-1}P_{m-1} \cdots L_2 P_2 L_1 P_1 A = U$. ... see example below ... ok

example

$A \in \mathbb{C}^{4\times 4} \Rightarrow L_3 P_3 L_2 \underbrace{P_2 L_1} P_1 A = U$, P_k permutes rows k and i for some $i > k$

$$P_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} , \; L_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -l_{21} & 1 & 0 & 0 \\ -l_{31} & 0 & 1 & 0 \\ -l_{41} & 0 & 0 & 1 \end{pmatrix} \Rightarrow P_2 L_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -l_{41} & 0 & 0 & 1 \\ -l_{31} & 0 & 1 & 0 \\ -l_{21} & 1 & 0 & 0 \end{pmatrix}$$

$$P_2 L_1 P_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -l_{41} & 1 & 0 & 0 \\ -l_{31} & 0 & 1 & 0 \\ -l_{21} & 0 & 0 & 1 \end{pmatrix} = \tilde{L}_1 \Rightarrow P_2 L_1 = \tilde{L}_1 P_2$$

$$L_3 \underbrace{P_3 L_2} \tilde{L}_1 P_2 P_1 A = U \Rightarrow L_3 \tilde{L}_2 \underbrace{P_3 \tilde{L}_1} P_2 P_1 A = U \Rightarrow L_3 \tilde{L}_2 \tilde{\tilde{L}}_1 P_3 P_2 P_1 A = U$$ ok

note

1. $Ax = b \Rightarrow PAx = Pb \Rightarrow LUx = Pb$ ...

2. complete pivoting interchanges rows and columns , $PA\tilde{P} = LU$

22. stability of Gaussian elimination

example

$$\left(\begin{array}{cc|c} \epsilon & 1 & 1 \\ 1 & 1 & 2 \end{array}\right) \rightarrow \left(\begin{array}{cc|c} \epsilon & 1 & 1 \\ 0 & 1-\frac{1}{\epsilon} & 2-\frac{1}{\epsilon} \end{array}\right)$$

exact solution : $x_1 = \dfrac{1}{1-\epsilon} = 1 + O(\epsilon)$

$$x_2 = \frac{2-\frac{1}{\epsilon}}{1-\frac{1}{\epsilon}} = \frac{1-2\epsilon}{1-\epsilon} = 1 + O(\epsilon)$$

Now consider the effect of roundoff error.

$$\left(\begin{array}{cc|c} \epsilon & 1 & 1 \\ 0 & -\frac{1}{\epsilon} & -\frac{1}{\epsilon} \end{array}\right) \Rightarrow \text{ computed solution : } \begin{array}{l} \tilde{x}_1 = 0 \\ \tilde{x}_2 = 1 \end{array}$$

The error is $||x - \tilde{x}||_\infty \approx 1 \Rightarrow \tilde{x}$ is inaccurate.

explanation

recall : $Ax = b \Rightarrow Ly = b \ , \ Ux = y$

$$A = \begin{pmatrix} \epsilon & 1 \\ 1 & 1 \end{pmatrix} \ , \ A^{-1} = \frac{1}{\epsilon - 1}\begin{pmatrix} 1 & -1 \\ -1 & \epsilon \end{pmatrix} \Rightarrow \kappa_\infty(A) \approx 4$$

$$U = \begin{pmatrix} \epsilon & 1 \\ 0 & 1-\frac{1}{\epsilon} \end{pmatrix} \ , \ U^{-1} = \frac{1}{\epsilon - 1}\begin{pmatrix} 1-\frac{1}{\epsilon} & -1 \\ 0 & \epsilon \end{pmatrix} \Rightarrow \kappa_\infty(U) \approx \frac{1}{\epsilon^2}$$

Since $\kappa(U) >> \kappa(A)$, a small change in the data can yield a large change in the computed solution, i.e. Gaussian elimination is an unstable algorithm for solving $Ax = b$ (in this example).

remedy : partial pivoting , even if $a_{kk}^{(k)} \neq 0$

$$\left(\begin{array}{cc|c} 1 & 1 & 2 \\ \epsilon & 1 & 1 \end{array}\right) \rightarrow \left(\begin{array}{cc|c} 1 & 1 & 2 \\ 0 & 1-\epsilon & 1-2\epsilon \end{array}\right) \text{ : same exact solution}$$

Consider the effect of roundoff error.

$$\left(\begin{array}{cc|c} 1 & 1 & 2 \\ 0 & 1 & 1 \end{array}\right) \Rightarrow \begin{array}{l} \tilde{x}_1 = 1 \\ \tilde{x}_2 = 1 \end{array} \Rightarrow ||x - \tilde{x}||_\infty = O(\epsilon) \Rightarrow \tilde{x} \text{ is accurate}$$

$$A = \begin{pmatrix} 1 & 1 \\ \epsilon & 1 \end{pmatrix} \Rightarrow \kappa_\infty(A) \approx 4 \ , \ U = \begin{pmatrix} 1 & 1 \\ 0 & 1-\epsilon \end{pmatrix} \Rightarrow \kappa_\infty(U) \approx 4 \quad \text{ok}$$

note

Suppose $A = LU$. Gaussian elimination with IEEE arithmetic yields $\tilde{L}, \tilde{U}$ st $\tilde{L}\tilde{U} = A + \delta A$, where $||\delta A|| \approx ||\tilde{L}|| \cdot ||\tilde{U}|| \cdot O(\epsilon_{\text{m}})$. Partial pivoting ensures that $|\tilde{L}_{ij}| \leq 1$, and if $||\tilde{U}|| \approx ||A||$, then GE + IEEE + PP is backward stable; this is often true in practice.

23. Cholesky factorization

- a symmetric form of LU factorization for hermitian positive definite matrices

recall

$A \in \mathbb{C}^{m \times m}$ is positive definite if $x^*Ax > 0$ for all $x \neq 0$

$$x^*Ax = (\overline{x}_1 \cdots \overline{x}_m) \begin{pmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mm} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} = \sum_{i,j=1}^{m} a_{ij}\overline{x}_i x_j$$

example

$$\begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix} \text{ is positive definite}$$

proof

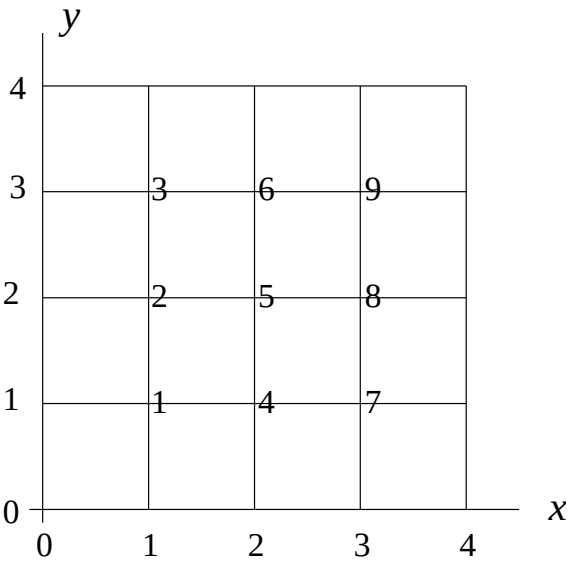
$$\begin{aligned} x^*Ax &= \sum_{i=1}^{m} 2\overline{x}_i x_i - \sum_{i=2}^{m} \overline{x}_i x_{i-1} - \sum_{i=1}^{m-1} \overline{x}_i x_{i+1} \\ &= |x_1|^2 + |x_m|^2 + \sum_{i=2}^{m} \overline{x}_i(x_i - x_{i-1}) + \sum_{i=1}^{m-1} \overline{x}_i(x_i - x_{i+1}) \\ &= |x_1|^2 + |x_m|^2 + \sum_{i=2}^{m} \left(\overline{x}_i(x_i - x_{i-1}) + \overline{x}_{i-1}(x_{i-1} - x_i)\right) \\ &= |x_1|^2 + |x_m|^2 + \sum_{i=2}^{m} (\overline{x}_i - \overline{x}_{i-1})(x_i - x_{i-1}) \\ &= |x_1|^2 + |x_m|^2 + \sum_{i=2}^{m} |x_i - x_{i-1}|^2 > 0 \text{ for } x \neq 0 \quad \text{ok} \end{aligned}$$

two-dimensional BVP

$-(\phi_{xx} + \phi_{yy}) = f$ for $(x, y) \in D \subset \mathbb{R}^2$: Poisson equation

$\phi = g$ for $(x, y) \in \partial D$: Dirichlet boundary condition

example : $D =$ unit square , $h = 1/N$, $(x_i, y_j) = (ih, jh)$, $i, j = 0 : N$



: $h = \dfrac{1}{4}$, lexicographic order

$\phi(x_i, y_j) \approx u_{ij}$, $-\left(D_+^x D_-^x + D_+^y D_-^y\right) u_{ij} = f_{ij}$

$-\frac{1}{h^2}\left(u_{i+1,j} - 2u_{ij} + u_{i-1,j} + u_{i,j+1} - 2u_{ij} + u_{i,j-1}\right) = f_{ij}$

$\frac{1}{h^2}\left(4u_{ij} - u_{i+1,j} - u_{i-1,j} - u_{i,j+1} - u_{i,j-1}\right) = f_{ij}$: 5-point stencil

$(i, j) = (1, 1) \Rightarrow \frac{1}{h^2}\left(4u_{11} - u_{21} - u_{01} - u_{12} - u_{10}\right) = f_{11}$

$\frac{1}{h^2}\left(4u_{11} - u_{21} - u_{12}\right) = f_{11} + \frac{1}{h^2}\left(g_{01} + g_{10}\right)$

1	2	3	4	5	6	7	8	9
u_{11}	u_{12}	u_{13}	u_{21}	u_{22}	u_{23}	u_{31}	u_{32}	u_{33}
4	−1		−1					
−1	4	−1		−1				
	−1	4			−1			
−1			4	−1		−1		
	−1		−1	4	−1		−1	
		−1		−1	4			−1
			−1			4	−1	
				−1		−1	4	−1
					−1		−1	4

$A_h u_h = f_h$

$$A_h = \begin{pmatrix} T & -I & & & \\ -I & T & -I & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -I \\ & & & -I & T \end{pmatrix} \quad : \quad \text{block tridiagonal , } T : \text{tridiagonal}$$

claim : A_h is positive definite

proof : omit

Cholesky factorization

given $A \in \mathbb{C}^{m \times m}$: hermitian , positive definite

$$A = \begin{pmatrix} a_{11} & w^* \\ w & B \end{pmatrix} \quad , \quad a_{11} = e_1^* A e_1 > 0$$

$$= \begin{pmatrix} 1 & 0 \\ \dfrac{w}{a_{11}} & I \end{pmatrix} \begin{pmatrix} a_{11} & w^* \\ 0 & B - \dfrac{ww^*}{a_{11}} \end{pmatrix} \quad : \text{ 1st step of Gaussian elimination}$$

$$= \begin{pmatrix} 1 & 0 \\ \dfrac{w}{a_{11}} & I \end{pmatrix} \begin{pmatrix} a_{11} & 0 \\ 0 & B - \dfrac{ww^*}{a_{11}} \end{pmatrix} \begin{pmatrix} 1 & \dfrac{w^*}{a_{11}} \\ 0 & I \end{pmatrix}$$

$$= \begin{pmatrix} \sqrt{a_{11}} & 0 \\ \dfrac{w}{\sqrt{a_{11}}} & I \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & B - \dfrac{ww^*}{a_{11}} \end{pmatrix} \begin{pmatrix} \sqrt{a_{11}} & \dfrac{w^*}{\sqrt{a_{11}}} \\ 0 & I \end{pmatrix} \quad : \text{ 1st step of Cholesky factorization}$$

$A = R_1^* A_2 R_1$, $R_1 :$ ut , pde

$A_2 = (R_1^*)^{-1} A R_1^{-1}$: hermitian , positive definite , pf : ...

$a_{22}^{(2)} = e_2^* A_2 e_2 > 0$, so we can continue

after m steps we have $A = \underbrace{R_1^* R_2^* \cdots R_m^*}_{R^*} \underbrace{R_m \cdots R_2 R_1}_{R} = R^* R$, $R :$ ut , pde

example

$$\begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix} = \begin{pmatrix} \sqrt{2} & 0 & 0 \\ \frac{-1}{\sqrt{2}} & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{3}{2} & -1 \\ 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} \sqrt{2} & \frac{-1}{\sqrt{2}} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} \sqrt{2} & 0 & 0 \\ \frac{-1}{\sqrt{2}} & \sqrt{\frac{3}{2}} & 0 \\ 0 & -\sqrt{\frac{2}{3}} & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{4}{3} \end{pmatrix} \begin{pmatrix} \sqrt{2} & \frac{-1}{\sqrt{2}} & 0 \\ 0 & \sqrt{\frac{3}{2}} & -\sqrt{\frac{2}{3}} \\ 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} \sqrt{2} & 0 & 0 \\ \frac{-1}{\sqrt{2}} & \sqrt{\frac{3}{2}} & 0 \\ 0 & -\sqrt{\frac{2}{3}} & \sqrt{\frac{4}{3}} \end{pmatrix} \begin{pmatrix} \sqrt{2} & \frac{-1}{\sqrt{2}} & 0 \\ 0 & \sqrt{\frac{3}{2}} & -\sqrt{\frac{2}{3}} \\ 0 & 0 & \sqrt{\frac{4}{3}} \end{pmatrix} \quad , \text{ check } \ldots \quad \underline{\text{ok}}$$

algorithm : Cholesky factorization

$R = A$

for $k = 1 : m$

  $R_{k,k:m} = R_{k,k:m} / \sqrt{R_{kk}}$

  for $j = k+1 : m$

    $R_{j,j:m} = R_{j,j:m} - R_{kj}^* R_{k,j:m}$

note : this is slightly different than the text

operation count

$$\sum_{k=1}^{m} \sum_{j=k+1}^{m} 2(m-j+1) = 2 \sum_{k=1}^{m} \sum_{j=1}^{m-k} j \sim 2 \sum_{k=1}^{m} \frac{(m-k)^2}{2} \sim \tfrac{1}{3} m^3 \text{ flops}$$

This is $\frac{1}{2}$ the work of LU factorization.

note

1. The bandwidth is preserved.
2. There is no need for pivoting because : a) $r_{jj} > 0$,
   b) the algorithm is backward stable.
3. to solve $Ax = b$: factor $A = R^* R$, solve $R^* y = b$, $Rx = y$
4. other versions : a) $A = LL^*$, L : lt , pde
   b) $A = LDL^*$, L : ult , D : diag , pde

24. eigenvalues

problem : given $A \in \mathbb{C}^{m\times m}$, find λ, $x \neq 0$ st $Ax = \lambda x$

e-value : frequency , growth rate , energy level , ...

e-vector : normal mode , ground state , excited state , ...

definition : $p_A(z) = \det(A - zI)$: characteristic polynomial of A

note

λ is an e-value of $A \Leftrightarrow p_A(\lambda) = 0$

algebraic multiplicity of λ = multiplicity of λ as a root of $p_A(z)$

geometric multiplicity of $\lambda = \dim \operatorname{null}(A - \lambda I)$

example

$A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \Rightarrow p_A(z) = \det \begin{pmatrix} 1-z & 1 \\ 0 & 1-z \end{pmatrix} = (1-z)^2 \Rightarrow \lambda = 1$

alg mult $= 2$, geom mult $= \dim \operatorname{null} \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} = 1$

definition

A and B are similar if there exists an invertible matrix X st $A = XBX^{-1}$.

note

Similar matrices have the same characteristic polynomial, the same e-values, and the same algebraic and geometric multiplicities.

definition

A is diagonalizable if it is similar to a diagonal matrix, i.e. if $A = XDX^{-1}$, where D is diagonal. In this case, the e-values of A are the diagonal elements of D and the corresponding e-vectors are the columns of X.

note

A is diagonalizable $\Leftrightarrow \mathbb{C}^m$ has a basis consisting of e-vectors of A

$\Leftrightarrow$ alg mult = geom mult for every e-value

example

$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ is not diagonalizable

definition

A is unitarily diagonalizable if there exists a unitary matrix Q st $A = QDQ^*$, where D is diagonal.

note

A is unitarily diagonalizable

$\Leftrightarrow \mathbb{C}^m$ has an orthonormal basis consisting of e-vectors of A

$\Leftrightarrow A$ is normal , i.e. $A^*A = AA^*$, pf ...

A is hermitian $\Rightarrow A$ is unitarily diagonalizable (spectral factorization)

example

$$A = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \ , \ A^*A = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

$$AA^* = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}\begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \Rightarrow A \text{ is normal}$$

$$A = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{i}{\sqrt{2}} & \frac{-i}{\sqrt{2}} \end{pmatrix}\begin{pmatrix} 1+i & 0 \\ 0 & 1-i \end{pmatrix}\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{-i}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{i}{\sqrt{2}} \end{pmatrix} = QDQ^*$$

Hence A is unitarily diagonalizable, but not hermitian.

definition : A Schur factorization has the form $A = QTQ^*$, where Q is unitary and T is upper triangular.

theorem : Any $A \in \mathbb{C}^{m\times m}$ has a Schur factorization.

proof : induction on m , $m = 1$ is ok , assume true for $m-1$

There exists λ , $x \neq 0$ st $Ax = \lambda x$. Let U be any unitary matrix st $u_1 = x/||x||_2$.

$U^*AU = \begin{pmatrix} \lambda & w^* \\ 0 & B \end{pmatrix}$, by induction hypothesis $B = VTV^*$: Schur factorization

define $Q = U\begin{pmatrix} 1 & 0 \\ 0 & V \end{pmatrix}$: unitary

$$Q^*AQ = \begin{pmatrix} 1 & 0 \\ 0 & V^* \end{pmatrix} U^*AU \begin{pmatrix} 1 & 0 \\ 0 & V \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & V^* \end{pmatrix}\begin{pmatrix} \lambda & w^* \\ 0 & B \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & V \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 \\ 0 & V^* \end{pmatrix}\begin{pmatrix} \lambda & w^*V \\ 0 & BV \end{pmatrix} = \begin{pmatrix} \lambda & w^*V \\ 0 & V^*BV \end{pmatrix} = \begin{pmatrix} \lambda & w^*V \\ 0 & T \end{pmatrix} \quad \text{ok}$$

eigenvalue-revealing factorizations

$A = XJX^{-1}$: Jordan form

$A = XDX^{-1}$: diagonalization

$A = QTQ^*$: Schur form

$A = QDQ^*$: unitary diagonalization

condition number of the e-value problem

example

$A = \begin{pmatrix} 1 & \epsilon^{-1} \\ \epsilon & 1 \end{pmatrix} \Rightarrow p(z) = \det(A - zI) = (1-z)^2 - 1 = z^2 - 2z \Rightarrow \lambda = 2\,,\, 0$

$\tilde{A} = \begin{pmatrix} 1 & \epsilon^{-1} \\ 0 & 1 \end{pmatrix} \Rightarrow \lambda = 1$

Hence a small change in A produced a large change in the e-values, i.e. the e-value problem for A is ill-conditioned.

Bauer-Fike thm

Assume A is diagonalizable with $A = XDX^{-1}$, where $D = \text{diag}(\lambda_1, \ldots, \lambda_m)$, and let α be an e-value of a perturbed matrix $A + \delta A$. Then there exists an index i st $|\alpha - \lambda_i| \leq ||\delta A||_2 \cdot \kappa_2(X)$, i.e. the condition number of the e-value problem for A is $\kappa_2(X)$.

proof : hw7

example

$A = \begin{pmatrix} 1 & \epsilon^{-1} \\ \epsilon & 1 \end{pmatrix} = XDX^{-1}\ ,\ \kappa_2(X) = \epsilon^{-1}$: hw7

note

If A is normal, then $A = QDQ^*$ with $\kappa_2(Q) = 1$, so the e-value problem for A is well-conditioned.

25. overview of algorithms for computing e-values

obvious algorithm : form $p(z) = \det(A - zI)$, solve $p(z) = 0$

example

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \Rightarrow \lambda = 1 \ , \ \tilde{A} = \begin{pmatrix} 1+\epsilon & 0 \\ 0 & 1-\epsilon \end{pmatrix} \Rightarrow \lambda = 1 \pm \epsilon$$

BF $\Rightarrow$ the e-value problem for A is well-conditioned; i.e. a small change in the matrix elements produces a small change in the e-values

$$p_A(z) = (1-z)^2 = z^2 - 2z + 1 \Rightarrow \lambda = 1$$

$$p_{\tilde{A}}(z) = (1+\epsilon-z)(1-\epsilon-z) = z^2 - 2z + 1 - \epsilon^2 \Rightarrow \lambda = 1 \pm \epsilon$$

Note that a small change in the coefficients of the characteristic polynomial produces a large change in the roots, i.e. the problem of finding the roots of the characteristic polynomial is ill-conditioned, and hence the obvious algorithm for computing the e-values of a matrix is unstable.

example (Wilkinson)

smart algorithm : find a stable eigenvalue-revealing factorization

$$\underbrace{Q_j^* \cdots Q_1^*}_{Q^*} A \underbrace{Q_1 \cdots Q_j}_{Q} \rightarrow \begin{cases} D \text{ if } A \text{ is hermitian} \\ T \text{ in general} \end{cases}$$

Each step is a unitary similarity transformation.

hermitian case

$$\begin{pmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{pmatrix} \underset{\rightarrow}{\text{phase 1}} \begin{pmatrix} * & * & & \\ * & * & * & \\ & * & * & * \\ & & * & * \end{pmatrix} \underset{\rightarrow}{\text{phase 2}} \begin{pmatrix} * & & & \\ & * & & \\ & & * & \\ & & & * \end{pmatrix}$$

A tridiagonal D

general case

$$\begin{pmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{pmatrix} \rightarrow \begin{pmatrix} * & * & * & * \\ * & * & * & * \\ & * & * & * \\ & & * & * \end{pmatrix} \rightarrow \begin{pmatrix} * & * & * & * \\ & * & * & * \\ & & * & * \\ & & & * \end{pmatrix}$$

upper Hessenberg T

note : phase 1 is direct , phase 2 is iterative

example (Wilkinson)

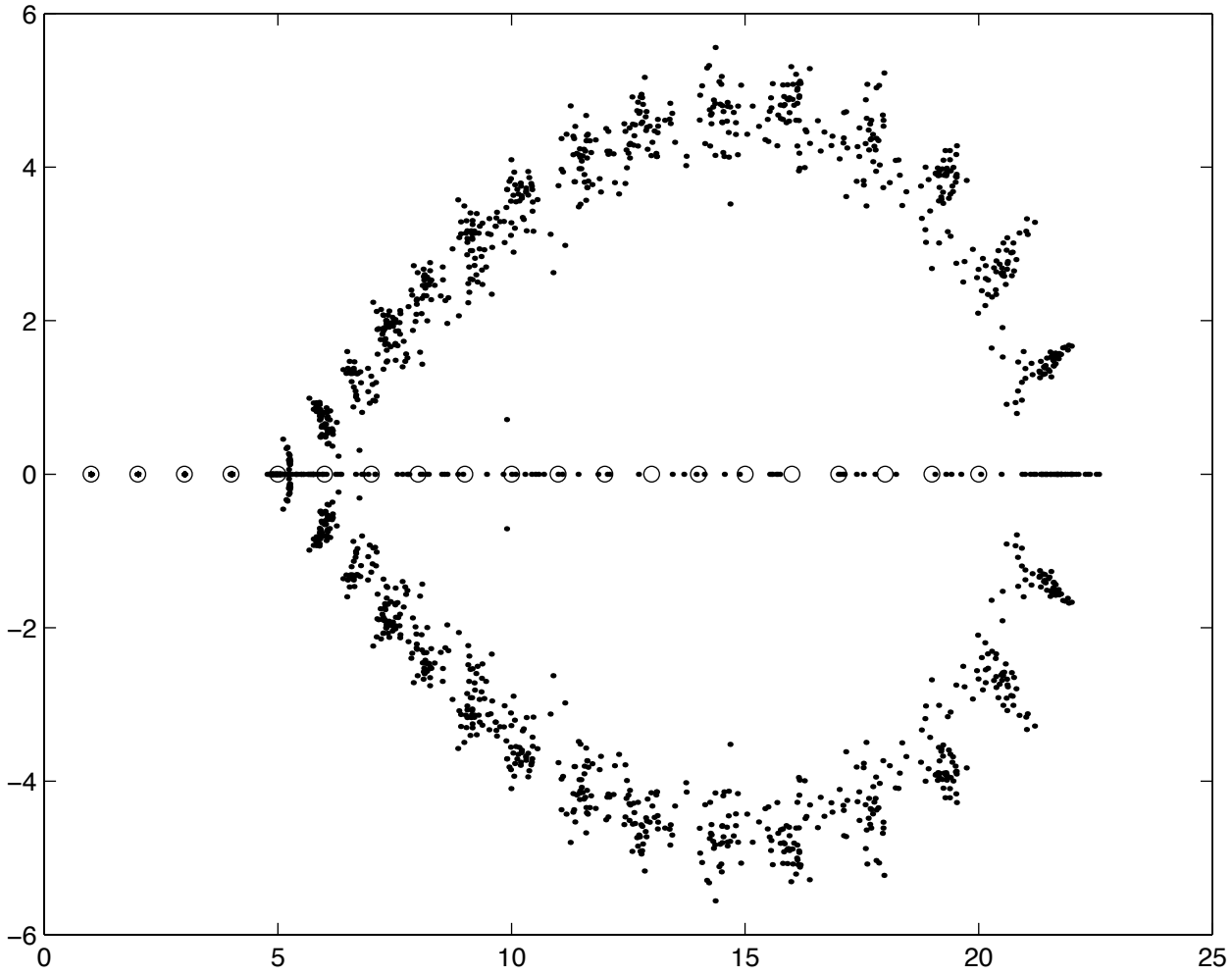
$A = \mathrm{diag}(1, 2, \ldots, 20)$, BF $\Rightarrow$ the e-value problem for A is well-conditioned

$$p(z) = (1-z)(2-z)\cdots(20-z) = \sum_{k=0}^{20} a_k z^k$$

$$\tilde{a}_k = a_k(1 + 10^{-10} r_k)\,,\ 0 \le r_k \le 1 \ :\ \text{random}\ ,\ \tilde{p}(z) = \sum_{k=0}^{20} \tilde{a}_k z^k\ ,\ \text{roots} = ?$$

code

```
plot( zeros(1,20) , 'o' ); hold on;
for i=1:100
  r = roots( poly(1:20) .* (ones(1,21)+1e-10*randn(1,21)) );
  plot( r , '.'  ); axis([0 , 25 , -6 , 6 ]);
end
```



This example shows that the roots of a polynomial can depend very sensitively on the coefficients of the polynomial.

26. reduction to upper Hessenberg or tridiagonal form (phase 1)

idea #1 : try to produce $A = QTQ^*$ as in Householder QR factorization

$$\underbrace{Q_m^* \cdots Q_1^*}_{Q^*} A = R \Rightarrow A = QR\,,\ Q_k = \begin{pmatrix} I & 0 \\ 0 & H \end{pmatrix},\ \begin{array}{l} H = I - 2P\,,\ P = vv^*/||v||_2^2 \\ v = x \pm ||x||_2\, e_1\,,\ Hx = \pm||x||_2\, e_1 \end{array}$$

choose Q_1 as in 1st step of QR factorization

$$Q_1^* A = \begin{pmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{pmatrix} \begin{pmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{pmatrix} = \begin{pmatrix} \boxed{*} & \boxed{*} & \boxed{*} & \boxed{*} \\ 0 & \boxed{*} & \boxed{*} & \boxed{*} \\ 0 & \boxed{*} & \boxed{*} & \boxed{*} \\ 0 & \boxed{*} & \boxed{*} & \boxed{*} \end{pmatrix}, \quad \square : \text{changed}$$

$$Q_1^* A\, Q_1 = \begin{pmatrix} * & * & * & * \\ 0 & * & * & * \\ 0 & * & * & * \\ 0 & * & * & * \end{pmatrix} \begin{pmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{pmatrix} = \begin{pmatrix} \boxed{*} & \boxed{*} & \boxed{*} & \boxed{*} \\ \boxed{*} & \boxed{*} & \boxed{*} & \boxed{*} \\ \boxed{*} & \boxed{*} & \boxed{*} & \boxed{*} \\ \boxed{*} & \boxed{*} & \boxed{*} & \boxed{*} \end{pmatrix} : \text{no progress}$$

idea #2 : try to produce $A = QHQ^*$, where H is upper Hessenberg

choose Q_1 to leave 1st row unchanged

$$Q_1^* A = \left(\begin{array}{c|ccc} 1 & 0 & 0 & 0 \\ \hline 0 & * & * & * \\ 0 & * & * & * \\ 0 & * & * & * \end{array}\right) \left(\begin{array}{c|ccc} * & * & * & * \\ \hline * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{array}\right) = \left(\begin{array}{c|ccc} * & * & * & * \\ \hline \boxed{*} & \boxed{*} & \boxed{*} & \boxed{*} \\ 0 & \boxed{*} & \boxed{*} & \boxed{*} \\ 0 & \boxed{*} & \boxed{*} & \boxed{*} \end{array}\right)$$

$$Q_1^* A\, Q_1 = \left(\begin{array}{c|ccc} * & * & * & * \\ \hline * & * & * & * \\ 0 & * & * & * \\ 0 & * & * & * \end{array}\right) \left(\begin{array}{c|ccc} 1 & 0 & 0 & 0 \\ \hline 0 & * & * & * \\ 0 & * & * & * \\ 0 & * & * & * \end{array}\right) = \left(\begin{array}{c|ccc} * & \boxed{*} & \boxed{*} & \boxed{*} \\ \hline * & \boxed{*} & \boxed{*} & \boxed{*} \\ 0 & \boxed{*} & \boxed{*} & \boxed{*} \\ 0 & \boxed{*} & \boxed{*} & \boxed{*} \end{array}\right) : \text{progress}$$

choose Q_2 to leave 1st two rows unchanged

$$Q_2^*\, Q_1^* A\, Q_1 = \left(\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & * & * \\ 0 & 0 & * & * \end{array}\right) \left(\begin{array}{cc|cc} * & * & * & * \\ * & * & * & * \\ \hline 0 & * & * & * \\ 0 & * & * & * \end{array}\right) = \left(\begin{array}{cc|cc} * & * & * & * \\ * & * & * & * \\ \hline 0 & \boxed{*} & \boxed{*} & \boxed{*} \\ 0 & 0 & \boxed{*} & \boxed{*} \end{array}\right)$$

$$Q_2^*\, Q_1^* A\, Q_1 Q_2 = \left(\begin{array}{cc|cc} * & * & * & * \\ * & * & * & * \\ \hline 0 & * & * & * \\ 0 & 0 & * & * \end{array}\right) \left(\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & * & * \\ 0 & 0 & * & * \end{array}\right) = \left(\begin{array}{cc|cc} * & * & \boxed{*} & \boxed{*} \\ * & * & \boxed{*} & \boxed{*} \\ \hline 0 & * & \boxed{*} & \boxed{*} \\ 0 & 0 & \boxed{*} & \boxed{*} \end{array}\right) \quad \underline{\text{ok}}$$

summary

$$A \in \mathbb{C}^{m\times m} \;\Rightarrow\; \underbrace{Q_{m-2}^* \cdots Q_1^*}_{Q^*} A \underbrace{Q_1 \cdots Q_{m-2}}_{Q} \;=\; H$$

algorithm : reduction to upper Hessenberg form

for $k = 1 : m-2$

$x = A_{k+1:m,k}$

$v_k = x + \text{sign}(x_1)||x||_2\, e_1$

$v_k = v_k/||v_k||_2$

$A_{k+1:m,k:m} = A_{k+1:m,k:m} - 2v_k(v_k^* A_{k+1:m,k:m})$

$A_{1:m,k+1:m} = A_{1:m,k+1:m} - 2(A_{1:m,k+1:m}v_k)v_k^*$

operation count

$$\left.\begin{array}{l} \text{1st inner loop} \;:\; \sum_{k=1}^{m-2} 4(m-k)^2 \sim 4\cdot\frac{1}{3}m^3 \\ \text{2nd inner loop} \;:\; \sum_{k=1}^{m-2} 4m(m-k) \sim 4m\cdot\frac{1}{2}m^2 = 2m^3 \end{array}\right\} \text{total} = \tfrac{10}{3}\, m^3 \text{ flops}$$

note

1. The algorithm is backward stable.

2. If A is hermitian, the algorithm reduces A to tridiagonal form and the operation count can be reduced to $\frac{4}{3}m^3$ flops. , pf : hw7

---

27. basic e-value algorithms (phase 2)

1. For the rest of the discussion on e-values, we assume A is real symmetric.

$\lambda_1, \ldots, \lambda_m$: e-values , real

$q_1, \ldots, q_m$: e-vectors , orthonormal

2. Given an approximate e-vector x, we can compute an approximate e-value $\hat{\alpha}$.

$||Ax - \hat{\alpha}x||_2 = \min_{\alpha} ||Ax - \alpha x||_2 \quad : \quad ||b - A\hat{x}||_2 = \min_{x} ||b - Ax||_2$

$x^T x\, \hat{\alpha} = x^T A x \quad : \quad A^T A\hat{x} = A^T b$

$$\Rightarrow\; \hat{\alpha} = \frac{x^T A x}{x^T x} = R_A(x) \;:\; \text{Rayleigh quotient} \;,\; R_A(q_j) = \lambda_j$$

error estimate

$R_A(x) = R_A(q_j) + \nabla R_A(q_j) \cdot (x - q_j) + O(||x - q_j||^2)$

$\nabla R_A(x) = \nabla \left(\dfrac{x^T A x}{x^T x} \right) = \dfrac{x^T x \cdot \nabla(x^T A x) - x^T A x \cdot \nabla(x^T x)}{(x^T x)^2}$

$\nabla(x^T x) = \nabla(x_1^2 + x_2^2) = (2x_1, 2x_2) = 2x^T$

$x^T A x = a_{11}x_1^2 + 2a_{12}x_1x_2 + a_{22}x_2^2$

$\nabla(x^T A x) = (2a_{11}x_1 + 2a_{12}x_2\,, 2a_{12}x_1 + 2a_{22}x_2) = 2(Ax)^T$

$\nabla R_A(x) = \dfrac{x^T x \cdot 2(Ax)^T - x^T A x \cdot 2x^T}{(x^T x)^2} = \dfrac{2}{x^T x}((Ax)^T - R_A(x)x^T)$

$\Rightarrow\ \nabla R_A(q_j) = 0$

$R_A(x) = \lambda_j + O(||x - q_j||^2)$: quadratic approximation

---

power method : $v\,,\ Av\,,\ A^2v\,,\ \ldots$

$v^{(0)}$: given , $||v^{(0)}||_2 = 1$

for $k = 1, 2, \ldots$

$w = Av^{(k-1)}$

$v^{(k)} = w/||w||_2$

$\lambda^{(k)} = (v^{(k)})^T A v^{(k)}$

example : $A = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 4 \end{pmatrix}$, $v^{(0)} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \cdot \dfrac{1}{\sqrt{3}}$,

k	$\lambda^{(k)}$
0	5.0
1	5.181818
2	5.208[illegible]1921
	4320 $= \lambda_{\max}$

theorem

Suppose $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_m|$ and $q_1^T v^{(0)} \neq 0$.

Then $||v^{(k)} - (\pm q_1)|| = O\left(\left|\dfrac{\lambda_2}{\lambda_1}\right|^k\right)$, $|\lambda^{(k)} - \lambda_1| = O\left(\left|\dfrac{\lambda_2}{\lambda_1}\right|^{2k}\right)$.

proof : $v^{(0)} = \alpha_1 q_1 + \alpha_2 q_2 + \cdots + \alpha_m q_m$

$v^{(k)} = \beta_k A^k v^{(0)} = \beta_k \left(\alpha_1 \lambda_1^k q_1 + \alpha_2 \lambda_2^k q_2 + \cdots + \alpha_m \lambda_m^k q_m\right)$

$= \beta_k \alpha_1 \lambda_1^k \left(q_1 + \dfrac{\alpha_2}{\alpha_1}\left(\dfrac{\lambda_2}{\lambda_1}\right)^k q_2 + \cdots + \dfrac{\alpha_m}{\alpha_1}\left(\dfrac{\lambda_m}{\lambda_1}\right)^k q_m \right)$

$\Rightarrow v^{(k)} \to \pm q_1$, the $\pm$ depends on sign(λ_1) ok

note : The power method has some limitations.

1. it gives only the largest e-value

2. $v^{(k)}$, $\lambda^{(k)}$ converge linearly and the convergence factor $\left|\frac{\lambda_2}{\lambda_1}\right|$ may not be small

idea : 1. apply power method to A^{-1} : $w = A^{-1}v \Leftrightarrow Aw = v$

2. ..... " ..... $(A-\mu I)^{-1}$ $\begin{cases} \text{e-vectors are } q_j \\ \text{e-values are } (\lambda_j - \mu)^{-1} \end{cases}$

$\uparrow$ shift

shifted inverse iteration

$v^{(0)}$: given , $||v^{(0)}||_2 = 1$

for $k = 1, 2, \ldots$

solve $(A-\mu I)w = v^{(k-1)}$

$v^{(k)} = w/||w||_2$

$\lambda^{(k)} = (v^{(k)})^T A v^{(k)}$

theorem

Suppose $|\lambda_J - \mu| < |\lambda_K - \mu| \leq |\lambda_i - \mu|$ for $i \neq J$, i.e. λ_J is the e-value of A closest to μ and λ_K is the next closest. Assume also that $q_J^T v^{(0)} \neq 0$.

Then $||v^{(k)} - (\pm q_J)|| = O\left(\left|\frac{\lambda_J - \mu}{\lambda_K - \mu}\right|^k\right)$, $|\lambda^{(k)} - \lambda_J| = O\left(\left|\frac{\lambda_J - \mu}{\lambda_K - \mu}\right|^{2k}\right)$.

proof

as before , $\lambda_1 \to \frac{1}{\lambda_J - \mu}$, $\lambda_2 \to \frac{1}{\lambda_K - \mu}$, $\left|\frac{\lambda_2}{\lambda_1}\right| \to \left|\frac{\lambda_J - \mu}{\lambda_K - \mu}\right|$ ok

Hence by using a suitable shift μ, any e-value of A can be obtained, and the convergence factor can be made small.

question : What is the effect of roundoff error?

claim : If $(A-\mu I)w = v$ is solved by a backward stable method, then $||w - \tilde{w}|| = \kappa(A-\mu I) \cdot O(\epsilon_{\text{m}})$. Hence if μ is close to λ_J, then $A - \mu I$ is ill-conditioned and the error in $\tilde{w}$ can be large, but $\tilde{v}^{(k)}$ is still a good approximation to q_J.

proof

suppose $(A-\mu I)^{-1}\tilde{w} = \tilde{v}$, where $||v - \tilde{v}|| = O(\epsilon_m)$

$(A-\mu I)(\tilde{w} - w) = \tilde{v} - v = \alpha_1 q_1 + \cdots + \alpha_m q_m$

$\Rightarrow \tilde{w} - w = \alpha_1(\lambda_1 - \mu)^{-1} q_1 + \cdots + \alpha_m(\lambda_m - \mu)^{-1} q_m \approx \alpha_J(\lambda_J - \mu)^{-1} q_J$ ok

Rayleigh quotient iteration : update μ

$v^{(0)}$: given , $||v^{(0)}||_2 = 1$

$\lambda^{(0)} = (v^{(0)})^T A v^{(0)}$

for $k = 1, 2, \ldots$

solve $(A - \lambda^{(k-1)} I) w = v^{(k-1)}$

$v^{(k)} = w / ||w||_2$

$\lambda^{(k)} = (v^{(k)})^T A v^{(k)}$

theorem

If $v^{(0)}$ is sufficiently close to an e-vector q_J, then

$$\left.\begin{array}{l} ||v^{(k+1)} - (\pm q_J)|| = O(||v^{(k)} - (\pm q_J)||^3) \\ |\lambda^{(k+1)} - \lambda_J| = O(|\lambda^{(k)} - \lambda_J|^3) \end{array}\right\} : \text{cubic convergence}.$$

proof

$$(A - \lambda^{(k)} I) w = v^{(k)} \Rightarrow w = (A - \lambda^{(k)} I)^{-1} (q_J + v^{(k)} - q_J)$$

$$\approx \frac{1}{\lambda_J - \lambda^{(k)}} q_J + \frac{1}{\lambda_K - \lambda^{(k)}} (v^{(k)} - q_J)$$

$$v^{(k+1)} = \frac{w}{||w||_2} \approx q_J + \frac{\lambda_J - \lambda^{(k)}}{\lambda_K - \lambda_J} (v^{(k)} - q_J)$$

$$||v^{(k+1)} - q_J|| = O(|\lambda_J - \lambda^{(k)}| \cdot ||v^{(k)} - q_J||) = O(||v^{(k)} - q_J||^3)$$

$$|\lambda^{(k+1)} - \lambda_J| = O(||v^{(k+1)} - q_J||^2) = O(||v^{(k)} - q_J||^6) = O(|\lambda^{(k)} - \lambda_J|^3) \quad \text{ok}$$

example

$$A = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 4 \end{pmatrix} \ , \ \lambda_1 = 5.214319743377 \ , \ v^{(0)} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \cdot \frac{1}{\sqrt{3}}$$

k	power method	shifted inverse iteration , $\mu = 5$	Rayleigh quotient iteration
0	5.0	5.0	5.0
1	5.181818	5.213114	5.213114
2	5.208192	5.214312617	5.214319743184
	2	6	10

operation count per step

A : full

power : $O(m^2)$

inverse : $O(m^2)$ if $A - \mu I = LU$ is precomputed

Rayleigh : $O(m^3)$ if $A - \lambda^{(k)} I = LU$ is computed at each step

A : tridiagonal

$O(m)$: this shows the benefit of phase 1

28, 29. QR algorithm

algorithm

$A^{(0)} = A$

for $k = 1, 2, \ldots$

$A^{(k-1)} = Q^{(k)} R^{(k)}$

$A^{(k)} = R^{(k)} Q^{(k)}$

note

1. $A^{(k)} = R^{(k)} Q^{(k)} = (Q^{(k)})^T Q^{(k)} R^{(k)} Q^{(k)} = (Q^{(k)})^T A^{(k-1)} Q^{(k)}$: unitary similarity transformation

2. $A^{(k)} \to \begin{cases} D \text{ if } A \text{ is normal} \\ T \text{ in general} \end{cases}$

theorem

Assume A is real symmetric, $A = QDQ^T$, $D = \text{diag}(\lambda_i)$, $|\lambda_1| > \cdots > |\lambda_m| > 0$, $Q = [q_1 \cdots q_m]$, $\det \Delta_k(Q) \neq 0$ for $k = 1 : m$. Then $A^{(k)} \to D$.

example: $A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = QR \Rightarrow A^{(k)} = A$ for all k , thm doesn't apply

proof

$A^{(k)} = (Q^{(k)})^T A^{(k-1)} Q^{(k)}$

$= (Q^{(k)})^T \cdots (Q^{(1)})^T A^{(0)} Q^{(1)} \cdots Q^{(k)} = (\underline{Q}^{(k)})^T A \underline{Q}^{(k)}$, $\underline{Q}^{(k)} = Q^{(1)} \cdots Q^{(k)}$

we will show that $\underline{Q}^{(k)} \to Q$, so that $A^{(k)} \to Q^T A Q = D$

$A = A^{(0)} = Q^{(1)} R^{(1)}$

$A^2 = A \cdot A = Q^{(1)} R^{(1)} Q^{(1)} R^{(1)} = Q^{(1)} Q^{(2)} R^{(2)} R^{(1)}$

$A^3 = A \cdot A^2 = Q^{(1)} R^{(1)} Q^{(1)} Q^{(2)} R^{(2)} R^{(1)} = Q^{(1)} Q^{(2)} R^{(2)} Q^{(2)} R^{(2)} R^{(1)}$

$= Q^{(1)} Q^{(2)} Q^{(3)} R^{(3)} R^{(2)} R^{(1)}$

$A^k = Q^{(1)} \cdots Q^{(k)} R^{(k)} \cdots R^{(1)} = \underline{Q}^{(k)} \underline{R}^{(k)}$

$A^k = \underline{Q}^{(k)} \underline{R}^{(k)} = [A^k e_1 \cdots A^k e_m]$: simultaneous power iteration

$\frac{A^k e_j}{||A^k e_j||_2} \to q_1$ for $j = 1 : m$, but $\text{span}(A^k e_1, \ldots, A^k e_m) = \begin{cases} \text{span}(\underline{q}_1^{(k)}, \ldots, \underline{q}_m^{(k)}) \\ \text{span}(q_1, \ldots, q_m) \end{cases}$

We will show that $\underline{q}_j^{(k)} \to q_j$ for $j = 1, 2$, but it is true for $j = 1 : m$.

$e_1 = \alpha_{11} q_1 + \alpha_{12} q_2 + \cdots + \alpha_{1m} q_m$

$\alpha_{11} = e_1^T q_1 = \det \Delta_1(Q) \neq 0$

$q_1 = \frac{1}{\alpha_{11}} e_1 - \left(\frac{\alpha_{12}}{\alpha_{11}} q_2 + \cdots + \frac{\alpha_{1m}}{\alpha_{11}} q_m \right)$

$q_1 = \beta_{11} e_1 + \beta_{12} q_2 + \cdots + \beta_{1m} q_m \ , \ \beta_{11} \neq 0$

$A^k q_1 = \lambda_1^k q_1 = \beta_{11} A^k e_1 + \beta_{12} \lambda_2^k q_2 + \cdots + \beta_{1m} \lambda_m^k q_m$

$\Rightarrow \ q_1 = \beta_{11} \frac{\underline{r}_{11}^{(k)} \underline{q}_1^{(k)}}{\lambda_1^k} + \beta_{12} \left(\frac{\lambda_2}{\lambda_1} \right)^k q_2 + \cdots + \beta_{1m} \left(\frac{\lambda_m}{\lambda_1} \right)^k q_m$

$\Rightarrow \ \underline{q}_1^{(k)} \to \ q_1$

$e_2 = \alpha_{21} q_1 + \alpha_{22} q_2 + \alpha_{23} q_3 + \cdots + \alpha_{2m} q_m$

$= \alpha_{21} \left(\frac{1}{\alpha_{11}} e_1 - \left(\frac{\alpha_{12}}{\alpha_{11}} q_2 + \cdots + \frac{\alpha_{1m}}{\alpha_{11}} q_m \right) \right) + \alpha_{22} q_2 + \alpha_{23} q_3 + \cdots + \alpha_{2m} q_m$

$= \frac{\alpha_{21}}{\alpha_{11}} e_1 + \frac{\alpha_{11}\alpha_{22} - \alpha_{21}\alpha_{12}}{\alpha_{11}} q_2 + \tilde{\alpha}_{23} q_3 + \cdots + \tilde{\alpha}_{2m} q_m$

$\alpha_{11}\alpha_{22} - \alpha_{21}\alpha_{12} = \det \begin{pmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{pmatrix} = \det \begin{pmatrix} e_1^T q_1 & e_1^T q_2 \\ e_2^T q_1 & e_2^T q_2 \end{pmatrix} = \det \Delta_2(Q) \neq 0$

$q_2 = \beta_{21} e_1 + \beta_{22} e_2 + \beta_{23} q_3 + \cdots + \beta_{2m} q_m \ , \ \beta_{22} \neq 0$

$A^k q_2 = \lambda_2^k q_2 = \beta_{21} A^k e_1 + \beta_{22} A^k e_2 + \beta_{23} \lambda_3^k q_3 + \cdots + \beta_{2m} \lambda_m^k q_m$

$\Rightarrow \ q_2 = \beta_{21} \frac{\underline{r}_{11}^{(k)} \underline{q}_1^{(k)}}{\lambda_2^k} + \beta_{22} \frac{\underline{r}_{12}^{(k)} \underline{q}_1^{(k)} + \underline{r}_{22}^{(k)} \underline{q}_2^{(k)}}{\lambda_2^k} + \beta_{23} \left(\frac{\lambda_3}{\lambda_2} \right)^k q_3 + \cdots + \beta_{2m} \left(\frac{\lambda_m}{\lambda_2} \right)^k q_m$

$\Rightarrow \ \underline{q}_2^{(k)} \to q_2$

$\Rightarrow \ \underline{Q}^{(k)} \to Q \ \Rightarrow A^{(k)} = (\underline{Q}^{(k)})^T A \, \underline{Q}^{(k)} \to \ Q^T A \, Q = D \quad \underline{\text{ok}}$

example

$$A = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 4 \end{pmatrix}, \quad A^{(5)} = \begin{pmatrix} 5.2138 & -0.0358 & -0.0153 \\ -0.0358 & 2.4308 & 0.1837 \\ -0.0153 & 0.1837 & 1.3554 \end{pmatrix} \quad \begin{matrix} \text{exact } \lambda_i \\ 5.2143 \\ 2.4608 \\ 1.3249 \end{matrix}$$

more details of QR algorithm

1. importance of phase 1

A : tridiagonal $\Rightarrow A^{(k)}$: tridiagonal , pf : hw7

2. deflation

If $|A^{(k)}_{j,j+1}| < \epsilon$, set $A^{(k)} = \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix}$ and continue iterating on A_1, A_2.

3. Convergence can be accelerated by shifting.

QR algorithm with shifts

$A^{(0)} = A$

for $k = 1, 2, \ldots$

$$A^{(k-1)} - \mu^{(k-1)} I = Q^{(k)} R^{(k)}$$
$$A^{(k)} = R^{(k)} Q^{(k)} + \mu^{(k-1)} I$$

$$A^{(5)} = \begin{pmatrix} 1.3261 & 0.0371 & 0.0000 \\ 0.0371 & 2.4596 & 0.0000 \\ 0.0000 & 0.0000 & 5.2143 \end{pmatrix}$$

How to choose $\mu^{(k)}$?

$A^{-k} = (\underset{\sim}{Q}^{(k)} \underline{R}^{(k)})^{-1} = (\underline{R}^{(k)})^{-1} (\underset{\sim}{Q}^{(k)})^T = \underset{\sim}{Q}^{(k)} (\underline{R}^{(k)})^{-T} = \underset{\sim}{Q}^{(k)} \underline{L}^{(k)}$

$A^{-k} = \underset{\sim}{Q}^{(k)} \underline{L}^{(k)} = [A^{-k} e_1 \cdots A^{-k} e_m]$: simultaneous inverse iteration

$$\frac{A^{-k} e_i}{||A^{-k} e_i||_2} \to q_m \text{ for } i = 1 : m$$

$\mu^{(k)} = R_A(\underset{\sim}{q}^{(k)}_m)$: we expect this to be a good choice

$= (\underset{\sim}{q}^{(k)}_m)^T A \underset{\sim}{q}^{(k)}_m = (\underset{\sim}{Q}^{(k)} e_m)^T A (\underset{\sim}{Q}^{(k)} e_m) = e_m^T (\underset{\sim}{Q}^{(k)})^T A \underset{\sim}{Q}^{(k)} e_m = e_m^T A^{(k)} e_m = A^{(k)}_{mm}$

---

30. other e-value algorithms

31. computing the svd

32. overview of iterative methods

$Ax = b\ ,\ A \in \mathbb{C}^{m\times m}\ ,\ b \in \mathbb{C}^m$

direct methods

LU, Cholesky, QR : $O(m^3)$ flops

iterative methods

$x_0\ , x_1\ , x_2, \ldots \to x$: may be faster than direct methods

classical : Jacobi , Gauss-Seidel , SOR

$Ax = b \Leftrightarrow x = Bx + c\ ,\ x_{n+1} = Bx_n + c$: fixed-point iteration

modern : multigrid , Krylov

---

33. Arnoldi iteration : for computing e-values or solving linear systems

given $A \in \mathbb{C}^{m\times m}$: general , $b \in \mathbb{C}^m$

recall : reduction to upper Hessenberg form

$\underbrace{Q_{m-2}^* \cdots Q_1^*}_{Q^*} A \underbrace{Q_1 \cdots Q_{m-2}}_{Q} = H \Rightarrow A = QHQ^*$, Q_k : Householder reflector

note : computing the columns of Q requires extra work

alternative

$AQ = QH\ ,\ Q = [q_1 \cdots q_m]$, these q_j are not the e-vectors of A

$Aq_1 = h_{11}q_1 + h_{21}q_2$ % $q_1 \to Aq_1 \to h_{11} \to h_{21}q_2 \to h_{21}, q_2$

$Aq_2 = h_{12}q_1 + h_{22}q_2 + h_{32}q_3$ % $q_2 \to Aq_2 \to h_{12}, h_{22} \to h_{32}q_3 \to h_{32}, q_3$

$\cdots$

$Aq_n = h_{1n}q_1 + h_{2n}q_2 + \cdots + h_{nn}q_n + h_{n+1,n}q_{n+1}$

$\cdots$

$Aq_m = h_{1m}q_1 + h_{2m}q_2 + \cdots + h_{mm}q_m$

Arnoldi iteration : computes $h_{jn}\ , q_j$

b : given , $q_1 = b/||b||_2$

for $n = 1, 2, \ldots$ % in practice, stop before $n = m$

  $v = Aq_n$

  for $j = 1 : n$

    $h_{jn} = q_j^* v$

    $v = v - h_{jn}q_j$

  $h_{n+1,n} = ||v||_2$

  $q_{n+1} = v/h_{n+1,n}$ % if $h_{n+1,n} = 0$: breakdown , later

note

1. Arnoldi iteration for computing $A = QHQ^*$ resembles classical Gram-Schmidt for computing $A = QR$.

2. Each step in Arnoldi iteration requires a matrix-vector product Aq_n and it produces a new orthonormal vector q_{n+1}.

3. Let $Q_n = [q_1 \cdots q_n] \in \mathbb{C}^{m\times n}$. (not a Householder reflector)

$$AQ_n = Q_{n+1}\tilde{H}_n \ , \ \tilde{H}_n = \begin{pmatrix} h_{11} & h_{12} & \cdots & \cdots & h_{1n} \\ h_{21} & h_{22} & \cdots & \cdots & h_{2n} \\ & h_{32} & \ddots & & \vdots \\ & & \ddots & \ddots & \vdots \\ & & & h_{n,n-1} & h_{nn} \\ & & & & h_{n+1,n} \end{pmatrix} \in \mathbb{C}^{(n+1)\times n}$$

alternative view of Arnoldi iteration

definition

$\mathcal{K}_n = \text{span}(b, Ab, A^2b, \ldots, A^{n-1}b)$: Krylov subspace

$K_n = [b\,, Ab\,, A^2b\,, \cdots, A^{n-1}b]$: Krylov matrix , $\mathbb{C}^{m\times n}$

note

1. $\text{range}\, K_n = \mathcal{K}_n$

2. $\mathcal{K}_1 \subseteq \mathcal{K}_2 \subseteq \cdots \subseteq \mathcal{K}_n \ , \ n \le m$

3. $\mathcal{K}_1 = \text{span}(b) = \text{span}(q_1) = \text{range}\, Q_1$

   $\mathcal{K}_2 = \text{span}(b, Ab) = \text{span}(q_1, q_2) = \text{range}\, Q_2$

   $\cdots$

   $\mathcal{K}_n = \text{span}(b, Ab, \ldots, A^{n-1}b) = \text{span}(q_1, q_2, \ldots, q_n) = \text{range}\, Q_n$

Hence the Arnoldi vectors $q_1, \ldots, q_n$ form an orthonormal basis for the Krylov subspace $\mathcal{K}_n$.

4. $K_n = Q_n R_n$: QR factorization, but K_n, R_n are not formed explicitly

5. $AQ_n = Q_{n+1}\tilde{H}_n \Rightarrow Q_n^* A Q_n = \underbrace{Q_n^* Q_{n+1}}_{I_{n\times(n+1)}} \tilde{H}_n = H_n \in \mathbb{C}^{n\times n}$

   : last column is zero

$$H_n = \begin{pmatrix} h_{11} & h_{12} & \cdots & \cdots & h_{1n} \\ h_{21} & h_{22} & \cdots & \cdots & h_{2n} \\ & h_{32} & \ddots & & \vdots \\ & & \ddots & \ddots & \vdots \\ & & & h_{n,n-1} & h_{nn} \end{pmatrix} \; : \text{ upper Hessenberg}$$

6. $Q_n Q_n^*$: orthogonal projector onto range $Q_n = \mathcal{K}_n$, $Q_n^* Q_n = I_{n\times n}$

$Q_n : \mathbb{C}^n \to \mathbb{C}^m$, $e_i \to Q_n e_i = q_i$, $i = 1 : n$

$Q_n^* : \mathbb{C}^m \to \mathbb{C}^n$, $q_i \to Q_n^* q_i = e_i$, $i = 1 : n$

$\Rightarrow$ think of Q_n, Q_n^* as change of basis matrices

note : $A \to Q_n Q_n^* A \to Q_n^* Q_n Q_n^* A Q_n = Q_n^* A Q_n = H_n$

Hence H_n is the orthogonal projection of A onto $\mathcal{K}_n$ represented in the basis $\{q_1, \ldots, q_n\}$ and we expect that $A \approx H_n$ in some sense.

7. (problem 33.2) Suppose $h_{n+1,n} = 0$ for some n. : breakdown

a) $Aq_n = h_{1n}q_1 + \cdots + h_{nn}q_n + \cancel{h_{n+1,n}q_{n+1}} \in \mathcal{K}_n$

b) recall : $\mathcal{K}_n = \text{span}(b, Ab, \ldots, A^{n-1}b) = \text{span}(q_1, q_2, \ldots, q_n)$

$\Rightarrow A\mathcal{K}_n = \text{span}(Ab, A^2b, \ldots, A^n b) = \text{span}(Aq_1, Aq_2, \ldots, Aq_n) \subseteq \mathcal{K}_n$, check ...

$\Rightarrow A\mathcal{K}_n \subseteq \mathcal{K}_n$: $\mathcal{K}_n$ is an <u>invariant subspace</u> of A

c) $\mathcal{K}_1 \subseteq \cdots \subseteq \mathcal{K}_n = \mathcal{K}_{n+1} = \mathcal{K}_{n+2} = \cdots$, check : $A^{n+1}b = AA^n b \in A\mathcal{K}_n \subseteq \mathcal{K}_n$

note : breakdown $\Rightarrow AQ_n = Q_{n+1}\tilde{H}_n = Q_n H_n$ (will use later)

now choose q_{n+1} orthogonal to $q_1, \ldots, q_n$ st $||q_{n+1}||_2 = 1$ and continue iterating

$Aq_{n+1} = h_{1,n+1}q_1 + \cdots + h_{n+1,n+1}q_{n+1} + h_{n+2,n+1}q_{n+2}$: this defines q_{n+2}

$\Rightarrow A = QHQ^*$, $H = \begin{pmatrix} H_n & * \\ 0 & * \end{pmatrix}$: upper Hessenberg with $h_{n+1,n} = 0$

d) An e-value of H_n is also an e-value of H and hence also of A.

<u>proof</u>: assume $H_n x = \lambda x$, $x \neq 0$, $x \in \mathbb{C}^n$, define $y = \begin{pmatrix} x \\ 0 \end{pmatrix}$, $y \neq 0$, $y \in \mathbb{C}^m$

then $Hy = \begin{pmatrix} H_n & * \\ 0 & * \end{pmatrix}\begin{pmatrix} x \\ 0 \end{pmatrix} = \begin{pmatrix} H_n x \\ 0 \end{pmatrix} = \begin{pmatrix} \lambda x \\ 0 \end{pmatrix} = \lambda y$ <u>ok</u>

e) If $Ax = b$ and A is invertible, then $x \in \mathcal{K}_n$.

proof : since A is invertible , it follows that H, H_n are also invertible

note : $b = Q_n Q_n^* b = Q_n H_n H_n^{-1} Q_n^* b = A Q_n H_n^{-1} Q_n^* b$

set $x = Q_n H_n^{-1} Q_n^* b$, then $Ax = b$

set $y = H_n^{-1} Q_n^* b$, then $x = Q_n y \in \text{range}\, Q_n = \mathcal{K}_n$ ok

Hence when Arnoldi iteration is applied to computing e-values or solving linear systems, breakdown means we can stop iterating.

---

34. application of Arnoldi iteration to computing e-values

$H_n = Q_n^* A Q_n \in \mathbb{C}^{n \times n}$: approximate unitary similarity transformation

The e-values of H_n approximate the e-values of A.

---

35. application of Arnoldi iteration to solving $Ax = b$

$A \in \mathbb{C}^{m \times m}$: general , $b \in \mathbb{C}^m$

idea

$p(z) = \det(A - zI) = \alpha_0 + \alpha_1 z + \alpha_2 z^2 + \cdots + \alpha_m z^m$

$p(A) = \alpha_0 I + \alpha_1 A + \alpha_2 A^2 + \cdots + \alpha_m A^m = 0$: Cayley-Hamilton thm

$$A^{-1} = \frac{-1}{\alpha_0}\left(\alpha_1 I + \alpha_2 A + \cdots + \alpha_m A^{m-1}\right) \ , \ \alpha_0 = p(0) = \det A$$

$$A^{-1} b = \frac{-1}{\alpha_0}\left(\alpha_1 b + \alpha_2 A b + \cdots + \alpha_m A^{m-1} b\right) \in \mathcal{K}_m = \mathbb{C}^m$$

$A^{-1} b \approx x_n \in \mathcal{K}_n$, where $||Ax_n - b||_2 = \min\limits_{x \in \mathcal{K}_n} ||Ax - b||_2$: generalized ls problem

This is the generalized minimum residual method. : GMRES

naive approach

$x \in \mathcal{K}_n \Rightarrow x = K_n y$, where $y \in \mathbb{C}^n$

$\min\limits_{x \in \mathcal{K}_n} ||Ax - b||_2 = \min\limits_{y \in \mathbb{C}^n} ||AK_n y - b||_2$: standard ls problem

solve for $\hat{y}$, e.g. by QR factorization of AK_n , set $x_n = K_n \hat{y}$

disadvantage : computing AK_n is expensive and numerically unstable

smart approach

$K_n = Q_n R_n$: Arnoldi iteration computes Q_n cheaply and stably without computing K_n, R_n explicitly.

$x \in \mathcal{K}_n \Rightarrow x = Q_n y$, $y \in \mathbb{C}^n$

$$\min_{x \in \mathcal{K}_n} ||Ax - b||_2 = \min_{y \in \mathbb{C}^n} ||AQ_n y - b||_2 = \min_{y \in \mathbb{C}^n} ||Q_{n+1}\tilde{H}_n y - b||_2$$

$$= \min_{y \in \mathbb{C}^n} ||Q_{n+1}^* Q_{n+1}\tilde{H}_n y - Q_{n+1}^* b||_2 = \min_{y \in \mathbb{C}^n} ||\tilde{H}_n y - Q_{n+1}^* b||_2$$

1. $v \in \text{range}\, Q_{n+1} \Rightarrow ||Q_{n+1}^* v||_2^2 = v^* Q_{n+1} Q_{n+1}^* v = v^* v = ||v||_2^2$

2. $Q_{n+1}^* Q_{n+1} = I_{(n+1)\times(n+1)}$

3. $Q_{n+1}^* b = ||b||_2 e_1$, $e_1 \in \mathbb{C}^{n+1}$

$$\min_{x \in \mathcal{K}_n} ||Ax - b||_2 = \min_{y \in \mathbb{C}^n} ||\tilde{H}_n y - ||b||_2 e_1||_2$$

algorithm : GMRES

$q_1 = b/||b||_2$

for $n = 1, 2, \ldots$

perform step n of Arnoldi iteration to create $\tilde{H}_n$, q_{n+1}

find $\hat{y}$: $||\tilde{H}_n \hat{y} - ||b||_2 e_1||_2 = \min_{y \in \mathbb{C}^n} ||\tilde{H}_n y - ||b||_2 e_1||_2$

$x_n = Q_n \hat{y}$

note

1. assume ls problem is solved by Householder QR factorization

$AQ_n \in \mathbb{C}^{m \times n}$: $2mn^2 - \frac{2}{3}n^3$ flops

$\tilde{H}_n \in \mathbb{C}^{(n+1)\times n}$: $O(n^2)$ flops and can be reduced to $O(n)$ by updating

2. Each step requires computing Aq_n.

convergence analysis of GMRES

$x \in \mathcal{K}_n = \text{span}(b, Ab, \ldots, A^{n-1}b)$

$x = \alpha_0 b + \alpha_1 Ab + \cdots + \alpha_{n-1} A^{n-1} b$

$= q(A)b$, where $q(z)$ is a polynomial of degree $\leq n - 1$

$r = b - Ax = b - Aq(A)b = (I - Aq(A))b = p(A)b$, $p(z) = 1 - zq(z)$

define : $\mathcal{P}_n = \{\, p(z) : p \text{ is a polynomial of degree} \le n \text{ st } p(0) = 1\}$

$||Ax_n - b||_2 = \min\limits_{x \in \mathcal{K}_n} ||Ax - b||_2 = \min\limits_{p \in \mathcal{P}_n} ||p(A)b||_2$

1. $\mathcal{K}_n \subseteq \mathcal{K}_{n+1} \Rightarrow ||r_{n+1}||_2 \le ||r_n||_2$, $r_n = b - Ax_n$

2. $\mathcal{K}_m = \mathbb{C}^m \Rightarrow ||r_m||_2 = 0$

3. $||r_n||_2 = \min\limits_{p \in \mathcal{P}_n} ||p(A)b||_2 \le \min\limits_{p \in \mathcal{P}_n} ||p(A)||_2 \cdot ||b||_2$

4. assume A is diagonalizable , $A = XDX^{-1}$

$p(A) = p(XDX^{-1}) = Xp(D)X^{-1}$

$||p(A)||_2 = ||Xp(D)X^{-1}||_2 \le ||X||_2 \cdot ||p(D)||_2 \cdot ||X^{-1}||_2$

$||r_n||_2 \le \kappa_2(X) \cdot ||b||_2 \cdot \min\limits_{p \in \mathcal{P}_n} \max\limits_{\lambda \in \mathrm{sp}(A)} |p(\lambda)|$

note : If $\mathrm{sp}(A)$ is <u>clustered away from the origin</u>, then there exists $p \in \mathcal{P}_n$ st $\max\limits_{\lambda \in \mathrm{sp}(A)} |p(\lambda)|$ decays rapidly as n increases, and GMRES converges rapidly.

<u>example</u>: $A \in \mathbb{R}^{200 \times 200}$ (pictures on pp. 272-273)

<u>case 1</u> : $\mathrm{sp}(A)$ is clustered away from the origin

$|\lambda - 2| \le \frac{1}{2}$ for all $\lambda \in \mathrm{sp}(A)$, consider $p(z) = \left(1 - \frac{z}{2}\right)^n \in \mathcal{P}_n$

$\max\limits_{\lambda \in \mathrm{sp}(A)} |p(\lambda)| \le \left(\frac{1}{4}\right)^n \Rightarrow$ rapid convergence

<u>case 2</u> : $\mathrm{sp}(A)$ is <u>not</u> clustered away from the origin $\Rightarrow$ slow convergence

<u>36, 37. Lanczos iteration</u>

$A \in \mathbb{R}^{m \times m}$, symmetric , Arnoldi $\rightarrow$ Lanczos : $A = QTQ^*$, T : tridiagonal

3-term recurrence , Gaussian quadrature , ...

<u>38. conjugate gradient method</u>

$A \in \mathbb{R}^{m \times m}$, symmetric , positive definite , ...

<u>39. biorthogonalization methods</u>

$Ax = b$, A : general , use Krylov subspaces for A and A^*

<u>40. preconditioning</u>

$Ax = b \rightarrow M^{-1}Ax = M^{-1}b$, for example $M = \mathrm{diag}A$

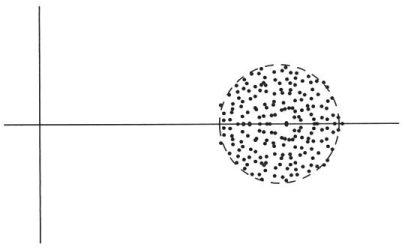


Figure 35.2. *Eigenvalues of the* 200×200 *matrix* A *of* (35.17). *The dashed curve is the circle of radius* $1/2$ *with center* $z = 2$ *in* $\mathbb{C}$. *The eigenvalues are approximately uniformly distributed within this disk.*

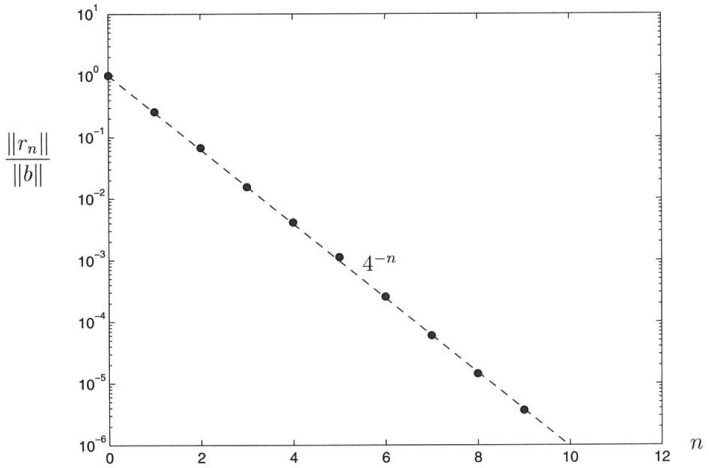


Figure 35.3. *GMRES convergence curve for the same matrix* A. *This rapid, steady convergence is illustrative of Krylov subspace iterations under ideal circumstances, when* A *is a well-behaved (or well-preconditioned) matrix.*

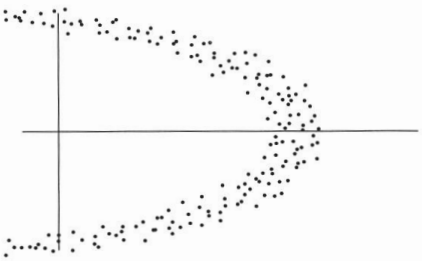


Figure 35.4. *Eigenvalues of a* 200×200 *matrix, like that of* (35.17) *except with a modified diagonal. Now the eigenvalues surround the origin on one side.*

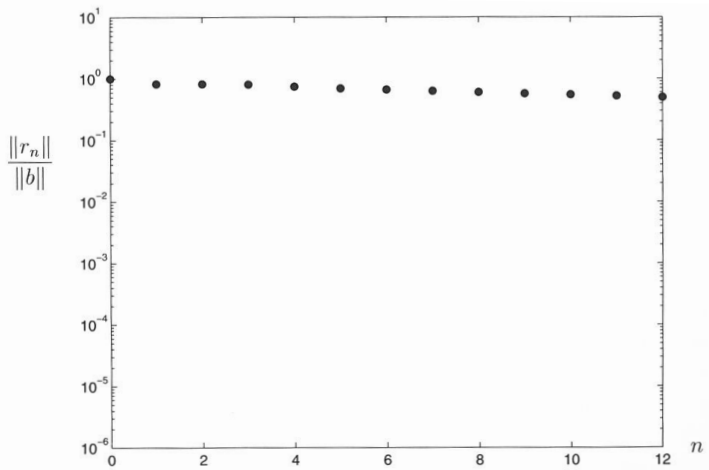


Figure 35.5. *GMRES convergence curve for the matrix of Figure* 35.4. *The convergence has slowed down greatly. When an iterative method stagnates like this, it is time to look for a better preconditioner.*