Computació Numèrica

Notes de classe

Tema 2. Àlgebra Lineal Numèrica

M. Àngela Grau Gotés

10 de febrer de 2023

Índex

1 Guia d'estudi

L'objectiu principal del tema és l'estudi de mètodes computacionals bàsics per a l'àlgebra lineal. La primera part d'aquest tema es refereix a mètodes computacionals per a la resolució de sistemes d'equacions lineals, la segona part al càlcul de valors propis i de singulars d'una matriu.

- Resolució de sistemes lineals no homogenis.
  - Mètodes directes: eliminació gaussiana, mètode de Gauss-Jordan, descompossició LU, factorització QR.
  - Mètodes iteratius: Jacobi, Gauss-Seidel i sobrerelaxació (SOR).
  - Mínims quadrats.
- Càlcul de vectors i valors propis.
  - Mètodes de la potència.
  - Mètode QR.
  - Valors singulars.

1.1 Glosari

- Càlcul de $||x||_1\,, ||x||_2\,, ||x||_\infty$ per a vectors.
- Càlcul de $||A||_1\,, ||A||_2\,, ||A||_\infty$ per a matrius.
- Concepte de matriu simètrica i matriu ortogonal.
- Concepte de radi espectral d'una matriu.
- Concepte de nombre de condició d'una matriu.
- Mètode de Cramer per a resoldre sistemes. Eficiència

- Nombre de condició. Vector residu. Fites de l'error
- Sistema d'equacions ben condicionat/mal condicionat
- Mètode de Gauss per a resoldre sistemes. Eficiència
- Pivotar. Estratègies.
- És necessari implementar estratègies de pivotament en la resolució de sistemes d'equacions lineals?
- Mètode compacte de resolució de sistemes. Cita exemples
- Factorització LU. Sempre es pot obtenir?
- Mètode iteratiu de resolució de sistemes d'equacions lineals
- Mètodes iteratius. Mètode de Jacobi. Mètode de Gauss - Seidel. Mètode SOR.
- Mètode de Jacobi - Matrius del mètode de Jacobi.
- Mètode de Gauss-Seidel. Matrius del mètode de Gauss-Seidel.
- Precondicionament
- Sistemes d'equacions sobredeterminats. Equacions normals.
- Sistemes sobredeterminats. Equacions normals. Mètode dels mínims quadrats.
- Valors i vectors propis.
- Mètode de la potència.
- Mètode QR.
- Valors singulars. Matriu Pseudoinversa.
- Descomposició en valors singulars (SVD)

1.2 Referències

1.2.1 Llibres de consulta online

[1] Accès UPCommons, Càlcul numèric: teoria i pràctica

Capítol 4, Sistemes lineals: Conceptes associats: pàgines 117-143. Problemes proposats: 1-9,13. Pràctiques resoltes: pàgines 147-154.

Capítol 7, Valors i vectors propis: Conceptes associats: pàgines 250-297. Problemes proposats: 1-6,8. Pràctiques resoltes: pàgines 298-304. Pràctiques proposades: pàgines 305-309.

[2] Accès UPCommons, Cálculo numérico (Edició en castellà)

[3] Accès lliure SIAM, Barrett, R., M. Berry, T. F. Chan, et al., Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, SIAM, Philadelphia, 1994.

[4] Accès lliure netlib.org, Barrett, R., M. Berry, T. F. Chan, et al., *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, SIAM, Philadelphia, 1994.*

[5] Accès Biblioteca, Cálculo Científico con MATLAB y Octave by A. Quarteroni, F. Saleri

Capítulo 5, Sistemas lineales: Conceptes i exercicis resolts: pàgines 127-170. Problemes i pràctiques proposades: del 5.1 al 5.17

Capítulo 6, Autovalores y autovectores: Conceptes i exercicis resolts: pàgines 173-190.

- Problemes i pràctiques proposades: del 6.1 al 6.10

[6] Accès lliure MathWorks, Cleve Moler - Llibre de text i codis
Numerical Computing with MATLAB®

[7] Catàleg bibilioteca UPC. Richard L. Burden, Douglas J. Faires, Annette M. Burden, Análisis Numérico, 10a edición. Ed.Cengage Learning, 2017.

[8] Catàleg bibilioteca UPC. James E. Gentle, Matrix Algebra Theory, Computations, and Applications in Statistics. Springer Texts in Statistics

[9] Lloc web, Holistic Numerical Methods Topics of Numerical Methods
Chapter 04 Simultaneous Linear Equations

1.2.2 Videos i lectures de comprensió del tema

- Eficiència
  1.- TOP500. The list
  2.- TOP500, rànquing dels 500 ordinadors més potents del món
  3.- Flops
  4.- LINPACK benchmarks
  5.- LAPACK
  6.- List of numerical libraries
- Algorismes
  1.- How Ordinary Elimination Became Gaussian Elimination
- Cas històric
  1.- Pont Tacoma Narrows 1940
  2.- Enfonsament del Pont Tacoma Narrows, Washington. Video original
  3.- Enfonsament del Pont Tacoma Narrows, Washington. Video
- Enginyeria Informàtica
  1.- The Anatomy of a Large-Scale Hypertextual Web Search Engine Autors: Sergey Brin and Lawrence Page
  2.- Algoritme PageRank, Wikipedia
  3.- El secreto de Google y el álgebra lineal
  4.- La descomposición en valores singulares (SVD) y algunas de sus aplicaciones
- MATLAB®
  1.- Documentació de MATLAB® - Sistemes d'equacions lineals
  2.- Documentació de MATLAB® - Factoritzacions
  3.- Documentació de MATLAB® - Iterative Methods for Linear Systems
  4.- Documentació de MATLAB® - Valores propios
  5.- Documentació de MATLAB® - Valores singulares

2 Sistemes d'equacions lineals

2.1 Introducció

Qualsevol sistema de m equacions lineals amb n incògnites,

$$\left\{\begin{array}{rcl} a_{11}x_1 + a_{12}x_2 + \ldots a_{1n}x_n & = & b_1\,, \\ a_{21}x_1 + a_{22}x_2 + \ldots a_{2n}x_n & = & b_2\,, \\ \vdots & & \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots a_{mn}x_n & = & b_m\,. \end{array}\right.$$

es pot representar per una matriu $A = (a_{ij})$ que recull els coeficients de les incògnites, el vector d'incógnites $x = (x_i)^t$, i el vector $b = (b_i)^t$, vector terme independent de tantes components com equacions $(_{1 \le i \le m,\ 1 \le j \le n})$: $\mathbf{A}\,\mathbf{x} = \mathbf{b}$.

Resoldre $\mathbf{A}\,\mathbf{x} = \mathbf{b}$ és un dels problemes fonamentals de la matemàtica aplicada. En aquest tema introduïm algunes **tècniques directes** i alguns **mètodes iteratius** per aproximar la solució d'un sistema d'equacions lineals.

Les tècniques directes (sense tenir en compte els errors d'arrodoniment) obtenen la solució exacte en un nombre finit d'operacions aritmètiques. Aquestes tècniques es basen en esència en el mètode conegut d'eliminació de Gauss, però acumulent error d'arrodoniment, encara que veurem alguna alternativa mitjançant l'ús de transformacions ortogonals en el cas de sistemes sobredeterminats.

Els mètodes iteratius consistiran en generar una successió de vectors que convergeixin, teòricament, a la solució exacta: el nombre de pasos per assolir la solució exacte és infinit. La solució aproximada pateix d'entrada un error de truncament. Aquesta tècnica s'empra per a sistemes lineals dispersos d'ordre n gran ja que per a sistemes grans, *mal condicionats* els errors d'arrodoniment dels mètodes directesens poden donar una solució sense sentit,

Com a conseqüència de treballar amb ordinadors i amb una aritmètica de punt flotant de precisió limitada, tant els mètodes directes com els mètodes iteratius es proporcionen una aproximació $\hat{\mathbf{x}}$ de la solució $\mathbf{x}$; una primera qüestió és decidir si $\hat{\mathbf{x}}$ i $\mathbf{x}$ estan relativament a prop.

2.1.1 Existència de solucions

Per a resoldre el sistema d'equacions $\mathbf{A}\,\mathbf{x} = \mathbf{b}$, es crea la matriu augmentada:

$$[A\,,b] \equiv \left[\begin{array}{cccc|c} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \vdots & \vdots & & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} & b_m \end{array}\right]$$

Hi ha sistemes sobredeterminats, amb més equacions que incògnites ($m > n$), hi ha sistemes no determinats, de menys equacions que incògnites ($m < n$) i sistemes amb el mateix nombre d'equacions que incògnites ($m = n$).

Teorema 2.1.1 (Teorema de Rouche-Frobenius) *En el cas $n = m$ l'existència de solucions del sistema lineal $\mathbf{Ax} = \mathbf{b}$ depèn del rang de la matriu A i la matriu $[A\,,b]$:*

- *Si $rang(A) = rang([A\,,b]) = r$ el sistema és compatible. Si $r = n$ el sistema té solució única. Si $r < n$ el sistema és indeterminat, amb $n - r$ graus de llibertat.*
- *Si $rang(A) \neq rang([A\,,b])$ el sistema és incompatible.*

La Regla de Cramer s'aplica en el cas de solució és única, $n = m$ i $det(A) = |A| \neq 0$.

Teorema 2.1.2 (Mètode de Cramer) *La solució única de $Ax = b$ és:*

$$x_i = \frac{|A_i|}{|A|}\,, \quad 1 \leq i \leq n \tag{2.1.1}$$

on $|A|$ és del determinant de la matriu a, i $|A_i|$ és el determinant de la matriu A_i obtinguda substituint la columna i de la matriu A pel vector b.

Example 2.1.1

La solució del sistema lineal $\begin{aligned} 2x_1 + 4x_2 + x_3 &= 1 \\ 8x_1 - x_2 + 3x_3 &= 0 \\ 2x_1 + 5x_2 \qquad &= 0 \end{aligned}$ es calcularia per:

$$x_1 = \frac{\begin{vmatrix} 1 & 4 & 1 \\ 0 & -1 & 3 \\ 0 & 5 & 0 \end{vmatrix}}{\begin{vmatrix} 2 & 4 & 1 \\ 8 & -1 & 3 \\ 2 & 5 & 0 \end{vmatrix}} = \frac{-15}{36}\,, \qquad x_2 = \frac{\begin{vmatrix} 2 & 1 & 1 \\ 8 & 0 & 3 \\ 2 & 0 & 0 \end{vmatrix}}{\begin{vmatrix} 2 & 4 & 1 \\ 8 & -1 & 3 \\ 2 & 5 & 0 \end{vmatrix}} = \frac{6}{36}\,, \qquad x_3 = \frac{\begin{vmatrix} 2 & 4 & 1 \\ 8 & -1 & 0 \\ 2 & 5 & 0 \end{vmatrix}}{\begin{vmatrix} 2 & 4 & 1 \\ 8 & -1 & 3 \\ 2 & 5 & 0 \end{vmatrix}} = \frac{42}{36}\,.$$

Una fórmula que ens proporciona la solució, però que es totalment inservible per obtenir la solució per l'elevat nombre d'operacions aritmètiques que involucra.

Eficiència Si la matriu és d'ordre n, observem que

- calen $n+1$ determinants d'ordre n per a calcular el vector solució $\mathbf{x}$ i n divisions.
- cada determinant d'ordre n requereix $n!-1$ sumes i $n!n$ productes, pel cap baix $n!(n+1)-1$.
- un total de $(n!(n+1)-1)(n+1)+n = n!(n+1)^2-1$. Per $n=10$ el total aproximat és 10^8 operacions.

Observeu la taula següent: temps emprat en resoldre un sistema lineal d'ordre n en un ordinador depenen del nombre de flops[1] de la CPU de l'ordinador emprat per resoldre'l:

	flops				
n	10^9 (Giga)	10^{10}	10^{11}	10^{12} (Tera)	10^{15} (Peta)
10	10^{-1} s	10^{-2} s	10^{-3} s	10^{-4} s	despreciable
15	17 h	1.74 h	1046 m	1 m	$0.6\,10^{-1}$ s
20	4860 anys	486 anys	48.6 anys	4.86 anys	1.7 dia
25	fora de l'abast	fora de l'abast	fora de l'abast	fora de l'abast	38365 dies

Eficiència - Cost computacional - Cost operacional El mètode de Cramer és un mètode inapropiat per obtenir la solució d'un sistema lineal per mitjà de l'ordinador degut a l'alt nombre d'operacions aritmètiques a realitzar.

El nostre principal objectiu és obtenir mètodes computacionals per a la resolució de sistemes d'equacions lineals dins de l'abast de càlcul d'un ordinador comercial.

2.1.2 Llibreries LINPACK, LAPACK i HPL

Els Linpack benchmarks es basen en LINPACK, una biblioteca de programari per l'àlgebra lineal numèrica en Fortran escrita en Fortran per Jack Dongarra, Jim Bunch, Cleve Moler i Gilbert Stewart a la dècada dels 70, s'empren per mesurar la rapidesa amb què un ordinador resol sistema d'equacions lineals $Ax=b$, d'ordre n i A matriu plena, una tasca comuna en enginyeria.

[1] En computació, FLOPS (de l'anglès Floating Point Operations Per Second) és una unitat que es sol utilitzar per mesurar els càlculs matemàtics que pot fer per segon una CPU i/o una GPU.

En l'actualitat, els HPL benchmarcks són una mesura de la potència de càlcul de coma flotant d'un sistema. L'última versió d'aquests benchmarks s'utilitza per construir la llista TOP500, classificant els superordinadors més potents del món.

2.2 Mètodes directes

Els mètodes directes són mètodes computacionals que ens donen la solució del sistema lineal $\mathbf{A}\,\mathbf{x} = \mathbf{b}$, en un nombre finit d'operacions, si no fos pels errors d'arrodoniment acumulats i les possibles imprecisions en el coneixement inicial de la matriu de coeficients i el terme independent. Es consideren adients per a sistemes lineals no massa grans $(100 - 500$ equacions) i amb pocs elements nuls.

Millor algoritme En tots els algoritmes caldrà considerar:

- el temps emprat per obtenir la solució (mesurat en nombre d'operacions). De res serveix un mètode que obtingui la solució en un temps clarament excesiu.
- els errors d'arrodoniment del mètode de càlcul.

Primer presentem algorismes molt econòmics computacionalment, i finalment discutirem com afecten els errors d'arrodoniment a la solució obtinguda.

2.2.1 Matriu de coeficients diagonal

Si la matriu de coeficients del sistema lineal $\mathbf{A}\,\mathbf{x} = \mathbf{b}$ és diagonal, per obtenir el vector solució $\mathbf{x}$ calen n divisions en punt flotant.

Definició 1 (Matriu diagonal) *$D = (d_{ij})$ tal que $d_{ij} = 0$ si $i \neq j$ per a tot $1 \leq i, j \leq n$*

$$D = \begin{pmatrix} d_{11} & 0 & \dots & 0 \\ 0 & d_{22} & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & d_{nn} \end{pmatrix}$$

Determinant Per una matriu diagonal d'ordre n el determinant és $det(A) = d_{11}d_{22}\dots d_{nn}$

Teorema 2.2.1 *Un sistema lineal diagonal d'ordre n, té solució si i només si $d_{ii} \neq 0$ per a tot $1 \leq i \leq n$. En aquest cas per obtenir el vector solució $\mathbf{x}$ calen n divisions en punt flotant:*

$$[D\,,b] = \left[\begin{array}{cccc|c} d_{11} & 0 & \dots & 0 & b_1 \\ 0 & d_{22} & \dots & 0 & b_2 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & d_{nn} & b_n \end{array}\right] \quad \Longrightarrow \quad \text{La solució és} \quad x_i = \frac{b_i}{d_{ii}}\,,\ 1 \leq i \leq n\,.$$

2.2.2 Matriu de coeficients triangular

Si la matriu de coeficients A és triangular, el càlcul de $det(A)$ i la resolució del sistema lineal $\mathbf{A}\,\mathbf{x} = \mathbf{b}$ es pot obtenir mitjançant algorismes d'un nombre baix d'operacions aritmètiques en punt flotant.

Definició 2 (Matriu triangular superior) *$U = (u_{ij})$ tal que $u_{ij} = 0$ si $1 \leq j < i \leq n$ i en aquest cas el determinant és $det(U) = u_{11}u_{22}\dots u_{nn}$.*

$$U = \begin{pmatrix} u_{11} & u_{12} & u_{13} & \dots & u_{1n} \\ 0 & u_{22} & u_{23} & \dots & u_{2n} \\ 0 & 0 & u_{33} & \dots & u_{3n} \\ \vdots & \vdots & \ddots & \ddots & \\ 0 & 0 & \dots & 0 & u_{nn} \end{pmatrix} \qquad L = \begin{pmatrix} l_{11} & 0 & 0 & \dots & 0 \\ l_{21} & l_{22} & 0 & \dots & 0 \\ l_{31} & l_{32} & l_{33} & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 \\ l_{n1} & l_{n2} & l_{n3} & \dots & l_{nn} \end{pmatrix}$$

Definició 3 (Matriu triangular inferior) *$L = (l_{ij})$ tal que $l_{ij} = 0$ si $1 \leq i < j \leq n$ i en aquest cas el determinant és $det(L) = l_{11}l_{22}\dots l_{nn}$.*

Teorema 2.2.2 *Un sistema lineal triangular superior d'ordre n, té solució si i només si $u_{ii} \neq 0$ per a tot $1 \leq i \leq n$. En aquest cas per obtenir el vector solució s'aplica l'algorisme de **substitució enrera**. Les fórmules són*

$$\left[\begin{array}{ccccc|c} u_{11} & u_{12} & u_{13} & \dots & u_{1n} & b_1 \\ 0 & u_{22} & u_{23} & \dots & u_{2n} & b_2 \\ 0 & 0 & u_{33} & \dots & u_{3n} & b_3 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \\ 0 & 0 & \dots & 0 & u_{nn} & b_n \end{array}\right] \Rightarrow \begin{array}{l} x_n = \dfrac{b_n}{u_{nn}}\,, \\ x_i = \dfrac{1}{u_{ii}}\left(b_i - \displaystyle\sum_{k=i+1}^{n} u_{ik}x_k\right), \\ 1 \leq i < n. \end{array} \tag{2.2.1}$$

Teorema 2.2.3 *Un sistema lineal triangular inferior d'ordre n, té solució si i només si $ll_{ii} \neq 0$ per a tot $1 \leq i \leq n$. En aquest cas per obtenir el vector solució s'aplica l'algorisme de **substitució endavant**. Les fórmules són*

$$\left[\begin{array}{ccccc|c} l_{11} & 0 & 0 & \dots & 0 & b_1 \\ l_{21} & l_{22} & 0 & \dots & 0 & b_2 \\ l_{31} & l_{32} & l_{33} & \ddots & \vdots & b_3 \\ \vdots & \vdots & \vdots & \ddots & 0 & \\ l_{n1} & l_{n2} & l_{n3} & \dots & l_{nn} & b_n \end{array}\right] \Rightarrow \begin{array}{l} x_1 = \dfrac{b_1}{l_{11}}\,, \\ x_i = \dfrac{1}{l_{ii}}\left(b_i - \displaystyle\sum_{k=1}^{i-1} l_{ik}x_k\right), \\ 1 \leq i < n. \end{array} \tag{2.2.2}$$

Eficiència Per avaluar l'eficiència dels algorismes per resoldre sistemes triangulars, recomptem el nombre d'operacions aritmètiques que cal realitzar perobtenir la solució.

Teorema 2.2.4 (Cost computacional per a matrius triangulars)
Per obtenir la solució del sistema lineal $\mathbf{A}\,\mathbf{x} = \mathbf{b}$ *calen* $\mathbf{n^2}$ *operacions aritmètiques.*
Per calular el determinant de la matriu de coeficients $\mathbf{A}$ *calen* $\mathbf{n-1}$ *productes.*

Demostració 2.2.1 *El recompte és:*

- *Cal* 1 *divisió per calcular* $x_i\,,\quad 1 \le i < n$, *en total* n *divisions.*
- *En l'expressió* $x_i = \dfrac{1}{l_{ii}}\left(b_i - \displaystyle\sum_{k=1}^{i-1} l_{ik}x_k\right)$ *efectuem* $i-1$ *productes i* $i-1$ *sumes, en total* $\displaystyle\sum_{i=2}^{n}(i-1) = \frac{n(n-1)}{2}$ *productes i el mateix nombre de sumes*[2]*.*
- *En l'expressió* $x_i = \dfrac{1}{u_{ii}}\left(b_i - \displaystyle\sum_{k=i+1}^{n} u_{ik}x_k\right)$ *efectuem* $n-i$ *productes i* $n-i$ *sumes, en total* $\displaystyle\sum_{i=1}^{n-1}(n-i) = \frac{n(n-1)}{2}$ *sumes i el mateix nombre de productes.*
- *Si acumulem les sumes, els productes i les divisions obtenim un total de* n^2 *operacions.*

	$+/-$	$*$	$/$	total
Diagonal	0	0	n	n
Trian. sup.	$\frac{n^2-n}{2}$	$\frac{n^2-n}{2}$	n	n^2
Trian. inf.	$\frac{n^2-n}{2}$	$\frac{n^2-n}{2}$	n	n^2

Taula 2.1: *Nombre d'operacions resoldre* $\mathbf{A}\,\mathbf{x} = \mathbf{b}$ *. Observar la reducció del cost respecte el mètode de Cramer aprofitant el fet que la matriu de coeficients és triangular.*

2.2.3 Matriu de coeficients plena

El primer mètode que es presenta en Àlgebra Lineal per resoldre $\mathbf{A}\,\mathbf{x} = \mathbf{b}$, acostuma a ser aquell en el que s'eliminen les incògnites mitjançant la combinació de les equacions. Aquest mètode es coneix com a mètode d'eliminació (o reducció).

[2]Recordar que $\displaystyle\sum_{k=1}^{m} 1 = m\,,\quad \sum_{k=1}^{m} k = \frac{(m+1)m}{2}$

Mètode d'eliminació de Gauss (MEG) Consta de dues fases. La primera fase consisteix en modificar el nostre sistema d'equacions per arribar a un sistema triangular superior equivalent. En la segona fase es resol el sistema triangular superior obtingut en la primera fase fent ús de *sustitució enrera.*

$$\left[\begin{array}{ccccc|c} a_{11} & a_{12} & a_{13} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} & b_2 \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} & b_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} & b_n \end{array}\right] \quad \rightsquigarrow \quad \left[\begin{array}{ccccc|c} u_{11} & u_{12} & u_{13} & \dots & u_{1n} & \widetilde{b}_1 \\ 0 & u_{22} & u_{23} & \dots & u_{2n} & \widetilde{b}_2 \\ 0 & 0 & u_{33} & \dots & u_{3n} & \widetilde{b}_3 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \\ 0 & 0 & \dots & 0 & u_{nn} & \widetilde{b}_n \end{array}\right]$$

El sistema $\mathbf{A}\,\mathbf{x} = \mathbf{b}$ es transforma en un sistema triangular $\mathbf{U}\,\mathbf{x} = \mathbf{c}$ equivalent[3].

Sobre la matriu ampliada $[A\,, b]$ només es permeten dues operacions:

- Substituir una fila per ella mateixa més un múltiple d'una altra fila. Aquesta operació no modifica el valor del determinant.
- Permutar dues files entre si. Aquesta operació canvia el signe del determinant.

Fixeu-vos que MEG ens permet calcular el determinant de la matriu A si multipliquem el determinant de la matriu U per $(-1)^c$, amb c el nombre de canvis de files.

Example 2.2.1

Per resoldre el sistema lineal $\begin{array}{rcl} 2x_1 + 4x_2 + x_3 & = & 1\,, \\ 8x_1 - x_2 + 3x_3 & = & 0\,, \\ 2x_1 + 5x_2 & = & 0\,. \end{array}$ traballem amb la matriu ampliada:

$$G^{(1)} = \begin{bmatrix} 2 & 4 & 1 & 1 \\ 8 & -1 & 3 & 0 \\ 2 & 5 & 1 & 0 \end{bmatrix} \quad \rightsquigarrow \quad G^{(2)} = \begin{bmatrix} 2 & 4 & 1 & 1 \\ 0 & -17 & -1 & -4 \\ 0 & 1 & -1 & -1 \end{bmatrix}$$

$$G^{(3)} = \begin{bmatrix} 2 & 4 & 1 & 1 \\ 0 & -17 & -1 & -4 \\ 0 & 0 & -\frac{18}{17} & -\frac{21}{17} \end{bmatrix} \quad \rightsquigarrow \quad x = \begin{bmatrix} -5/12 \\ 1/6 \\ 7/6 \end{bmatrix} \text{ i } det(A) = 36$$

Algorisme Descrivim el mètode amb més detall. La matriu del sistema A és redueix a triangular superior en $n-1$ passos si A té n files. Introduïm la notació $G^{(k)}$ la matriu transformada en el pas k, a l'inici començem amb la matriu ampliada $G^{(1)} = [A\,, b]$.

[3] Els dos sistemes tenen una única solució i coincideix

Primer pas Començem amb la matriu ampliada $G^{(1)} = [A\,,b]$, el primer pas és

1. verifiquem si $g_{11}^{(1)} \neq 0$, (pivot).
2. s'escull la fila 1, (fila pivot).
3. per cada fila per sota de la fila pivot calculem $m_{i1} = g_{i1}^{(1)}/g_{11}^{(1)}$, (multiplicador).
4. per cada fila per sota de la fila pivot restem m_{i1} vegades la fila pivot de la fila i.

El resultat del primer pas és una matriu, $G^{(1)}$, amb la primera columna tot zero, llevat de $g_{11}^{(1)}$. El primer pas falla si $g_{11}^{(1)} = 0$. El nombre de divisions és $n-1$, i per cada fila són n productes i n sumes; en total $n(n-1)$ per cada operació aritmètica.

Segon pas El segon pas és, començem amb la matriu $G^{(2)}$ i

1. verifiquem si $g_{22}^{(2)} \neq 0$, (pivot).
2. s'escull fila pivot, la fila 2 de la matriu $G^{(2)}$.
3. per cada fila per sota de la fila pivot calculo $m_{i2} = g_{i2}^{(2)}/g_{12}^{(2)}$, (multiplicador).
4. per cada fila per sota de la fila pivot restem m_{i2} vegades la fila pivot de la fila i. Els zeros de la primera columna es mantenen.

El resultat és una matriu, $G^{(3)}$, amb $g_{m2}^{(3)} = 0$ per $2 < m \leq n$. El segon pas falla si $g_{22}^{(2)} = 0$. El nombre de divisions és $n-2$, i per cada fila són $n-1$ productes i$n-1$ sumes; en total $(n-1)(n-2)$ per cada operació aritmètica.

Pas k, $\mathbf{1 < k < n}$ En general, en el pas $1 < k < n$, reduim la columna k de la matriu $G^{(k)}$, modificant des de la fila k fins a la n amb la fórmula

$$m_{i\,k} = \frac{g_{i\,k}^{(k)}}{g_{k\,k}^{(k)}}\,, \quad i = k+1 \ldots n\,,$$

$$NovaFila(i) = Fila(i) - m_{ik} \cdot Fila(k)\,, i = k+1 \cdots n\,.$$

sempre i quan $g_{kk}^{(k)} \neq 0$. El resultat és una matriu, $G^{(k+1)}$, amb $g_{mk}^{(k+1)} = 0$ per $m > k$. El pas falla si $g_{kk}^{(k)} = 0$. El nombre de divisions és $n-k$, i per cada fila són $n-k$ productes i sumes; en total $(n-k)(n-k+1)$ per cada operació aritmètica.

Pas final El procès finalitza en $n-1$ pasos, ja que la matriu $G^{(n)}$ resultant del pas $n-1$ tindrà forma triangular. Ara resta resoldre el sistema triangular resultant fent ús de l'agoritme de substitució enrera 2.2.1

	$+/-$	$*$	$/$	parcial	total
Resoldre	$\frac{n^2-n}{2}$	$\frac{n^2-n}{2}$	n	n^2	$\frac{2n^3}{3}+\mathcal{O}(n^2)$
Reduir matriu	$\frac{n^3-n}{3}$	$\frac{n^3-n}{3}$	$\frac{n^2-n}{2}$	$\frac{4n^3+3n^2-7n}{6}$	

Taula 2.2: *Nombre d'operacions resoldre* $\mathbf{A}\,\mathbf{x}=\mathbf{b}$ *transformant la matriu de coeficients a triangular. Observar la reducció del cost respecte el mètode de Cramer.*

Eficiència Per avaluar l'eficiència de l'algorisme MEG, recomptem el nombre d'operacions aritmètiques que cal realitzar per obtenir la solució del sistema lineal $\mathbf{Ax}=\mathbf{b}$.

Teorema 2.2.5 (Cost computacional per a matrius plenes)
El nombre total d'operacions aritmètiques de l'algoritme MEG per transformar la matriu A en triangular una matriu triangular superior U són $\dfrac{2n^3}{3}+\mathcal{O}(n^2)$.

Demostració 2.2.2 *Acumulem les operacions de cada pas.*

El nombre total de divisions per triangular matriu és:

$$\sum_{k=1}^{n-1}(n-k)=\frac{1}{2}n(n-1)=\frac{n^2-n}{2}\ \triangleright\ \frac{n^2}{2}+\mathcal{O}(n)\,.$$

El nombre total de multiplicacions/sumes per triangular matriu és[4]*:*

$$\sum_{k=1}^{n-1}(n-k+1)(n-k)=\sum_{k=1}^{n-1}(n^2+n-k(2n+1)+k^2)=$$

$$=(n^2+n)(n-1)-(2n+1)\sum_{k=1}^{n-1}k+\sum_{k=1}^{n-1}k^2=$$

$$=(n^2+n)(n-1)-(2n+1)\frac{n(n-1)}{2}+\frac{n(n-1)(2n-1)}{6}=$$

$$=\frac{n(n-1)(n+1)}{3}=\frac{n^3-n}{3}\ \triangleright\ \frac{n^3}{3}+\mathcal{O}(n^2)\,.$$

Si acumulem les sumes, els productes i les divisions obtenim un total de $\dfrac{2n^3}{3}+\mathcal{O}(n^2)$ *operacions aritmètiques.*

Observeu la taula següent: temps emprat en resoldre un sistema lineal d'ordre n en un ordinador depenen del nombre de flops de la CPU de l'ordinador emprat per resoldre'l:

[4]Recordar $\displaystyle\sum_{i=1}^{m}i^2=\frac{m(m+1)(2m+1)}{6}$.

	flops		
n	10^9 (Giga)	10^{12} (Tera)	10^{15} (Peta)
10^2	$7 \cdot 10^{-4}$ s	despreciable	despreciable
10^4	11 m	0.7 s	$7 \cdot 10^{-4}$ s
10^6	21 anys	7.7 mes	11 m
10^8	fora de l'abast	fora de l'abast	21 anys

Estratègies de pivotar Què passa si el pivot del pas k de l'algorisme MEG és zero? La solució més emprada és l'estratègia **pivot parcial**. Es busca la primera fila per sota de la fila k que tingui valor no nul, i s'intercanvien les dues files.

Example 2.2.2

$$\left[\begin{array}{cc|c} 0 & 1 & 1 \\ 1 & 0 & 0 \end{array}\right] \qquad \begin{aligned} x_1 + 3x_2 + \ x_3 &= -3\,, \\ 3x_1 + 9x_2 + 4x_3 &= -7\,, \\ 2x_1 - \ x_2 + \ x_3 &= 6\,. \end{aligned}$$

Errors d'arrodoniment L'algorisme MEG pot presentar problemes quan treballem amb aritmètica de punt flotant de precisió limitada. Què passa si el pivot del pas k és proper a zero? Un exemple:

Example 2.2.3

$$\left[\begin{array}{cc|c} \epsilon & 1 & 1 \\ 1 & 1 & 2 \end{array}\right] \rightsquigarrow \left[\begin{array}{cc|c} \epsilon & 1 & 1 \\ 0 & 1-\epsilon^{-1} & 2-\epsilon^{-1} \end{array}\right] \rightsquigarrow \begin{aligned} x_2 &= (2-\epsilon^{-1})/(1-\epsilon^{-1}) \rhd 1 \\ x_1 &= (1-x_2)\epsilon^{-1} \rhd 0 \end{aligned}$$

En un ordinador, si ϵ és prou petit, els termes $2-\epsilon^{-1}$ i $1-\epsilon^{-1}$ es computaran com un mateix número; la solució que s'obté del sistema és $(0,1)^t$. Però la solució correcte és $\hat{x} = (1,1)^t$. La solució calculada per x_2 és exacte però totalment inexacte per x_1. Si permutem les dues equacions, obtenim per substitució enrera la solució $\hat{x} = (1,1)^t$.

$$\left[\begin{array}{cc|c} \epsilon & 1 & 1 \\ 1 & 1 & 2 \end{array}\right] \rightsquigarrow \left[\begin{array}{cc|c} 1 & 1 & 2 \\ \epsilon & 1 & 1 \end{array}\right] \rightsquigarrow \left[\begin{array}{cc|c} 1 & 1 & 2 \\ 0 & 1-\epsilon & 1-2\epsilon \end{array}\right] \rightsquigarrow \begin{aligned} x_2 &= (1-2\epsilon)/(1-\epsilon) \approx 1 \\ x_1 &= 2 - x_2 \approx 1 \end{aligned}$$

El problema anterior no rau en la petitesa del terme a_{ii}, sinó en la petitesa relativa respecte dels altres elements de la seva fila. La conclusió que podem extreure és que un bon algorisme ha d'incloure l'intercanvi d'equacions quan les circumstàncies ho exigeixin.

Principalment s'apliquen dues tècniques

Pivot parcial. S'agafa com a pivot l'element de més gran magnitud de tota la columna k.

Pivot parcial escalat. S'agafa com a pivot l'element de la columna k o per sota de la diagonal principal que té la grandària relativa més gran respecte dels altres elements de la fila.

Consulteu Càlcul numèric: teoria i pràctica pàgina 120 i 121.

Estabilitat. El métode de Gauss és numèricament estable i no cal fer intercanvis de files i columnes si la matriu A és diagonal dominant[5] o si la matriu A és simètrica[6] i definida positiva[7].

Mètode de Gauss-Jordan El mètode de Gauss transforma la matriu de coeficients en una matriu triangular superior. El mètode de Gauss-Jordan continua el procés de transformació fins a obtenir una matriu diagonal ($a_{ij} = 0$ per a qualsevol $i \neq j$).

$$\left[\begin{array}{cccc|c} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ a_{31} & a_{32} & \dots & a_{3n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} & b_n \end{array}\right] \Longrightarrow \left[\begin{array}{ccccc|c} a_{11} & 0 & 0 & \dots & 0 & \bar{\bar{b}}_1 \\ 0 & a_{22} & 0 & \dots & 0 & \bar{\bar{b}}_2 \\ 0 & 0 & a_{33} & \dots & 0 & \bar{\bar{b}}_3 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \\ 0 & 0 & \dots & 0 & a_{nn} & \bar{\bar{b}}_n \end{array}\right]$$

Algorisme (MEGJ) En general, en el pas $k < n$, reduim la columna k de la matriu $G^{(k)}$, modificant totes les files, llevat de la fila k, amb la fórmula

$$m_{i\,k} = \frac{a_{i\,k}^{(k-1)}}{a_{k\,k}^{(k-1)}}\,, \qquad i \neq k,$$

$$NovaFila(i) = Fila(i) - m_{ik} \cdot Fila(k)\,,\ i \neq k\,.$$

Eficiència. El sistema diagonal és més fàcil de resoldre, però la reducció a sistema diagonal és més costosa però s'usa per calcular la matriu inversa.

[5] $|a_{ii}| \geq \sum_{\substack{i\neq j \\ 1\leq j\leq n}} |a_{ij}|\,,\ 1\leq i\leq n$

[6] $A^t = A$

[7] $x^t A x > 0\,, \forall x \neq 0$

2.2.4 Mètode Compactes

El tret principal d'aquests métodes es treballar sols amb la matriu A i presentar-la com $A = BC$ on B i C són matrius més econòmiques d'invertir (nombre operacions).
Les descomposicions més conegudes per a matrius $n \times n$ (quadrades) són

$A = LU$, L triangular inferior i U triangular superior.

$A = R^t R$, R triangular superior i A simètrica definida positiva.

$A = QR$, Q ortogonal i R triangular superior. Aquest algorisme també es pot aplicar a matrius $n \times m$.

Factorització LU La descomposició LU (factorització) d'una matriu quadrada no singular(quadrada) A significa expressar la matriu com la multiplicació d'una matriu triangular inferior L i una matriu triangular superior U. Per al cas en què es necessiti alguna operació de canvi de fila com en l'eliminació de Gauss, incloem una matriu de permutació[8] P que representa les operacions de canvi de fila necessàries, així s'escriu la descomposició LU com

$$PA = LU \tag{2.2.3}$$

Per a què serveix la descomposició LU?
• Es pot utilitzar per resoldre un sistema d'equacions lineals com

$$A\mathbf{x} = b \tag{2.2.4}$$

Un cop tenim la descomposició LU, de la matriu de coeficients A s'expressa com $A = P^t LU$, en aquest punt recordem que és més eficient utilitzar les matrius triangulars inferior/superior per resoldre l'equació 2.2.4. El procediment és el següent:

$$A\mathbf{x} = b \to P^t LU\mathbf{x} = b \to LU\mathbf{x} = Pb \to U\mathbf{x} = L^{-1}(Pb) \to \mathbf{x} = U^{-1}\left(L^{-1}(Pb)\right) \tag{2.2.5}$$

Tingueu en compte que la premultiplicació de L^{-1} i U^{-1} per un vector es pot realitzar mitjançant els algorismes de la substitució cap endavant i cap enrere, respectivament.

• Es pot utilitzar si necessitem resoldre contínuament el sistema d'equacions lineals amb la mateixa matriu de coeficients A per a diferents vectors b, és una opció raonable en termes de temps de càlcul i precisió gauardar la descomposició LU de la matriu de coeficients A i aplicar el procés de substitució cap endavant/enrere. El cas d'obtenir la matriu inversa d'una matriu A.

[8]Es verifica $P^t P = I_n,\ PP^t = I_n$

Algorisme LU Plantejem el problema com un problema de coeficients indeterminats, $A = L\,U$, si escrivim els elements obtenim l'algorisme per calcular els elements de les matrius L i U.

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & \vdots & & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{pmatrix} = \begin{pmatrix} l_{11} & 0 & 0 & \dots & 0 \\ l_{21} & l_{22} & 0 & \dots & 0 \\ l_{31} & l_{32} & l_{33} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \dots & l_{nn} \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} & \dots & u_{1n} \\ 0 & u_{22} & u_{23} & \dots & u_{2n} \\ 0 & 0 & u_{33} & \dots & u_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & u_{nn} \end{pmatrix}$$

Per $k = 1, \dots, n$, especificar un valor be per a l_{kk} o be per a u_{kk} i calcular la resta de valors mitjançant:

$$\begin{aligned} l_{kk}u_{kk} &= a_{kk} - \sum_{r=1}^{k-1} l_{kr}\,u_{rk}\,, \\ l_{ik} &= \frac{1}{u_{kk}}\left(a_{ik} - \sum_{r=1}^{k-1} l_{ir}\,u_{rk}\right)\,, \quad i = k+1, \dots, n \\ u_{kj} &= \frac{1}{l_{kk}}\left(a_{kj} - \sum_{r=1}^{k-1} l_{kr}\,u_{rj}\right)\,, \quad j = k+1, \dots, n \end{aligned} \tag{2.2.6}$$

Les equacions (2.2.6) són un sistema de n^2 equacions i $n^2 + n$ incògnites, si es demanana que $l_{ii} = 1$ (métode de Doolitle) o que $u_{ii} = 1$ (métode de Crout) obtenim un sistema de n^2 equacions i n^2 incògnites. El cost de la factorització és $\dfrac{4n^3 + 2n}{6}$.

Si la solució del sistema (2.2.6) amb la condició $l_{ii} = 1$ s'obté sense pivotar, resulta que U és la matriu resultant de l'algorisme MEG aplicat a la matriu A, $U = G^{(n)}$ i L és la matriu dels multiplicadors de l'algorisme MEG, $l_{ik} = m_{ik}$ per $k < i$, $k, i = 2, \dots, n-1$.

Teorema 2.2.6 (Existència factorització)
Una matriu A, regular, admet factorització LU si i només si totes les matrius A_i, $i = 1, \dots, n$, són regulars.

Les matrius A_i són les submatrius de la matriu A formades per les i primeres files i les i columnes de la matriu A, per $i = 1, \dots, n$

Teorema 2.2.7 (Existència factorització)
Si la matriu A és diagonal dominant o si la matriu A és simètrica i definida positiva, admet factorització LU sense permuntacions de files i columnes.

Consulteu Càlcul numèric: teoria i pràctica des de la pàgina 121 fins la pàgina 125.
MATLAB® , consulteu LU matrix factorization

Mètode de Txoleski Només pot emprar-se aquest mètode quan la matriu A és simètrica i definida positiva[9]. Serveix com a test per dir si una matriu és definida positiva.

Teorema 2.2.8 (Existència) *Tota matriu A simètrica i definida positiva es pot factoritzar com $A = R^tR$ per R triangular superior o $A = \mathcal{L}\mathcal{L}^t$ per $\mathcal{L}$ triangular inferior (factorització de Txoleski).*

Demostració 2.2.3 *En la descomposició LU (Doolittle) d'una matriu definida positiva observem que $u_{11} = a_{11} > 0$ i $u_{kk} = \dfrac{detA_k}{detA_k - 1} > 0$ per $k = 2 \div n$, llavors la descomposició s'expressa com*

$$A = LU = LDD^{-1}U = L'D^{-1}U \qquad D = diag(u_{11}, \ldots, u_{nn})$$

i del fet que A és simètrica

$$A = A^t = U^tD^{-1}\left(L'\right)^t \to U^t = L' = LD \Longrightarrow A = LDL^t.$$

Notem per $D^{1/2} = diag(\sqrt{u_{11}}, \ldots, \sqrt{u_{nn}})$ i per per $\mathcal{L}$ la matriu $LD^{1/2}$, obtenim $A = \mathcal{L}\mathcal{L}^t$

Algorisme Plantejem el problema com un problema de coeficients indeterminats, $A = L\,D\,L^t$, si escrivim els elements obtenim l'algorisme per calcular els elements de les matrius L i D.

Per $k = 1, \ldots, n$, calcular els valors mitjançant:

$$\begin{aligned} d_{kk} &= a_{kk} - \sum_{r=1}^{k-1} l_{kr}^2\, d_{rr}\,, \\ l_{ik} &= \frac{1}{d_{kk}}\left(a_{ik} - \sum_{r=1}^{k-1} l_{ir}\, d_{rr}\, l_{rk}\right), \quad i = k+1, \ldots, n \end{aligned} \tag{2.2.7}$$

El nombre d'operacions es redueix quasi a la meitat, el cost de la factorització és $\mathcal{O}(n^3)$.
Consulteu Càlcul numèric: teoria i pràctica pàgina 125.
MATLAB® , consulteu Cholesky factorization

Factorització QR La factorització QR expressa la matriu A com el producte de dues matrius, una ortogonal ($Q^tQ = I$) i l'altre triangular superior (R). Així, el sistema lineal $Ax = b$ es reduiex a resoldre $Rx = Q^tb$, uns sistema triangular

Aquesta factorització és més costosa que la **LU** però les matrius A i $R = Q^tA$ tenen el mateix nombre de condició. La factorització **QR** no és única. En MATLAB® la factorització QR s'obté per [Q, R, P]=qr(A). Consulteu QR factorization
Consulteu Càlcul numèric: teoria i pràctica Capítol 7, en particular des la pàgina 274 fins la pàgina 288.

[9]En aquest cas la propietat és preservada per a la matriu inversa, A^{-1} , i per a qualsevol submatriu principal; a més, recordem que tots els determinants de les matrius esmentades són positius

2.3 Fites d'error

Una de les fonts d'errors més freqüent en els mètodes numèrics en resoldre sistemes lineals prové de la propagació dels errors en les dades on juga un paper important la sensibilitat de la matriu del sistema (nombre de condició). L'altra font d'error depèn de l'acumulació d'errors d'arrodoniment en el mètode emprat i farem una anàlisi per al mètode d'eliminació gaussiana.

En tot moment farem servir normes matricials consistents en les normes vectorials (vegeu apèndix A).

2.3.1 Nombre de condició

Un sistema d'equacions lineals $Ax = b$ es diu *ben condicionat* quan els errors de la matriu de coeficients A i del vector terme independent b produeixen en la solució un error del mateix ordre.

$$\begin{array}{c} \|A - \bar{A}\| < \epsilon \\ \\ \|b - \bar{b}\| < \epsilon \end{array} \quad \Longrightarrow \left\{ \begin{array}{ll} \|x - \bar{x}\| \simeq \epsilon\,, & \text{ben condicionat,} \\ \\ \|x - \bar{x}\| \gg \epsilon\,, & \text{mal condicionat,} \end{array} \right.$$

Un sistema d'equacions lineals $Ax = b$ es diu *mal condicionat* quan els errors de la matriu de coeficients A i del vector terme independent b produeixen en la solució del sistema un error d'ordre superior a l'error de les dades.

Definició 4 (Nombre de condició) *Sigui A una matriu, i $\|\cdot\|$ qualsevol norma multiplicativa, es defineix el nombre de condició de la matriu A per*

$$\mathcal{K}(A) = \left\{ \begin{array}{ll} \|A\|\|A^{-1}\|\,, & det(A) \neq 0 \\ \infty\,, & altrament \end{array} \right.$$

Propietats

- $\mathcal{K}(A) \geq 1, \quad \mathcal{K}(I) = 1.$
- Si $B = zA$, per $z \neq 0$ real, llavors $\mathcal{K}(B) = \mathcal{K}(A)$.
- $\mathcal{K}(AB) \leq \mathcal{K}(A)\,\mathcal{K}(B)$.
- $\mathcal{K}_2(A) = \sigma_n/\sigma_1$, amb σ_1 i σ_n valors singulars.
- $\mathcal{K}_2(A) = \mathcal{K}_2(AQ) = \mathcal{K}_2(QA)$ per Q matriu ortogonal (o unitària).

Fites de l'error Sigui el sistema sistema lineal $Ax = b$, estudiem com canvia la solució $x^* = x + \delta x$ si considerem que actua una pertorbació δb en el vector terme independent b, si considerem que actua una pertorbació δA en la matriu del sistema A.

Teorema 2.3.1 (Fites de l'error)

$$\text{Errors en } b \quad \frac{\|\delta x\|}{\|x\|} \le \mathcal{K}(A)\frac{\|\delta b\|}{\|b\|} \tag{2.3.1}$$

$$\text{Errors en } A \quad \frac{\|\delta x\|}{\|x + \delta x\|} \le \mathcal{K}(A)\frac{\|\delta A\|}{\|A\|} \tag{2.3.2}$$

$$\text{Errors en } A \text{ i } b \quad \frac{\|\delta x\|}{\|x + \delta x\|} \le \frac{\mathcal{K}(A)}{1 - \mathcal{K}(A)\frac{\|\delta A\|}{\|A\|}}\left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|b\|}\right) \tag{2.3.3}$$

Demostració 2.3.1 *Consulteu Càlcul numèric: teoria i pràctica Capítol 4, en particular des la pàgines* $127 - 128 - 129$.

Vector residu Com a criteri de comparació entre la solució exacta x, i la solució calculada $x^* = x + \delta x$, del sistema lineal $Ax = b$ definim el vector residu $r(x^*)$ per:

$$r(x^*) = A\,\delta x = Ax - Ax^* = b - Ax^*\,.$$

Llavors l'expressió (2.3.1) s'escriu

$$\frac{\|x^* - x\|}{\|x\|} = \frac{\|\delta x\|}{\|x\|} \le \mathcal{K}(A)\frac{\|r(x^*)\|}{\|b\|} \tag{2.3.4}$$

que ens diu que no podem quedar-nos amb l'avaluació del vector residu, ja que l'error vé afectat pel nombre de condició.

2.4 Mètodes iteratius

Són mètodes que construeixen una successió de vectors convergent a la solució exacte amb un nombre finit d'operacions en cada iteració, si no fos pels errors d'arrodoniment acumulats i les possibles imprecisions en el coneixement inicial de la matriu A i el vector b. Pel seu baix cost computacional es consideren adients per a sistemes lineals d'ordre alt.

En aquest curs presentem tres métodes iteratius estacionaris, el mètode de Jacobi, el mètode de Gauss-Seidel i els mètodes de SOR.

2.4.1 Métodes iteratius estacionaris

En general, en els métodes iteratius per resoldre el sistema lineal d'ordre n, $\mathbf{Ax} = \mathbf{b}$, s'empra una matriu $\mathbf{B}$ escollida de tal manera que el problema original tingui la forma $\mathbf{x} = \mathbf{Bx} + \mathbf{c}$, i la solució $\mathbf{x}^*$ del problema original sigui la solució de la formulació equivalent. En aquest cas es diu que els dos sistemes són equivalents (consistents).

$$\left.\begin{array}{r} Ax = b \\ Ax^* = b \end{array}\right\} \Longleftrightarrow \left\{\begin{array}{r} x = B\,x + c \\ x^* = B\,x^* + c \end{array}\right.$$

L'equació $x = Bx + c$ suggereix un procés iteratiu que es concreta en escriure:

$$x^{(k+1)} = B\,x^{(k)} + c, \quad k \geq 0, \text{ per un } x^{(0)} \text{ donat.} \tag{2.4.1}$$

El vector inicial $x^{(0)}$ arbitrari, ara bé si es té un bon candidat, aquest és el que cal emprar.

Convergència El nostre objectiu és escollir una matriu B de manera que la successió

$$x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(k)}, \ldots$$

resultant de 2.4.1 convergeixi a la solució x^* del sistema d'equacions lineal d'ordre n, $Ax = b$.

Teorema 2.4.1 (*a posteriori*) *Donat un sistema lineal d'ordre n, $Ax = b$, construïm el sistema equivalent $x = Bx + c$, que es resol de forma iterativa.*

Definim el vector residu per $r^{(k)} = b - Ax^{(k)}$ per tot $x^{(k)}$ resultant de 2.4.1, llavors

$$\lim_{k\to\infty} x^{(k)} = x^* \Longleftrightarrow \lim_{k\to\infty} r^{(k)} = 0\,.$$

Demostració 2.4.1 *Observem que*

$$r^{(k)} = b - Ax^{(k)} = Ax^* - Ax^{(k)} = A\left(x^* - x^{(k)}\right)\,.$$

Teorema 2.4.2 (*a priori*) *Donat un sistema lineal d'ordre n, $Ax = b$, construïm el sistema equivalent $x = Bx + c$. Si $\|B\| < 1$ per alguna norma matricial compatible (vegeu A.1.1), llavors la successió generada pel mètode iteratiu $x^{(k+1)} = B\,x^{(k)} + c$ és convergent a la solució x^* del sistema lineal $Ax^* = b$ per qualsevol vector inicial $x^{(0)}$.*

Demostració 2.4.2 *A partir de $x^{(k+1)} = B\,x^{(k)} + c$ i de $x^* = B\,x^* + c$ es dedueix que*

$$x^{(k)} - x^* = B\left(x^{(k-1)} - x^*\right) = \cdots = B^k\left(x^{(0)} - x^*\right) . \tag{2.4.2}$$

Es pren la norma vectorial i la norma matricial compatible amb la norma vectorial en què $\|B\| < 1$, s'obté

$$||x^{(k)} - x^*|| = ||B^k|| \cdot ||x^{(0)} - x^*|| \leq ||B||^k \cdot ||x^{(0)} - x^*|| . \tag{2.4.3}$$

Si $\|B\| < 1$ es pot concloure que $\lim\limits_{k\to\infty}\left(x^{(k)} - x^*\right) = 0$ per qualsevol vector inicial $x^{(0)}$.

Procediment computacional Donat un sistema lineal d'ordre n, $Ax = b$, construïm el sistema equivalent $x = Bx + c$, si $\|B\| < 1$, es pren $x^{(0)}$ arbitrari, i es genera la successió de vectors $x^{(1)}$, $x^{(2)}$, $x^{(3)}$, ... a partir de la recurrència $x^{(k+1)} = B\,x^{(k)} + c$.

Com en tot mètode iteratiu, cal especificar un criteri de parada fent ús dels teoremes de convergència i un nombre màxim d'iteracions M, per assegurar que el procés s'atura si no s'assoleix la convergència.

Criteris de parada Si $\beta = ||B|| < 1$, es verifica que:

$$x^{(k)} - x^* = B\left(x^{(k)} - x^*\right) - B\left(x^{(k)} - x^{(k-1)}\right) \quad \Longrightarrow \tag{2.4.4}$$

$$||x^{(k)} - x^*|| \leq \frac{\beta}{1-\beta} \cdot ||x^{(k)} - x^{(k-1)}|| . \tag{2.4.5}$$

La darrera expressió ens informa que és segur aturar el procès computacional si $||x^{(k)} - x^{(k-1)}|| < \epsilon$ i en aquest cas l'error absolut es fita per $||x^{(k)} - x^*|| \leq \dfrac{\beta\epsilon}{1-\beta}$.

En l'equació 2.4.4 afegim que $B\left(x^{(k)} - x^{(k-1)}\right) = B^k\left(x^{(1)} - x^{(0)}\right)$, obtenim

$$||x^{(k)} - x^*|| \leq \frac{\beta^k}{1-\beta} \cdot ||x^{(1)} - x^{(0)}|| . \tag{2.4.6}$$

La darrera expressió ens permet determinar el nombre d'iteracions del mètode iteratiu 2.4.1 per obtenir la solució aproximada pel procès computacional amb un error absolut prefixat.

Cost computacional Si la matriu A és d'ordre $n \times n$, cada iteració de 2.4.1 són n^2 sumes i n^2 productes; després de k iteracions són $2n^2k$ operacions a realitzar per determinar una aproximació x^k de la solució x^*.

Si reduïm el nombre d'iteracions del mètode, reduïem el cost computacional; en l'expressió (2.4.6) el nombre d'iteracions depèn del valor de $\beta = ||B|| < 1$.

Definició 5 (Velocitat de convergència) $R = -\log(\rho(B))$

2.4.2 Mètode de Jacobi

El mètode Jacobi es basa en la resolució de cada variable localment respecte a les altres variables: de la primera equació resolem x_1, de la segona equació resolem x_2, ... Suposem $a_{ii} \neq 0 \ \forall i = 1 \div n$, si cal es permuten les equacions.

$$\left.\begin{array}{r} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \quad \vdots \qquad\qquad \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{array}\right\} \iff \left\{\begin{array}{l} x_i^{(k+1)} = \dfrac{1}{a_{ii}}\left(b_i - \displaystyle\sum_{\substack{j=1\\ j\neq i}}^{n} a_{ij}x_j^{(k)}\right) \\ i = 1, 2, \ldots, n, \quad k \geq 0\,. \end{array}\right.$$

Notació matricial Per convertir $Ax = b$ en un sistema de la forma $x = Bx + c$, expressem la matriu A com a suma de tres matrius: $A = L_A + D_A + U_A$, la diagonal, la part trinagular estrictament superior i la part triangular estrictament inferior:

$$\underbrace{\begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{pmatrix}}_{A} = \underbrace{\begin{pmatrix} 0 & 0 & \ldots & 0 \\ a_{21} & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ a_{n1} & \ldots & a_{nn-1} & 0 \end{pmatrix}}_{L_A} + \underbrace{\begin{pmatrix} a_{11} & 0 & \ldots & 0 \\ 0 & a_{22} & 0\ldots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & a_{nn} \end{pmatrix}}_{D_A} + \underbrace{\begin{pmatrix} 0 & a_{12} & \ldots & a_{1n} \\ 0 & 0 & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & a_{n-1n} \\ 0 & 0 & \ldots & 0 \end{pmatrix}}_{U_A}.$$

El sistema $Ax = b$ és equivalent a $(L_a + D_A + U_A)\,x = b$, resolent la variable x_i de la fila i queda $D_Ax = b - (L_a + U_A)\,x$ equivalent a $x = (D_A)^{-1}b - (D_A)^{-1}(L_a + U_A)\,x$, que permet definir el mètode de Jacobi en forma matricial com:

$$x^{(k+1)} = (D_A)^{-1}b - (D_A)^{-1}(L_A + U_A)\,x^{(k)}\,, \quad k \geq 0\,. \tag{2.4.7}$$

Teorema 2.4.3 *Si A és diagonal dominant estricte, la successió $x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(k)}, \ldots$ resultant de 2.4.7 convergeix per qualsevol vector inicial $x^{(0)}$.*

Demostració 2.4.3 *Si A és diagonal dominant esctrite vol dir que $|a_{ii}| > \displaystyle\sum_{\substack{1\leq j\leq n\\ i\neq j}} |a_{ij}|$ i en aquest cas $\|B_J\|_\infty = \displaystyle\max_{1\leq i\leq n} \sum_{\substack{j=1\\ j\neq i}}^{n} \left|\frac{a_{ij}}{a_{ii}}\right| < 1$.*

2.4.3 Mètode de Gauss-Seidel

El mètode Gauss-Seidel és com el mètode Jacobi, excepte que utilitza valors actualitzats tan aviat com estiguin disponibles. Suposem $a_{ii} \neq 0 \ \forall i = 1 \div n$, si cal es permuten les equacions.

$$\left.\begin{array}{r} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \quad\ \vdots \qquad\qquad \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{array}\right\} \Longleftrightarrow \left\{\begin{array}{l} x_i^{(k+1)} = \dfrac{1}{a_{ii}}\left(b_i - \displaystyle\sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right) \\ \\ i = 1, 2, \ldots, n, \quad k \geq 0\,. \end{array}\right.$$

Notació matricial El sistema $Ax = b$ és equivalent a $(L_a + D_A + U_A)\,x = b$, resolent la variable x_i de la fila i i incorporant els valors de les incoògnites ja calculades, queda $(L_a + D_A)\,x = b - U_A x$ equivalent a

$$x = (L_a + D_A)^{-1}\,b - (L_a + D_A)^{-1}\,U_A x\,,$$

que permet definir el mètode de Gauss-Seidel en forma matricial com:

$$x^{(k+1)} = (L_A + D_A)^{-1}b - (L_A + D_A)^{-1}U_A\,x^{(k)}\,, \quad k \geq 0\,. \tag{2.4.8}$$

Teorema 2.4.4 *Si A és diagonal dominant estricte o és simètrica definida positiva, la successió $x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(k)}, \ldots$ resultant de 2.4.8 convergeix per qualsevol vector inicial $x^{(0)}$.*

Demostració 2.4.4 *Consulteu Càlcul numèric: teoria i pràctica Capítol 4, en particular des la pàgines* $135 - 136$.

2.4.4 Mètode SOR (successive overrelaxation)

Les tècniques de relaxació s'empren per accelerar la convergència del mètode de Gauss-Seidel.

Suposem $a_{ii} \neq 0 \ \forall i = 1 \div n$, si cal es permuten les equacions. Si sumem i restem x_i^k en el mètode de Gauss-Seidel, tenim:

$$x_i^{(k+1)} = x_i^{(k)} + \underbrace{\frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i}^{n} a_{ij}x_j^{(k)}\right)}_{\text{correcció}}, \quad k \geq 0\,.$$

Expressió que podem intpretar com que cada iterat $x_i^{(k+1)}$ s'obté de l'anterior $x_i^{(k)}$ més una correcció. El mètode de relaxació consisteix en multiplicar la correcció per un paràmetre ω, anomenat **paràmetre de relaxació**. Per components seria:

$$x_i^{(k+1)} = x_i^{(k)} + \frac{\omega}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i}^{n} a_{ij}x_j^{(k)}\right), \quad k \geq 0\,. \tag{2.4.9}$$

En termes matricials $x^{(k+1)} = B\,x^{(k)} + c$,, el vector c seria $c_{sor} = \omega\,(D+\omega L)^{-1}\,b$, i la matriu B seria $B_{sor} = (D+\omega L)^{-1}\,((1-\omega)D - \omega U)$, per $\omega = 1$ tenim el mètode de Gauss-Seidel.

Lamentablement, no hi ha cap regla general per seleccionar el valor òptim del factor de relaxació ω. El teorema de Kahan ens servirà per tenir una idea dels valors que pot pendre el paràmetre ω:

Teorema 2.4.5 *Si A té tots els elements diagonals no nuls, $a_{ii} \neq 0\,, i = 1 \div n$ llavors*

$$|\omega - 1| \leq \rho(B_{sor})\,. \tag{2.4.10}$$

Demostració 2.4.5 *Com que L_A és triangular inferior estricta, $|det((D+\omega L)^{-1})| = |det(D^{-1})|$, com que U_A també és triangular estricte, llavors $|det((1-\omega)D - \omega U)| = |det((1-\omega)D)| = |(1-\omega)^n|\,|\,det(D)|$ i per tant $|det(B_{sor})| = |1-\omega|^n$. Però, el producte dels valors propis també coincideix amb el valor de $det(B_{sor})$, llavors $|1-\omega|^n \leq |\rho(B_{sor})|^n$ i per tant $|1-\omega| \leq |\rho(B_{sor})|\,$.*

Conseqüència Si el mètode és convergent, $|1-\omega| \leq |\rho(B_{sor})| < 1$, resulta que $0 < \omega < 2\,$.

Pregunta Quin valor de $\omega_0 \in (0,2)$ minimitza el radi espectral $\rho(B_{sor})$? Lamentablement, no hi ha cap regla general per seleccionar el valor òptim del factor de relaxació ω, per cert tipus de matrius es pot demostrar que:

$$\omega_{\text{òptim}} = \frac{2}{1+\sqrt{1-\rho(B_J)^2}} \quad \text{i} \quad \rho(B_{sor}) = \omega_{\text{òptim}} - 1\,.$$

Consulteu Càlcul numèric: teoria i pràctica Capítol 4, pàgines $138-141$.

Example 2.4.1

Determineu les 10 primeres iteracions del mètode de Jacobi, del mètode Gauss-Seidel i del mètode de sobrerelaxació prenent $x^{(0)} = (0\,,0\,,0\,,0)^\top$ del sistema $Ax = b$ donat per

$$A = \begin{pmatrix} 10 & -1 & 2 & 0 \\ -1 & 11 & -1 & 3 \\ 2 & -1 & 10 & -1 \\ 0 & 3 & -1 & 8 \end{pmatrix} \qquad b = \begin{pmatrix} 6 \\ 25 \\ -11 \\ 15 \end{pmatrix}$$

que té per solució $x^* = (1\,,2\,,-1\,,1)^\top$.

2.4.5 Mètodes iteratius no estacionaris

Els mètodes no estacionaris difereixen dels mètodes estacionaris en què els càlculs impliquen informació que canvia en cada iteració. Normalment, les constants es calculen prenent productes interns de residus o altres vectors derivats del mètode iteratiu.

Alguns d'aquests mètodes són: Mètode del gradient conjugat (`CG`) i variants: `MINRES`, `SYMMLQ`, `CGNE`, `CGNE`, `GMRES`, `BiCG`, `QMR`, `Bi-CGSTAB`.

Consulteu Documentació de MATLAB® - Iterative Methods for Linear Systems

Vector residu i vector gradient $Ax = b\,,$ A simètrica i definida positiva.

Resoldre el sistema lineal $Ax = b$ és equivalent al problema de minimitzar la funció definida per

$$\phi(x) = \frac{1}{2}x^T Ax - x^T c \qquad (2.4.11)$$

Obs. El gradient d'aquesta funció és $\nabla\phi(x) = Ax - b$.

Consulteu (accès biblioteca UPC):

James E. Gentle Matrix Algebra Theory, Computations, and Applications in Statistics

2.5 Sistemes lineals sobredeterminats

Aplicació Suposem que volem donar un model lineal (o polinomial) per a un conjunt de dades mesurades per observació; en aquest cas realitzarem un nombre de mesures més gran que el nombre d'incògnites per reduir la influència dels errors.

Sigui A una matriu $m \times n$, $m \geq n$, b un vector de m components, x el vector de n d'incògnites. El sistema d'equacions $Ax = b$ no té solució, llavors busquem x tal que Ax sigui la millor aproximació de b. (mètode dels mínims quadrats).

2.5.1 Mètode dels mínims quadrats

Teorema 2.5.1 (Equacions normals) *Si x és la solució dels sistema d'equacions normals, $A'(b - Ax) = 0$, llavors*

$$||r_x||_2 \leq ||r_y||_2.$$

Demostració 2.5.1 *La matriu de coeficients del sistema d'equacions normals, A^tA és no singular si i només si $rank(A) = n$, i en aquest cas les equacions normals tenen solució única.*

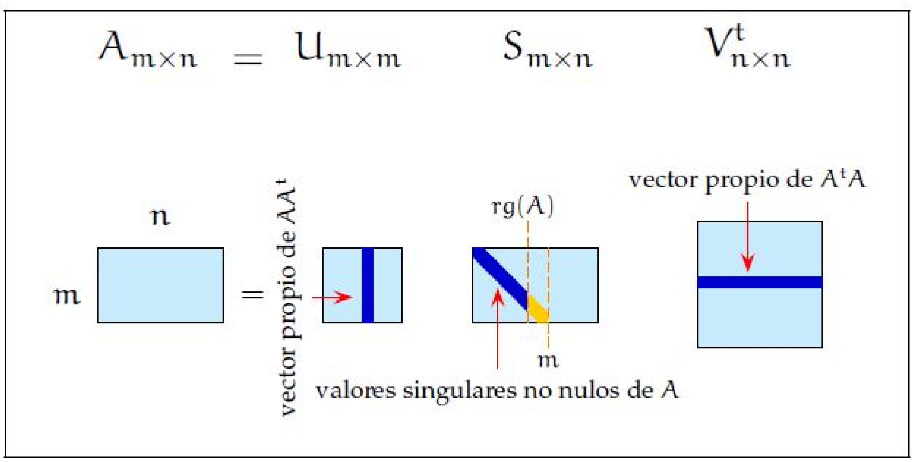

Figura 2.1: Descomposició en Valors Singulars

Descomposició en valors singulars (SVD) La descomposició en valors singulars[10] d'una matriu $\mathbf{A}$ d'ordre $\mathbf{m} \times \mathbf{n}$ consta de tres matrius $A = USV^t$, vegeu figura 3.1. La matriu S

[10] S'anomenen **valors singulars** de la matriu A les n arrels quadrades (pos.) dels valors propis no negatius de $A^t A$.

d'ordre $\mathbf{m} \times \mathbf{n}$ conté els valors singulars en la diagonal principal. Les matrius U i V són matrius ortogonals, U d'ordre $\mathbf{m} \times \mathbf{m}$ i V d'ordre $\mathbf{n} \times \mathbf{n}$ tals que:

$$A = USV^t = [U_1, U_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} [V_1, V_2]^t$$

L'algorisme de Golub i Reinsch (1970) per al càlcul de valors singulars no calcula en cap moment la matriu A^tA, sino que treballa directament sobre la matriu A. Bàsicament, consisteix en dos grans passos: primer es transforma la matriu inicial en una de més senzilla i després s'aplica una variant del mètode QR per obtenir una successió de matrius convergent a una matriu diagonal que conté els valors singulars.

Consulteu la bibliografia per a més detalls, en aquest curs farem ús de MATLAB® per obtenir les matrius de la descomposició en valors singulars: `[U,S,V] = svd(A)`.

3 Valors i vectors propis. Valors singulars

3.1 Introducció

En el camp de l'enginyeria els valors i els vectors propis tenen una rellevància destacada per analitzar models amb oscilacions i resonàncies. El seu coneixement és bàsic en:

Sistemes elèctrics de corrent alterna. Vibració natural d'estructures. Mecànica quàntica. Làsers. Resonància Magnètica Nuclear (NMR). Anàlisis de Components Principals.

Els valors propis i els vectors propis d'una matriu $\mathbf{n} \times \mathbf{n}$ per $n \geq 4$ es calculen numèricament enlloc de fer-ho analíticament. Els mètodes numèrics que s'utilitzen a la pràctica depenen del significat geomètric de valors propis i vectors propis. L'essència de tots aquests mètodes es recull en el mètode de les potències.[1]

3.2 Mètode de les potències

Teorema 3.2.1 *Sigui* $A = (a_{i\,j})_{n\times n}$ *per* $a_{i\,j} \in \mathbb{R}$ $_{1\leq i,j\leq n}$*; matriu quadrada de nombres reals d'ordre* n *tal que:*

1. $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|$
2. *Existeixen* n *vectors propis linealment independents,* $\left\{v^{(1)}, v^{(2)}, \ldots, v^{(n)}\right\}$.
3. *Per* $x^{(0)} = a_1 v^{(1)} + a_2 v^{(2)} + \cdots + a_n v^{(n)}$ *amb* $a_1 \neq 0$

Llavors el mètode iteratiu $\boldsymbol{x^{(k+1)} = A\, x^{(k-1)}}$ *és convergent i*

$$\lim_{k\to\infty} x^{(k)} = \lim_{k\to\infty} A^k\, x^{(0)} = \lim_{k\to\infty} \lambda_1^k a_1 v^{(1)}\,.$$

[1] Per repasar els principals conceptes consulteu l'àppendix B d'aquest document.

3.2.1 Algorismes

El mètode iteratiu $\boldsymbol{x^{(k)}} = \boldsymbol{A\, x^{(k-1)}}$ convergeix al vector propi $v^{(1)}$ del valor de mòdul màxim λ_1. Aquest resultat ens permet, establir un algorisme per calcular el valor propi i el vector propi de mòdul mmàxim d'una matriu real.

Algorisme, valor propi de mòdul màxim. El teorema anterior proporciona un procediment per deteminar el valor propi de mòdul màxim i el vector propi corresponent,

$$\begin{array}{lr} Escollir & x^0 = (\ , \ , \dots , \ )^t \,, \\ Calcular\ mentre\ no\ \ es\ compleixi\ criteri\ de\ parada & z^k = A x^k \,, \\ vap\ aproximat & m_{k+1} = z^k(i)\ tal\ que\ |z^k(i)| = \|z^k\|_\infty \,, \\ vep\ aproximat & x^{k+1} = \dfrac{1}{m_{k+1}}\, z^k \,, \\ criteri\ de\ parada & \|x^{k+1} - x^k\|_\infty < \epsilon \,. \end{array} \tag{3.2.1}$$

L'aproximació del valor propi és m_{k+1} i la del vector propi és x^{k+1}.

Tots els valors propis El mètode de la potència només proporciona una parella valor propi - vector propi; aquest mètode es pot modificar per donar totes les parelles valor i vector propi d'una matriu si fem ús de les propietats dels valors propis (vegeu B.1.1). Podem establir algorismes per valors propis que no són el valor propi de mòdul màxim, la combinació d'algorismes ens proporciona un procediment per obtenir tots els valors propis i tots els vectors propis d'una matriu.

- Mètode de la potència inversa per obtenir λ^{-1}, el valor propi de mòdul mínim: $A\, x^{(k)} = x^{(k-1)}$
- Mètode de la potència amb desplaçament per obtenir $\lambda - \mu$, el valor propi de més llunyà a λ: $x^{(k)} = (A - \mu\, I) x^{(k-1)}$
- Mètode de la potència inversa amb desplaçment per obtenir $(\lambda - \mu)^{-1}$, el valor propi de més proper a λ: $(A - \mu\, I)\, x^{(k)} = x^{(k-1)}$

Example 3.2.1

Calculeu, amb quatre xifres significatives, el valor propi dominant de la matriu:

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

$$x^0_{max} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad x^0_{min} = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}$$

Vegeu el document . . . de MATLAB® del campus

3.2.2 Mètode de deflacció de Wielandt

Sigui λ_1 un valor propi de la matriu A de dimensió n i $v^{(1)}$ un vector propis associat tal que la seva primera component és igual a 1. Considereu la matriu següent: $B = A - v^{(1)}\, a_1$, on a_1 denota la primera fila de la matriu A. Aquesta nova matriu, B, té tota la primera fila igual a zero, Si λ_2 i $v^{(2)}$ són, respectivament, valor i vector propis de la matriu A amb la primera component de $v^{(2)}$ igual a 1, podem escriure

$$B\left(v^{(1)} - v^{(2)}\right) = \lambda_1 v^{(1)} - \lambda_2 v^{(2)} - \lambda_1 v^{(1)} + \lambda_2 v^{(1)} = \lambda_2\left(v^{(1)} - v^{(2)}\right) \tag{3.2.2}$$

i pertant λ_2 és un valor propi de la matriu B i $v^{(2)}$ és el vector propi associat.

3.2.3 Quocient de Rayleigh

La convergència del algorisme del mètode de la potència depèn de la magnitud del quocient $\theta = |\lambda_2|/|\lambda_1|$. Si $\theta \approx 1$ la convergència del mètode s'enlenteix i calen més iteracions per a una exactitud fixada. Per a matrius simètriques, s'aconsella la successió r_k per a l'aproximació del valor propi[2]:

$$r_k = \frac{{x^{(k)}}^t \cdot A \cdot x^{(k)}}{{x^{(k)}}^t \cdot x^{(k)}}\,. \tag{3.2.3}$$

En aquest cas, la convergència de r_k cap a $|\lambda_1|$ és de l'ordre de $\left(\frac{|\lambda_2|}{|\lambda_1|}\right)^2$.
Es pot aplicar per qualsevol tipus de matriu, però no es manté l'ordre de convergència.

3.3 Mètode QR de Francis (1961)

El mètode de la potència i el mètode de la potència inversa només proporcionen una parella valor propi - vector propi; tots dos mètodes es poden modificar per donar totes les parelles valor i vector propi d'una matriu.

[2]El sistema $x \cdot \lambda = A \cdot x$ és sobredeterminat, r_k és la solució de les equacions normals.

Hi ha, però mètodes per obtenir tots els valors propis en un sol algorisme. Un d'ells és l'anomentat mètode **QR**. Aquest és la base de tot el programari modern de càlcul de valors propis, inclòs MATLAB® .

El procediment transforma, via matrius ortogonals, la matriu A en una matriu T triangular superior (recordeu *Teorema de Schur*). Si la matriu A és simètrica, la transformació obté una matriu diagonal.

Es requereix menys esforç computacional en l'algorisme de la reducció a matriu triangular si l'algorisme s'aplica a matrius de Hessenberg.

Matrius Hessenberg

Definició 6 *Una matriu Hessenberg superior (inferior) té zeros en tots els elements per sota (sobre) de la primera subdiagonal.*

Si la matriu és simètrica o hermitiana, aleshores la forma és tridiagonal.

Transformacions de Householder Per transformar una matriu A qualsevulla en una matriu Hessenberg s'apliquen transformacions ortogonals, per exemple transformacions de Householder.

Definició 7 (Matriu de Householder) *Donat un vector $v \in \mathbb{R}^n$ definim la transformació associada a v per*

$$H_v = \begin{cases} I, & v = 0, \\ I - \dfrac{2}{v^t v} v v^t, & v \neq 0. \end{cases} \tag{3.3.1}$$

La matriu H_v de la transformació és simètrica, ortogonal i verifica $H_v^2 = I$.

Les característiques més importants d'aquesta transformació són:

- Si $x \perp v \ \Rightarrow H_v x = x$.
- Si $x \parallel v \ \Rightarrow H_v x = -x \ \ H_v v = -v$.
- Si $||x|| = ||v||, \ v = x - y \ \Rightarrow H_v x = y$.

Algorisme QR per valors i vectors propis L' algorisme consisteix en una successió de transformacions ortogonals per transformar les columnes de la matriu A a una forma triangular superior (Teorema de Schur)

1. Transformem la matriu A en una matriu H Hessenberg superior fent ús de transformacions de Householder. Iniciem el mètode fent $H_1 = H$

2. Obtenim la factorització QR de la matriu H, $H_1 = Q_1\, R_1$
3. Multiquem Q i R en ordre invers per obtenir una nova H, $H_2 = R_1\, Q_1$
4. Repetim els pasos 2 i 3 fins que ...
5. La que la matriu H té forma triangular, i pertant a dir la diagonal de H convergeix als valors propis.

Teorema 3.3.1 *La transformació $A \to H$ és una transformació de similaritat via matrius ortogonals.*

Demostració 3.3.1 *El mètode es fonamenta en el fet que qualsevulla matriu quadradra admet descompossició QR: existeixen matrius* $\boldsymbol{Q}$ *i* $\boldsymbol{R}$ *tals que $A = QR$, amb Q ortogonal ($Q^{-1} = Q^t$) i R triangular superior.*

Llavors la transformació $H_{i+1} \to H_i$ és una transformació de similaritat, ja que verifica

$$H_{i+1} = R_i\, Q_i = Q_i^{-1}\, Q_i R_i\, Q_i = Q_i^t H_i Q_i$$

3.4 Valors singulars

S'anomenen **valors singulars** de la matriu A les n arrels quadrades (pos.) dels valors propis no negatius de $A^t\, A$.

Definició 8 *Si* $\mathbf{A}$ *és una matriu* $\mathbf{m} \times \mathbf{n}$*, la matriu $A^t A$ és quadrada $n \times n$ simètrica i definida positiva, els seus n valors propis, λ_i, són reals i no negatius.*

*S'anomenen **valors singulars** de la matriu A les n arrels quadrades (pos.) dels valors propis no negatius de $A^t\, A$:*

$$\sigma_i = +\sqrt{\lambda_i}\,, \quad i = 1, \ldots, n \tag{3.4.1}$$

per a λ_i valor propi de la matriu $A^t A$. Ordenats usualment, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$.

3.4.1 Descomposició en valors singulars (SVD)

La descomposició en valors singulars d'una matriu $\mathbf{A}$ d'ordre $\mathbf{m} \times \mathbf{n}$ consta de tres matrius $A = USV^t$, vegeu figura 3.1. La matriu S d'ordre $\mathbf{m} \times \mathbf{n}$ conté els valors singulars en la diagonal principal. Les matrius U i V són matrius ortogonals, U d'ordre $\mathbf{m} \times \mathbf{m}$ i V d'ordre $\mathbf{n} \times \mathbf{n}$ tals que:

$$A = USV^t = [U_1, U_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} [V_1, V_2]^t$$

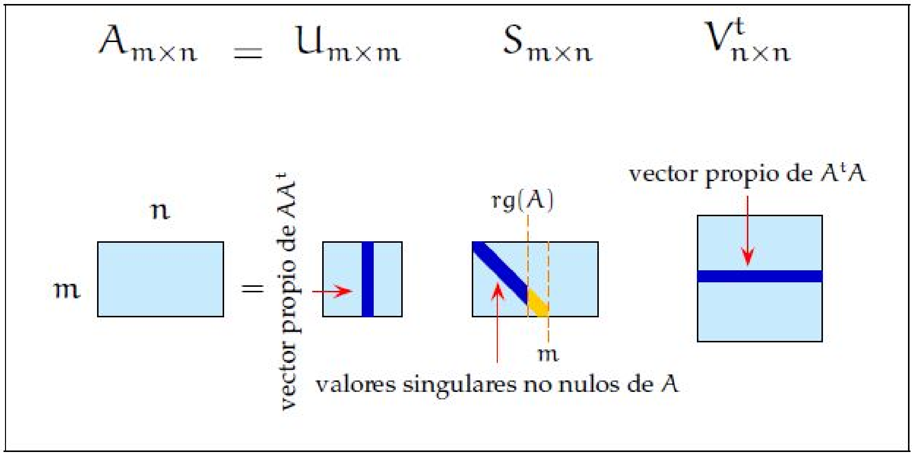

Figura 3.1: Descomposició en Valors Singulars

L'algorisme de Golub i Reinsch (1970) per al càlcul de valors singulars no calcula en cap moment la matriu A^tA, sino que treballa directament sobre la matriu A. Bàsicament, consisteix en dos grans passos: primer es transforma la matriu inicial en una de més senzilla i després s'aplica una variant del mètode QR per obtenir una successió de matrius convergent a una matriu diagonal que conté els valors singulars.

Consulteu la bibliografia per a més detalls, en aquest curs farem ús de MATLAB® per obtenir les matrius de la descomposició en valors singulars: `[U,S,V] = svd(A)`.

3.4.2 Matriu Pseudoinversa de Moore-Penrose

Sigui A una matriu $m \times n$, la pseudoinversa de Moore-Penrose, A^+ és la única matriu $n \times m$ tal que

$$A\,A^+\,A = A\,, \quad A^+\,A\,A^+ = A^+.$$

La matriu A^+ es pot calcular a partir de la descompossició en valors singulars de la matriu A.

Teorema 3.4.1 (matriu pseudoinversa: A^+)

$$A = USV^t = [U_1, U_2]\begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix}[V_1, V_2]^t \Longrightarrow A^+ = V_1\Sigma_1^{-1}U_1^t$$

La matriu Σ_1 és diagonal amb els valors singulars no nuls.

Teorema 3.4.2 (Mínims quadrats) *Sigui A una matriu $m \times n$, $m \geq n$, b un vector de m components, x el vector de n d'incògnites. Els sistema lineal $\mathbf{Ax} = \mathbf{b}$ és incompatible. La solució que minimitzi el residu ($r_y = b - Ay$) ens proporciona una solució aproximada del sistema d'equacions. La solució de residu mínim de $A\,x = b$ és $x = A^{+}b$.*

Consulteu la bibliografia per a més detalls, en aquest curs farem ús de MATLAB® per obtenir la matriu pseudoinversa: `pinv(A)`.

4 Per pràcticar

4.1 Exercicis

Nombre de condició d'una matriu.

1 Resoleu el sistema

$$\left\{\begin{array}{rcrcl} 2x & - & 4y & = & 1 \\ -2.998x & + & 6.001y & = & 2 \end{array}\right.$$

per qualsevol mètode que conegeu. Compareu la solució amb la del sistema obtingut substituin la segona equació per $-2.998x + 6y = 2$. Com són les dues solucions? És un problema estable?

2 Fent ús de les normes $||\cdot||_1$, $||\cdot||_2$, i $||\cdot||_\infty$, calculeu $||x - x^*||$ i $||Ax^* - b||$ en el cas següent:

$$(A|\,b) = \begin{pmatrix} 3.3330 & 15920 & -10.333 & | & 15913 \\ 2.2220 & 16.710 & 9.6120 & | & 28.544 \\ 1.5611 & 5.1791 & 1.6852 & | & 8.4254 \end{pmatrix} \quad x = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad x^* = \begin{pmatrix} 1.2001 \\ 0.99991 \\ 0.92538 \end{pmatrix}.$$

3 Calculeu els nombres de condició de les matrius següents fent servir les normes $||\cdot||_1$, $||\cdot||_2$, i $||\cdot||_\infty$:

$$(a)\begin{pmatrix} a+1 & a \\ a & a-1 \end{pmatrix} \qquad (b)\begin{pmatrix} 0 & 1 \\ -2 & 0 \end{pmatrix} \qquad (c)\begin{pmatrix} \alpha & 1 \\ 1 & 1 \end{pmatrix}.$$

4 Demostreu per als nombres de condició de dues matrius A i B: $\kappa(AB) \leq \kappa(A)\kappa(B)$.

5 Demostreu que el nombre de condició d'una matriu A verifica: $\kappa(\lambda A) = \kappa(A) \quad (\lambda \neq 0)$.

6 Feu una predicció de com petits canvis en el terme independent b afecten a la solució x del sistema d'equacions $Ax = b$. Poseu a prova la vostra predicció per a

$$A = \begin{pmatrix} 1 & 2 \\ 1 & 2.01 \end{pmatrix}$$

i per als casos $b = (4, 4)^t$ i $b = (3, 5)^t$.

7 Feu una predicció de com petits canvis en A afecten a la solució x del sistema d'equacions $Ax = b$. Poseu a prova la vostra predicció per a $b = (100, 1)^t$ i les matrius següents

$$A_1 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}, \qquad A_2 = \begin{pmatrix} 1 & 1 \\ 1 & 0.01 \end{pmatrix}.$$

Sabeu donar una explicació del que s'observa?

Mètodes iteratius

8 Estudieu la convergència dels mètodes de Jacobi i de Gauss-Seidel per a la matriu

$$A = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}, \qquad |\rho| < 1.$$

9 Considereu el sistema $Ax = b$ on $A = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & 1 \\ a & 0 & 1 \end{pmatrix}$. Per a quins valors de a convergeix el mètode de Jacobi?

10 Feu servir $(0.33116, 0.70000)^t$ com a punt inicial per resoldre pel mètode de Gauss-Seidel el sistema $Ax = b$. Què passa?

$$A = \begin{pmatrix} 0.96326 & 0.81321 \\ 0.81321 & 0.68654 \end{pmatrix}, \quad b = \begin{pmatrix} 0.88824 \\ 0.74988 \end{pmatrix}.$$

4.2 Pràctiques i problemes

MATLAB®

1 Calculeu el determinant, la transposta, la inversa, els valors propis i els vectors propis de la matriu A definida per:

$$A = \begin{pmatrix} 1 & 2 & 1 \\ -2 & 7 & 3 \\ 1 & -1 & 10 \end{pmatrix}$$

2 Escriu un script de MATLAB® i resol el sistema lineal següent fent ús de la regla de Cramer:

$$\begin{aligned} 2x_1 + 4x_2 + x_3 &= 1\,, \\ 8x_1 - x_2 + 3x_3 &= 0\,, \\ 2x_1 + 5x_2 \phantom{{}+3x_3} &= 0\,. \end{aligned}$$

3 Verifiqueu si un petit canvi a la matriu de coeficients del sistema indueix grans canvis en la solució per als sistemes d'equacions següents:

$$\left\{\begin{array}{rcrcr} 7x & + & 2y & = & 6 \\ x & + & 0.28y & = & 2 \end{array}\right. \qquad \left\{\begin{array}{rcrcr} 7x & + & 2y & = & 6 \\ x & + & 0.285y & = & 2 \end{array}\right.$$

Mètodes directes.

4 Genereu un script en `Matlab` per trobar la solució d'un sistema lineal triangular superior (inferior) $AX = B$ pel mètode de substitució enrera (endavant). Feu jocs de proves.

$$\text{a)} \left\{\begin{array}{rcr} 3x - 2y + z - t & = & 8 \\ 4y - z + 2t & = & -3 \\ 2z + 3t & = & 11 \\ 5t & = & 15 \end{array}\right. ; \qquad \text{b)} \left\{\begin{array}{lcr} 5x & = & -10 \\ x + 3y & = & 4 \\ 3x + 4y + 2z & = & 2 \\ -x + 3y - 6z - t & = & 5 \end{array}\right. .$$

5 Resoleu el sistema $UX = B$ amb $U = (u_{ij})_{15\times 15}$ i i $B = (b_{i1})_{15\times 1}$ definides per

$$u_{ij} = \begin{cases} \cos(ij) & i \le j\,, \\ \\ 0 & i > j\,, \end{cases} \qquad \text{i} \qquad b_{i1} = \tan(i)\,.$$

6 Resoleu el sistema $LX = B$ amb $L = (u_{ij})_{20\times 20}$ i i $B = (b_{i1})_{20\times 1}$ definides per

$$l_{ij} = \begin{cases} i+j & i \geq j\,, \\ \\ 0 & i < j\,, \end{cases} \qquad \text{i} \qquad b_{i1} = i\,.$$

7 Genereu un script en `Matlab` per trobar la solució d'un sistema lineal $Ax = b$ pel mètode d' eliminació gaussiana. Feu jocs de proves.

$$A = \begin{pmatrix} 10 & -7 & 0 & 1 \\ -3 & 2 & 6 & 0 \\ 5 & -1 & 5 & 0 \\ -1 & 0 & 2 & -1 \end{pmatrix} \quad \text{i} \quad b = \begin{pmatrix} 7 \\ 4 \\ 6 \\ -3 \end{pmatrix}.$$

8 Genereu un script en `Matlab` per trobar la solució d'un sistema lineal $Ax = b$ pel mètode de Gauss-Jordan. Feu jocs de proves.

$$A = \begin{pmatrix} 6 & -2 & 2 & 4 \\ 12 & -4 & 2 & 2 \\ 3 & -13 & 9 & 3 \\ -6 & 4 & 1 & -18 \end{pmatrix} \quad \text{i} \quad b = \begin{pmatrix} 12 \\ 34 \\ 27 \\ -38 \end{pmatrix}.$$

9 Adapteu el script en `Matlab` del mètode de Gauss-Jordan per a resoldre sistemes lineals al càlcul de la matriu inversa. Feu jocs de proves.

$$A = \begin{pmatrix} 6 & 2 & 1 & -1 \\ 2 & 4 & 1 & 0 \\ 1 & 1 & 4 & -1 \\ -1 & 0 & -1 & 3 \end{pmatrix}$$

Mètodes compactes.

10 Trobeu una descomposició $A = LU$ de les matrius següents:

$$A = \begin{pmatrix} 2 & -2 & -4 \\ 1 & 1 & -1 \\ 3 & 7 & 5 \end{pmatrix}, \qquad A = \begin{pmatrix} 6 & 2 & 1 & -1 \\ 2 & 4 & 1 & 0 \\ 1 & 1 & 4 & -1 \\ -1 & 0 & -1 & 3 \end{pmatrix}$$

A partir de la descomposició trobada, calculeu el determinant de la matriu A.

11 Trobeu la descomposició LU de la matriu del sistema d'equacions lineals A i després resoleu el sistema d'equacions lineals $Ax = b$ següents:

$$(A\,|\,b) = \left(\begin{array}{ccc|c} 1 & 6 & 0 & 3 \\ 2 & 1 & 0 & 1 \\ 0 & 2 & 1 & 1 \end{array}\right), \qquad (A\,|\,b) = \left(\begin{array}{rrrr|r} 6 & -2 & 2 & 4 & 0 \\ 12 & -8 & 4 & 10 & -10 \\ 3 & -13 & 3 & 3 & -39 \\ -6 & 4 & 2 & -18 & -16 \end{array}\right),$$

12 Calculeu la inversa de la matriu

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 4 & 9 & 16 \\ 1 & 8 & 27 & 64 \\ 1 & 16 & 81 & 216 \end{pmatrix}.$$

pel mètode de Gauss-Jordan i després a partir de la descomposició LU.

13 Avalueu l'eficiència dels mètodes directes (eliminació gaussiana, LU, Txoleski, \, `linsolve`) per al sistema lineal $\mathbf{Ax} = \mathbf{b}$, amb $\mathbf{A}$ la matriu de Hilbert d'ordre n, $\mathbf{x} = (1, 1, \ldots, 1)^t$ preneu $\mathbf{b} = A\,x$ i $n = 6$. La matriu de Hilbert es defineix per $H = (h_{ij}) = \left(\dfrac{1}{i+j-1}\right)$ per a $(1 \le i \le m,\ 1 \le j \le n)$.

14 Trobeu la descomposició QR de la matriu del sistema d'equacions lineals A i després resoleu el sistema d'equacions lineals $Ax = b$ següents:

$$a)\ \left\{\begin{array}{rcr} 6x_1 + 2x_2 - x_3 - x_4 & = & 0 \\ 2x_1 + 4x_2 + x_3 & = & 7 \\ -x_1 + x_2 + 4x_3 - x_4 & = & -1 \\ -x_1 - x_3 + 3x_4 & = & -2 \end{array}\right. ; \qquad b)\ \left\{\begin{array}{rcl} 0.05x_1 + 0.07x_2 + 0.06x_3 + 0.05x_4 & = & 0.23 \\ 0.07x_1 + 0.10x_2 + 0.08x_3 + 0.07x_4 & = & 0.32 \\ 0.06x_1 + 0.08x_2 + 0.10x_3 + 0.09x_4 & = & 0.33 \\ 0.05x_1 + 0.07x_2 + 0.09x_3 + 0.10x_4 & = & 0.31 \end{array}\right. .$$

Mètodes iteratius.

15 Escriviu un programa que implementi els algoritmes de Jacobi Gauss - Seidel i *SOR*. Introduiu un test de convergència. Proveu-lo per a diferents sistemes.

16 Resoleu pel mètode de Jacobi i de Gauss-Seidel els sistemes $Ax = b$ donat per

$$A = \begin{pmatrix} 10 & -1 & 2 & 0 \\ -1 & 11 & -1 & 3 \\ 2 & -1 & 10 & -1 \\ 0 & 3 & -1 & 8 \end{pmatrix} \qquad b = \begin{pmatrix} 6 \\ 25 \\ -11 \\ 15 \end{pmatrix}$$

prenent $x^{(0)} = (0\,,0\,,0\,,0)^\top$ i una tolerància $0.5 \cdot 10^{-12}$. Quantes iteracions calen?

17 Demostreu que el mètode de Jacobi convergeix i, en canvi el de Gauss-Seidel no convergeix, pel sistema següent:

$$\begin{cases} x_1 + 2x_2 - 2x_3 &= 1 \\ x_1 + x_2 + x_3 &= 1 \\ 2x_1 + 2x_2 + x_3 &= 1\,. \end{cases}$$

18 Determineu el radi espectral de les matrius d'iteració dels mètodes de Jacobi i de Gauss-Seidel per resoldre el sistema lineal $Ax = b$ de valors:

$$A = \begin{pmatrix} -4 & 1 & 1 & 1 & 1 \\ 1 & -4 & 1 & 1 & 1 \\ 1 & 1 & -4 & 1 & 1 \\ 1 & 1 & 1 & -4 & 1 \\ 1 & 1 & 1 & 1 & -4 \end{pmatrix}, \qquad b = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}.$$

Trobeu la solució del sistema lineal $Ax = b$ aplicant el mètode més ràpid (dels tres anteriors) fins que la diferència entre una iteració i la següent sigui inferior a $0.5 \cdot 10^{-8}$, comenceu amb $x^{(0)} = 0$. Quantes iteracions són necessàries?

19 Per als sistema d'equacions lineals:

$$\begin{cases} 3x - y + z = 1 \\ x + 4y - z = 3 \\ x - y - 5z = -2 \end{cases} \qquad \begin{cases} 3x + y - z = 2 \\ x + 5y - z = 1 \\ -x + y - 4z = -3 \end{cases}$$

a) Formula el mètode de Jacobi i el mètode de Gauss-Seidel. Estudia la convergència de cada mètode.

b) Calculeu les 10 primeres iteracions per cada mètode partint de $x^0 = 0$

c) Calcula la solució exacta fent ús de funcions de `MATLAB`.

d) Estimeu els errors absolut i relatiu corresponents a la darrera iteració calculada. Quantes xifres significatives heu obtingut?

Mètode dels mínims quadrats.

20 Determineu la solució d'error quadràtic mínim per al sistema lineal següent:

$$\left\{\begin{array}{rcrcrcrcl} x & + & y & + & z & & & = & 2 \\ x & & & + & z & + & 2t & = & 3 \\ x & + & y & & & + & t & = & 4 \\ & - & y & + & 2z & & & = & 2 \\ -x & + & y & - & z & + & t & = & 1 \end{array}\right. .$$

21 Els pesos atòmics de l'oxigen i del nitrogen són aproximadament $O = 16$ i $N = 14$; utilitzeu els pesos moleculars dels sis òxids de nitrogen de la taula per tal d'ajustar els pesos atòmics per mínims quadrats

Compost	NO	N_2O	NO_2	N_2O_3	N_2O_5	N_2O_4
Pes molecular	30.006	44.013	46.006	76.012	108.010	92.011

22 Determineu una funció quadràtica que satisfaci al màxim (error quadràtic mínim) la taula següent:

X	8	10	12	16	20	40
Y	0.88	1.22	1.64	2.72	3.96	11.96

Mètode de la potència.

23 Escriviu una programa que realitzi m passos del mètode de la potència normalitzat quan aquest s'aplica a una matriu A de $n \times n$, a partir d'un vector inicial $x^{(0)}$. En cada pas, imprimiu el vector $x^{(k)}$ i el quocient r_k en curs. Feu un joc de proves amb les matrius de l'exercici anterior.

24 Calculeu, amb quatre xifres significatives, el valor propi dominant de la matriu

$$A = \begin{pmatrix} 3.2 & 8.4 & 5.6 \\ 2.3 & 8.8 & 6.0 \\ 3.5 & 8.4 & 5.3 \end{pmatrix}.$$

25 Escriviu una programa que realitzi m passos del mètode de la potència inversa quan aquest s'aplica a una matriu A de $n \times n$, a partir d'un vector inicial $x^{(0)}$. En cada pas, imprimiu el vector $x^{(k)}$ i el quocient r_k en curs. Feu un joc de proves.

26 Calculeu els valor propis de mòdul màxim i mínim, així com els vectors propis corresponents de la matriu següent:

$$A = \begin{pmatrix} 1 & 2 & -2 & 4 \\ 2 & 12 & 3 & 5 \\ 3 & 13 & 0 & 7 \\ 2 & 11 & 2 & 2 \end{pmatrix}.$$

27 Afegiu el criteri del quocient de Rayleigh per a matrius simètriques als vostres scripts del mètode de la potència. Calculeu els valor propis de mòdul màxim i mínim, així com els vectors propis corresponents de les matrius següents:

$$a) \begin{pmatrix} 0 & 12 & 16 & -15 \\ 12 & 388 & 309 & 185 \\ 16 & 309 & 312 & 80 \\ -15 & 185 & 80 & -600 \end{pmatrix}, \qquad b) \begin{pmatrix} 5 & 4 & 1 & 1 \\ 4 & 5 & 1 & 1 \\ 1 & 1 & 4 & 2 \\ 1 & 1 & 2 & 4 \end{pmatrix}.$$

28 Feu la implementació numèrica del mètode **QR** per trobar els valors propis d'una matriu A de $n \times n$. Recordeu que MATLAB® té una rutina que us fa els càlculs de la descompossició **QR** d'una matriu.

29 Determineu els valors propis de les matrius seguents pel mètode **QR**.

$$a) \begin{pmatrix} 1 & 2 & 1 & 2 \\ 2 & 2 & -1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \qquad b) \begin{pmatrix} 0 & 12 & 16 & -15 \\ 12 & 388 & 309 & 185 \\ 16 & 309 & 312 & 80 \\ -15 & 185 & 80 & -600 \end{pmatrix} \qquad c) \begin{pmatrix} 1 & -4 & -1 & -4 \\ 2 & 0 & 5 & -4 \\ -1 & 1 & -2 & 3 \\ -1 & 4 & -1 & 6 \end{pmatrix}.$$

30 Donada la matriu

$$N = \begin{pmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 \end{pmatrix}$$

calculeu amb MATLAB® una matriu ortogonal Q, tal que la matriu $Q^T N Q$ sigui diagonal.

Descompossició en valors singulars (SVD)

31 La taula següent ens dóna el nombre de bacteris per unitat de volum en funció del temps transcorregut:

Hores x	0	1	2	3	4	5	6
Bacteris y	32	47	65	92	132	190	275

Calculeu una corba del tipus $y = ab^x$ que aproximi aquest núvol de punts. Feu una predicció del nombre de bacteris al cap de 7 hores. Cal que feu us de la rutina `svd` del `Matlab` que dóna la descompossició en valors singulars.

32 La intesitat de radiació d'una font radioactiva ve donada per

$$I = I_0 \, e^{\alpha \, t}$$

Determineu les constants α i I_0 sabent que s'han fet les mesures següents:

t	0.2	0.3	0.4	0.5	0.6	0.7	0.8
I	3.16	2.38	1.75	1.34	1.00	0.74	0.56

D'examen.

33 Doneu tots els valors propis de la matriu A pel mètode QR. Quantes iteracions us calen? Quin és el criteri de parada que useu?

$$A = \begin{pmatrix} 5 & -2 & 0 & 0 & -2 \\ -2 & 5 & -2 & 0 & 0 \\ 0 & -2 & 5 & -2 & 0 \\ 0 & 0 & -2 & 5 & -2 \\ -2 & 0 & 0 & -2 & 5 \end{pmatrix}$$

34 Plantegeu el mètode de la potència per a la matriu

$$A = \begin{pmatrix} 9 & 10 & 8 \\ 10 & 5 & -1 \\ 8 & -1 & 3 \end{pmatrix}$$

per a calcular el valor propi de mòdul màxim, i realitzeu les iteracions necessàries per a determinar el valor propi i el vector propi amb tres decimals correctes. Realment, caldria fer nombroses iteracions per a obtenir un resultat que s'aproximi suficientment a la solució; quins criteris cal utilitzar per a aturar el procés iteratiu?

35 Empreu una tècnica de mínims quadrats per ajustar la taula de dades:

X	0.25	0.50	0.75	1.00	1.25	1.50	1.75
Y	0.40	0.50	0.90	1.28	1.60	1.66	2.02

a funcions del tipus següents:
(35.a) $y = b_0 + b_1 x$ **(35.b)** $y = a_0 + a_1 x + a_2 x^2 + a_3 x^3$
(35.c) $y = Be^{\beta x}$, **(35.d)** $y = Ax^{\alpha}$. Es demana:

a) Explicitar els sistemes lineals resoldre en tots els casos.
b) Donar els valors ajustats, b_0, b_1, a_0, a_1, a_2, a_3, B, β, A i α així com el valor del residu obtingut per cada equació.
c) Representeu conjuntament els punts ($*$), la recta (verd), el polinomi (vermell), la corba $y = Be^{\beta x}$ (blau) i la corba $y = Ax^{\alpha}$ (cyan)
d) Quin d'aquests tipus us sembla el més adequat per representar les dades? Raoneu la resposta.

36 Sigui una funció no lineal $y = f(t)$, amb t una variable real, es vol aproximar f per un *polinomi quadràtic* de variable t. S'ha avaluat f en 6 punts, s'ha obtingut la taula següent:

t_i	8	10	12	16	20	40
y_i	0.88	1.22	1.64	2.72	3.96	11.96

a) Plantejeu el problema de determinar els coeficients del polinomi com un sistema lineal $Ax = b$, i.e. definiu les components del vector d'incògnites x, expresseu A i b en funció de t_i i y_i per a $1 \leq i \leq 8$.

b) Resoleu el problema fent ús de les equacions normals i la factorització LU de la matriu dels sistema d'equacions normals. Doneu la solució obtinguda i calculeu el vector residu.

c) Determineu el polinomi interpolador en els tres primers punts donats, $t_i = 8,\ 10,\ 12$. Doneu l'equació del polinomi resultant.

d) Determineu el polinomi interpolador en els tres darrers punts donats, $t_i = 16,\ 20,\ 40$. Doneu l'equació del polinomi resultant.

e) Avalueu els tres polinomis obtinguts en $t = 11.3$ i en $t = 29.0$ Comenteu les diferències entre els valors de les solucions trobades. Tots els resultats són creïbles? Quins us mereixen més confiança i per què.

Appendices

A Àlgebra Matricial

Alguns resultats que utilitzem en aquest curs sobre álgebra lineal i matrius.

Notació

Per a matrius i vectors emprarem la notació següent:

$$A = (a_{ij})_{\substack{1\le i\le m \\ 1\le j\le n}} \Rightarrow A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{1n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}.$$

$$u = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} \quad \Rightarrow u = (u_1, u_2, \dots, u_n)^t = (u_i)^t_{1\le i\le n}.$$

Tipus de matrius

Una matriu $A = (a_{ij})\ {}_{1\le i,j\le n}$, $a_{ij} \in \mathbb{R}$ es diu:

- Ortogonal si $A^t = A^{-1}$ o $A^t A = A A^t = I_{n\,n}$.
- Unitària si $U^* U = U U^* = I_{n\,n}$, $\quad (u_{ij} \in \mathbb{C})$.
- Simètrica si $A^t = A$. Hermitiana si $(U^t)^* = U$.
- Tridiagonal si $a_{ij} = 0$ si $|i-j| > 1$.
- Definida positiva si $x^t A x > 0$, $\forall x \neq 0$.
- Estrictament diagonal dominant si $|a_{ii}| > \sum\limits_{\substack{1\le j\le n \\ i\neq j}} |a_{ij}|$.

- Diagonal dominant si $|a_{ii}| \geq \sum_{\substack{1\leq j\leq n \\ i\neq j}} |a_{ij}|\,.$

Producte escalar de vectors

Definició 9 *El producte escalar és una aplicació de $\mathbb{R}^n \times \mathbb{R}^n$ en $\mathbb{R}$, que notarem per $\langle\cdot,\cdot\rangle$, que verifica les propietats següents:*

1) $\langle u,u\rangle \geq 0\,,\ (\langle u,u\rangle = 0 \Leftrightarrow u = 0)$

2) $\langle u,v\rangle \leq \langle v,u\rangle$,

3) $\langle u, \alpha v + w\rangle = \alpha\,\langle u,v\rangle + \langle u,w\rangle$.

Val la desigualtat de Cauchy-Schwarz

4) $|\,\langle u,v\rangle\,|^2 \leq \langle u,u\rangle\,\langle v,v\rangle$.

Per a dos vectors de u i v de $\mathbb{R}^n$, definim el producte escalar com

$$\langle u,v\rangle = u_1v_1 + \cdots + u_nv_n\,.$$

Normes vectorials

Definició 10 (Norma d'un vector) *Una norma és una aplicació de $\mathbb{R}^n$ en $\mathbb{R}$, que notarem per $||\cdot||$, que verifica les propietats següents:*

1) $||u|| \geq 0\,,\ (||u|| = 0 \Leftrightarrow u = 0)$

2) $||ku|| \leq |k|\,||u||\,,$

3) $||u+v|| \leq ||u|| + ||v||\,.$

Val la propietat multiplicativa

4) $|\langle u,v\rangle| \leq ||u||_2 \cdot ||v||_2\,, \qquad ||u||_2 = \langle u,u\rangle^{\frac{1}{2}}.$

Definició 11 *Si $u = (u_1\,,\cdots, u_n)$ és un vector qualsevol de $\mathbb{R}^n$, les normes vectorials més usuals són:*

- ***Norma-1*** $\quad ||u||_1 = \sum_{i=1}^{n} |u_i|\,.$

- ***Norma-2*** $||u||_2 = \sqrt{\sum_{i=1}^{n} u_i^2}\,.$

- ***Norma infinit*** $||u||_\infty = \max_{1\le i\le n} |u_i|\,.$

Teorema A.1.1 *Equivalència de normes*

Les normes $||u||_1$, $||u||_2$ i $||u||_\infty$ són equivalents, vol dir que per $c_{qp}\,, C_{qp} > 0$ adients,

$$c_{qp}||u||_p \le ||u||_q \le C_{qp}||u||_p\,, \qquad \forall u \in \mathbb{R}^n\,.$$

Convergència de successions Una successió $\left\{\mathbf{x}^{(k)}\right\}_{k=1,\ldots,\infty}$ de vectors de $\mathbb{R}^n$ es diu que convergeix a $\mathbf{x}$ si, donat un ϵ existeix un n tal que

$$||\mathbf{x}^{(k)} - \mathbf{x}|| < \epsilon \qquad k > n$$

i s'escriu $\lim_{k\to\infty} \mathbf{x}^{(k)} = \mathbf{x}$. I és equivalent a la convergència per components: $\lim_{k\to\infty} \mathbf{x}_i^{(k)} = \mathbf{x}_i$

Normes matricials

Si $A = (a_{ij})_{1\le i,j\le n}$ és una matriu qualsevol de $\mathbb{R}^{n\times n}$

Definició 12 (Norma d'una matriu) *Una norma és una aplicació del conjunt de totes les matrius reals $\mathbb{R}^{m\times n}$ en $\mathbb{R}$, que notarem per $||\cdot||$, que verifica les propietats següents:*

1) $||A|| \ge 0\,,\ (||A|| = 0 \Leftrightarrow A = 0)$

2) $||kA|| \le |k|\,||A||\,,$

3) $||A + B|| \le ||A|| + ||B||\,.$

Adicionalment la propietat, que no verifiquen totes les normes, condició de **multiplicativa**

4) $||A \cdot B|| \le ||A|| \cdot ||B||\,.$

Donada una matriu, A i un vector x qualsevol, Ax és el vector transformat, cal que es compleixi la condició de **consistència**

$$||Ax|| \le ||A|| \cdot ||x||$$

Teorema La expressió

$$||A|| = \max_{x, x \neq 0} \frac{||Ax||}{||x||} = max \left\{ ||Ax|| \ : \ ||x|| = 1 \right\} \tag{A.1.1}$$

defineix una norma matricial compatible amb la norma vectorial emprada en l'expressió.

Definició 13 *Si $A = (a_{ij})_{1 \leq i,j \leq n}$ és una matriu qualsevol de $\mathbb{R}^{n \times n}$, les normes matricials compatibles amb les vectorials definides amb la mateixa notació, són:*

- ***Norma-1*** $||A||_1 = \max_j \sum_{i=1}^{n} |a_{ij}| \, .$
- ***Norma-2*** $||A||_2 = \max_i \sqrt{\lambda_i} \, , \quad \lambda_i$ *vap de* $A^t A$
- ***Norma infinit*** $||A||_\infty = \max_i \sum_{j=1}^{n} |a_{ij}| \, .$
- ***Norma de Frobenius*** $||A||_2 = \sqrt{\sum_{i,j=1}^{n} a_{ij}^2} = \sqrt{tr(A^t A)}.$

Transformacions ortogonals

Una matriu A de $\mathbb{R}^{n \times n}$ sempre verifica

$$\langle x, Ay \rangle = \left\langle A^t x, y \right\rangle, \qquad \forall x \in \mathbb{R}^n, y \in \mathbb{R}^n$$

Les matrius ortogonals Q, conserven el producte escalar i la norma euclídea; es a dir

$$\langle Qx, Qy \rangle = \langle x, y \rangle \, , \qquad \forall x, y \in \mathbb{R}^n$$

$$||Qx||_2 = ||x||_2, \qquad \forall x \in \mathbb{R}^n$$

En aquest cas diu que les normes $||A||_2$ i $||x||_2$ són invariants per tranformacions ortogonals.

Matrius convergents

Una matriu $A = (a_{ij})_{1 \leq i,j \leq n}$ quadrada es diu **convergent** si les potències de la matriu tenen les components amb límit zero,

$$\lim_{k \to \infty} (A^k)_{ij} = 0.$$

Teorema A.1.2 *Una matriu A és **convergent** si i només si, $\rho(A) < 1$.*

Example A.1.1

$$A = \begin{pmatrix} 1/2 & 0 \\ 1/4 & 1/2 \end{pmatrix} \qquad \Rightarrow \qquad A = \begin{pmatrix} 1/2^k & 0 \\ k/2^{k+1} & 1/2^k \end{pmatrix}$$

Radi espectral

Es defineix el **radi espectral** de una matriu A, i es nota per $\rho(A)$ com el màxim dels mòduls del valors propis de la matriu,

$$\rho(A) = \max_i \{|\lambda_i| \, : \, Av_i = \lambda_i v_i\}$$

Geomètricament representa el radi del cercle mínim que conté a tots els valors propis de la matriu A.

Teorema A.1.3 *El radi espectral d'una matriu és una* ***fita inferior*** *de totes les normes multiplicatives de la matriu,*

$$\rho(A) \leq ||A||_r \quad r = \{1, 2, \infty\} \tag{A.1.2}$$

B Vectors i valors propis

Sigui $A = (a_{i\,j})_{n\times n}$ per $a_{i\,j} \in \mathbb{R} \quad {}_{1\leq i,j\leq n}$, A matriu quadrada de nombres reals; $\lambda \in \mathbb{R}$ un real i $v \in \mathbb{R}^n$ un vector.

Definició 1 (Vector i valor propi) *Si l'equació* $\boldsymbol{Av} = \boldsymbol{\lambda v}$ *té alguna solució no trivial* $(v \neq 0)$ *es diu que* $\boldsymbol{\lambda}$ *és un valor propi i* $\boldsymbol{v}$ *un valor propi de la matriu* A*.*

Definició 2 (Espectre) *Conjunt format per tots els valors propis d'una matriu:*

$$spec(A) = \{\lambda_1, \lambda_2, \ldots, \lambda_n, \,/\, \lambda_k \text{valor propi } A\}\ .$$

Definició 3 (Radi espectral) *Valor propi de mòdul màxim*

$$\rho(A) = \max\left(\{|\lambda_1|, \ldots, |\lambda_n|, \,/\, \lambda_k \text{valor propi } A\}\right)$$

Definició 4 (Polinomi característic) *Polinomi associat a una matriu, les arrels del qual són els valors propis de la matriu* A*:*

$$p(\lambda) = det(A - \lambda\, I) = 0\,.$$

Propietats Si $\boldsymbol{v}$ i $\boldsymbol{\lambda}$ compleixen $\boldsymbol{Av} = \boldsymbol{\lambda v}$, llavors

$$\begin{aligned} (-A)v &= -\lambda v\,, \\ (A - \mu\, I)v &= (\lambda - \mu)v\,, \\ A^{-1}v &= \frac{1}{\lambda}v\,, \\ (A - \mu\, I)^{-1}v &= \frac{1}{(\lambda - \mu)}v\,, \end{aligned} \tag{B.1.1}$$

Teorema B.1.1 *El conjunt* $\left\{v^{(1)}, v^{(2)}, \ldots v^{(k)}\right\}$ *de vectors propis de valor propi* $\lambda_1, \lambda_2, \ldots, \lambda_k$*, tots diferents, és un conjunt de vectors linealment independent.*

Teorema de Schur Dues matrius, A i B quadrades d'ordre n es diuen **semblants** si existeix una matriu S tal que $A = S^{-1}\,B\,S$. Una matriu és **diagonalitzable** si és semblant a una matriu diagonal.

Teorema B.1.2 *A i B semblants, λ i v tals que $Av = \lambda v$, llavors λ és un valor propi de B de un vector propi Sv.*

Teorema B.1.3 *Teorema de Schur*

Per a qualsevol matriu quadrada A, existeix una matriu U ortogonal tal que $T = U^{-1}AU$ és triangular superior.

Els elements de la diagonal de T són els valors propis de la matriu A, en ser T una matriu triangular.

C Factorització QR

La factorització QR expressa la matriu A com el producte de dues matrius, una ortogonal ($Q^tQ = I$) i l'altre triangular superior (R).

Per transformar una matriu A en triangular superior s'apliquen transformacions ortogonals, per exemple transformacions de Householder.

Definició 1 (Matriu de Householder) *Donat un vector $v \in \mathbb{R}^n$ definim la transformació associada a v per*

$$H_v = \begin{cases} I, & v = 0, \\ I - \dfrac{2}{v^t v} v v^t, & v \neq 0. \end{cases} \tag{C.2.1}$$

La matriu H_v de la transformació és simètrica, ortogonal i verifica $H_v^2 = I$.

Les característiques més importants d'aquesta transformació són:

- Si $x \perp v \;\Rightarrow H_v x = x$.
- Si $x \parallel v \;\Rightarrow H_v x = -x \;\; H_v v = -v$.
- Si $||x|| = ||v||,\; v = x - y \;\Rightarrow H_v x = y$.

Un exemple de com les matrius de Householder transformen una matriu A, en una matriu triangular superior, tots els canvis aplicats formen les matrius Q i R de la descomposició.

$$\mathbf{A} \equiv \begin{pmatrix} 1 & -1 & -1 \\ 2 & 0 & 1 \\ -2 & 7 & 1 \end{pmatrix} \;\Rightarrow\; \mathbf{R} \equiv \begin{pmatrix} 3 & -5 & -1/3 \\ 0 & 5 & 19/15 \\ 0 & 0 & -17/15 \end{pmatrix} \;\wedge\; \mathbf{Q} \equiv \begin{pmatrix} 1/3 & 2/15 & 14/15 \\ 2/3 & 2/3 & -1/3 \\ -2/3 & 11/15 & 2/15 \end{pmatrix}.$$

En aquest mètode s'aplica la tercera propietat de les matrius de Householder prenent x el vector columna a transformar en cada pas; per una matriu 3×3 cal reduïr 2 columnes. La matriu R és la matriu resultant d'aplicar les dues transformacions a la matriu A. La matriu Q és la matriu de les transformacions de Householder.

Example C.2.1

$$\mathbf{A} \equiv \begin{pmatrix} 1 & -1 & -1 \\ 2 & 0 & 1 \\ -2 & 7 & 1 \end{pmatrix} \Rightarrow$$

$$x^{(1)} = \begin{pmatrix} 1 \\ 2 \\ -2 \end{pmatrix}, \ \left\| x^{(1)} \right\| = 3, \ \Rightarrow \ y^{(1)} = \begin{pmatrix} 3 \\ 0 \\ 0 \end{pmatrix}, \ v^{(1)} = x^{(1)} - y^{(1)} = \begin{pmatrix} -2 \\ 2 \\ -2 \end{pmatrix} \ \Rightarrow \ H_1$$

$$H_1 = I - \frac{2}{v^t v} v v^t = I - \frac{2}{12} \begin{pmatrix} -2 \\ 2 \\ -2 \end{pmatrix} (-2, 2, -2) = \begin{pmatrix} 1/3 & 2/3 & -2/3 \\ 2/3 & 1/3 & 2/3 \\ -2/3 & 2/3 & 1/3 \end{pmatrix} \ \Rightarrow \mathbf{H_1\, A}$$

$$\mathbf{H_1\, A} \equiv \begin{pmatrix} 3 & -5 & -1/3 \\ 0 & 4 & 1/3 \\ 0 & 3 & 5/3 \end{pmatrix}$$

$$x^{(2)} = \begin{pmatrix} -5 \\ 4 \\ 3 \end{pmatrix} \ \Rightarrow \ y^{(2)} = \begin{pmatrix} -5 \\ \sqrt{4^2+3^2} \\ 0 \end{pmatrix} = \begin{pmatrix} -5 \\ 5 \\ 0 \end{pmatrix} \ \Rightarrow \ v^{(2)} = x^{(2)} - y^{(2)} = \begin{pmatrix} 0 \\ -1 \\ 3 \end{pmatrix} \ \Rightarrow \ H_2$$

$$H_2 = I - \frac{2}{v^t v} v v^t = I - \frac{2}{10} \begin{pmatrix} 0 \\ -1 \\ 3 \end{pmatrix} (-0, -1\ 3) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 4/5 & 3/5 \\ 0 & 3/5 & -4/5 \end{pmatrix} \ \Rightarrow \mathbf{H_2\, H_1\, A}$$

$$\mathbf{H_2\, H_1\, A} \equiv \begin{pmatrix} 3 & -5 & -1/3 \\ 0 & 5 & 19/15 \\ 0 & 0 & -17/15 \end{pmatrix} \longrightarrow \text{matriu } \mathbf{R}$$

$$\mathbf{H_1\, H_2} \equiv \begin{pmatrix} 1/3 & 2/15 & 14/15 \\ 2/3 & 2/3 & -1/3 \\ -2/3 & 11/15 & 2/15 \end{pmatrix} \longrightarrow \text{matriu } \mathbf{Q}$$

D Preconditioning

Es pot millorar la convergència i l'estabilitat de la majoria de métodes iteratius transformant el sistema lineal per tenir un espectre més favorable. Aquesta transformació es realitza aplicant una segona matriu, anomenada matriu precondicionadora, al sistema d'equacions. Aquest procés transforma el sistema lineal $Ax = b$ en un sistema equivalent $\widetilde{A}\widetilde{x} = \widetilde{b}$

El precondicionador ideal (A^{-1}) transforma la matriu de coeficients A en una matriu d'identitat. A la pràctica, trobar un bon precondicional requereix compensacions. La transformació (M) potser de tres tipus:

- precondicionament per l'esquerre $(M^{-1}A)\,x = (M^{-1}b)$.
- precondicionament per la dreta $(AM^{-1})\,(Mx) = b\,.$
- split; usualment per matrius simètriques, la matriu precondicionadora M tal que $M = H\,H^t$ (split) per mantenir la simetria del sistema transformat $(H^{-1}AH^{-t})\,H^t x = (H^{-1}b)$.

Direct solvers:
Sequential, loosing sparsity

Iterative solvers:
easy parallel and sparse,
but possibly slowly convergent

Combination of both methods:

Include preconditioner $M \approx A$ in the form $M^{-1} A x = M^{-1} b$, such that

- M is easy to deal with in parallel (reduced approximate direct solver)
- spectrum of $M^{-1} A$ is much better clustered

Or include preconditioner $M \approx A^{-1}$ in the form $M A x = M b$, such that

- M is easy to deal with in parallel (reduced approximate inverse)
- spectrum of $M A$ is much better clustered