Curso MS993/MT404
Métodos computacionais em álgebra linear

Responsável[1] : Prof^o Eduardo Abreu - DMA - IMECC - UNICAMP
eabreu@ime.unicamp.br

2^o Semestre 2016
6^a 08h-12h, Sala 124 no IMECC

http://www.ime.unicamp.br/∼ms993 ou
http://www.ime.unicamp.br/∼mt404

[1] Comentários, críticas ou sugestões são todos bem-vindos; favor, enviar para *eabreu@ime.unicamp.br*

O curso MS933/MT404

Métodos Computacionais em Álgebra Linear

- **O curso MS933/MT404**

  Métodos Computacionais em Álgebra Linear

- Fornecer uma visão teórica e desenvolver habilidades práticas computacionais de métodos numéricos aplicados para resolver problemas de álgebra linear numérica em grande escala. Particular ênfase recai sobre sistemas de equações lineares de grande porte e de problemas relacionados.

- Com efeito, observa-se que o desenvolvimento de tais métodos computacionais em álgebra linear tem sua base em resultados matemáticos rigorosos e, portanto, não são dependentes de uma linguagem de programação particular.

David M. Young (1971)

"*The availability of very high-speed computers with large, fast memories has made it possible to obtain accurate numerical solutions of mathematical problems which, although algorithms for handling them were well known previously, could not be used in practice because the number of calculations required would have been prohibitive. A problem for which this is particularly true is that of solving a large system of linear algebraic equations where the matrix of the system is very sparse.... These problems, in turn, arise in studies in such areas as neutron diffusion, fluid flow, elasticity, steady-state heat flow, and weather prediction...*"

ORGANIZAÇÃO DA EMENTA EM TÓPICOS

Pretende-se discutir os itens 1) e 2) a seguir ao longo de todo o curso, incluindo aplicações pertinentes de modelos $Ax = b$

ORGANIZAÇÃO DA EMENTA EM TÓPICOS

1) Algoritmos básicos para operações com vetores e matrizes. Matrizes com estruturas especiais: operações e armazenamento. Normas.
2) Análise de sensibilidade de sistemas lineares: Número de condição. Condicionamento de um sistema linear.
3) Métodos diretos para resolução de sistemas lineares: Fatorações: LU e Cholesky.
4) Métodos iterativos (*indiretos*) para resolução de sistemas lineares: 4.1) Métodos de ponto fixo ou métodos estacionários (e.g., Richardson, Gauss-Jacobi, Gauss-Seidel e) e 4.2) Métodos dos gradientes conjugados (Métodos em Subespaços de Krylov).
5) O problema de quadrados mínimos. Fatorações QR e SVD.

Motivation

- Nonlinear dynamics models (e.g. PDEs) provide a quantitative description for many central models in physical, biological, engineering sciences, and more...

- Although a rich development of mathematical theories to solve **PDEs** and **Nonlinear models** have been achieved, analytical techniques provide only a limited account for the array of complex phenomena governed by such models
  – *The Abel Symposium (2010 Edition)*, Olso (Norway).
  – *International Congress of Mathematicians (ICM 2014)*, Seoul (Korea).
  – *The International Congress on Industrial and Applied Mathematics (ICIAM 2015)*, Pequim (China)

- Thus, **numerical analysis and computational methods for solving nonlinear models** have emerged as the most versatile tool **to complement** mathematical theory and real-world experiments

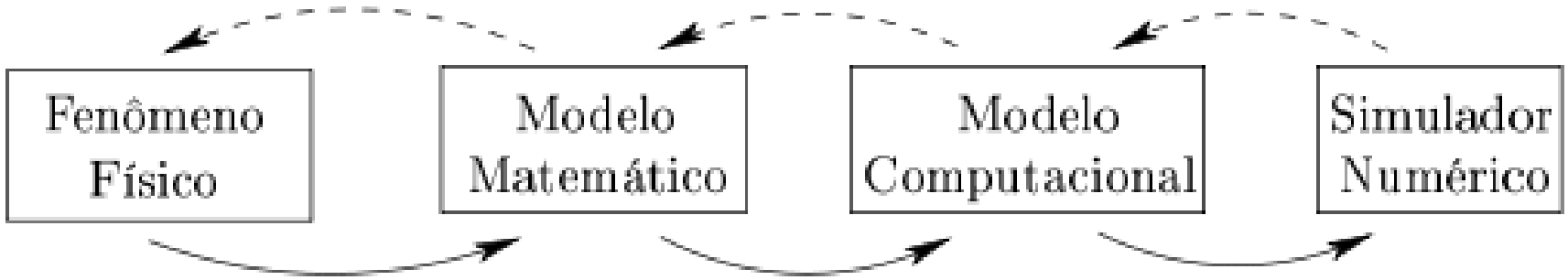


FIG 1. TIPOS DE ERROS E INCERTEZAS

We are primary interested in numerically solving

$$Ax = b$$

where matrix A is $n \times n$ and nonsingular, along with vector x is $n \times 1$ and vector b is $n \times 1$ (with $b \neq 0$). (We will also discuss the case when A is singular)

MOTIVATION:

GOOGLE AND OCEAN CIRCULATION

GOOGLE: *Kurt Bryan and Tanya Leise, The US 25,000,000,000 Eigenvector: The Linear Algebra behind Google, SIAM Rev., 48(3) (2006) 569-581*

OCEAN CIRCULATION: *M. B. van Gijzen, C. B. Vreugdenhil, and H. Oksuzoglu, The Finite Element Discretization for Stream-Function Problems on Multiply Connected Domains, J. Comp. Phys., 140 (1998) 30-46.*

More good references ...

Tobin A. Driscoll, Kim-Chuan Toh, and Lloyd N. Trefethen. From Potential Theory to Matrix Iterations in Six Steps. SIAM Rev., 40(3) (2006) 547-578.

M. Benzi, G. H. Golub, J. Liesen. Numerical solution of saddle point problems Acta numerica, 14 (2005) 1-137.

Yousef Saad, Henk A. van der Vorstb. Iterative solution of linear systems in the 20th century Journal of Computational and Applied Mathematics, 123(12) (2000) 1-33.

Y. Saad. Practical use of polynomial preconditionings for the conjugate gradient method SIAM Journal on Scientific and Statistical Computing, 6(4) (1985) 865-881.

Mark Embree, Josef A. Sifuentes, Kirk M. Soodhalter, Daniel B. Szyld, and Fei Xue. Short-Term Recurrence Krylov Subspace Methods for Nearly Hermitian Matrices. SIAM Journal on Matrix Analysis and Applications, 33(2) (2012) 480-500.

Klaus Schiefermayr. A Lower Bound for the Norm of the Minimal Residual Polynomial Constructive Approximation, 33(3) (2011) 425-432.

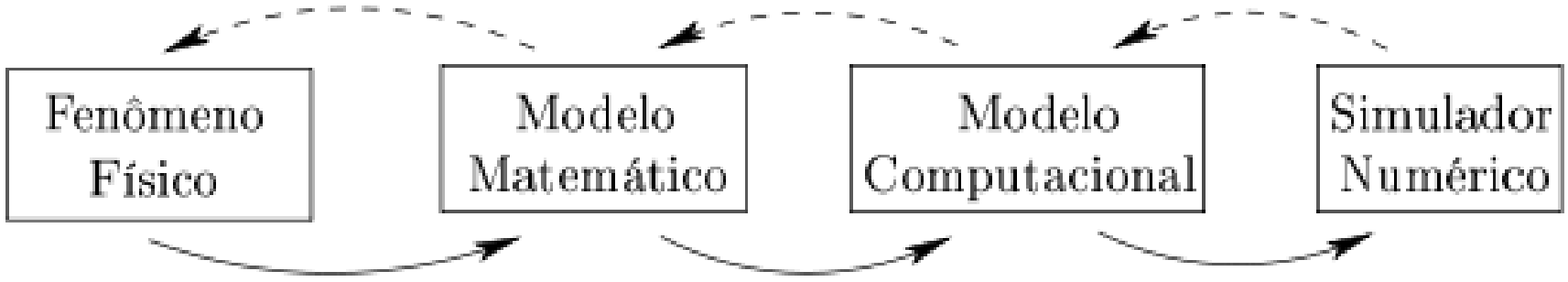


FIG 1. TIPOS DE ERROS E INCERTEZAS

We are primary interested in numerically solving

$$Ax = b$$

where matrix A is $n \times n$ and nonsingular, along with vector x is $n \times 1$ and vector b is $n \times 1$ (with $b \neq 0$). (We will also discuss the case when A is singular)

INTRODUCTORY PART:

CLASSIFICATION OF NUMERICAL METHODS FOR SOLVING $Ax = b$

Introductory part: Classification of numerical methods for solving $Ax = b$

- The numerical solution methods for linear systems of the form $Ax = b$ are broadly classified into **direct methods** and the **iterative methods**.

- For large systems, direct methods become impractical due to the phenomenon of *fill-in* or *fill*, caused by the generation of new entries during the factorisation phase.

- Iterative methods generate a sequence of approximations that only **converges in the limit** to the solution. Beginning with a **given approximate solution**, these methods modify the components of the approximation in each iteration, until a required convergence is achieved.

- Thus, **roughly speaking** numerical methods for solving linear systems of equations can generally be divided into two classes:

- **Direct methods**. In the absence of roundoff error such methods would yield the exact solution within a finite number of steps.

- **Iterative methods**. These are methods that are useful for problems involving special, very large matrices.

OVERVIEW OF DIRECT METHODS

- Direct Solution Methods
  - Gaussian Elimination and LU Decomposition
  - Special Matrices
  - Ordering Strategies

Gaussian Elimination and LU Decomposition

Assumptions:

- The given matrix A is real, $n \times n$ and nonsingular.
- The problem $A\mathbf{x} = \mathbf{b}$ therefore has a unique solution $\mathbf{x}$ for any given vector $\mathbf{b}$ in $\mathcal{R}^n$.

The basic direct method for solving linear systems of equations is **Gaussian elimination**. The bulk of the algorithm involves only the matrix A and amounts to its decomposition into a product of two matrices that have a simpler form. This is called an **LU decomposition**.

How to use it

- solve linear equations when A is in upper triangular form. The algorithm is called *backward substitution*.
- transform a general system of linear equations into an upper triangular form, where backward substitution can be applied. The algorithm is called *Gaussian elimination*.

Backward Substitution

Start easy:

- If A is diagonal:

$$A = \begin{pmatrix} a_{11} & & & \\ & a_{22} & & \\ & & \ddots & \\ & & & a_{nn} \end{pmatrix},$$

then the linear equations are uncoupled and the solution is obviously

$$x_i = \frac{b_i}{a_{ii}}, \quad i = 1, 2, \ldots, n.$$

Triangular Systems

An **upper triangular** matrix

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ & a_{22} & \ddots & \vdots \\ & & \ddots & \vdots \\ & & & a_{nn} \end{pmatrix},$$

where all elements below the main diagonal are zero:
$a_{ij} = 0, \forall i > j$.
Solve *backwards*: The last row reads $a_{nn}x_n = b_n$, so $x_n = \frac{b_n}{a_{nn}}$.
Next, now that we know x_n, the row before last can be written as $a_{n-1,n-1}x_{n-1} = b_{n-1} - a_{n-1,n}x_n$, so $x_{n-1} = \frac{b_{n-1} - a_{n-1,n}x_n}{a_{n-1,n-1}}$. Next the previous row can be dealt with, etc. We obtain the **backward substitution** algorithm.

Computational Cost - Naive but Useful

What is the cost of this algorithm? In a simplistic way we just count each floating point operation (such as $+$ and $*$) as a *flop*. The number of flops required here is

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

Simplistic but not ridiculously so: doesn't take into account data movement between elements of the computer's memory hierarchy. In fact, concerns of data locality can be crucial to the execution of an algorithm. The situation is even more complex on multiprocessor machines. *But still: gives an idea...*

LU Decomposition / Factorization

The Gaussian elimination procedure *decomposes* A into a product of a unit lower triangular matrix L and an upper triangular matrix U. This is the famous **LU decomposition**. Together with the ensuing backward substitution the entire solution algorithm for $A\mathbf{x} = \mathbf{b}$ can therefore be described in three steps:

1. *Decomposition*:
$$A = LU$$
2. *Forward substitution*: solve
$$L\mathbf{y} = \mathbf{b}.$$
3. *Backward substitution*: solve

Pivoting - It is noteworthy that A_ε, $0 < \varepsilon << 1$

In a nutshell, perform permutations to increase numerical stability.
Trivial but telling examples:
For

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

or

$$A_\varepsilon = \begin{pmatrix} \varepsilon & 1 \\ 1 & 0 \end{pmatrix}$$

G.E. will fail (for A) or perform poorly (for A_ε).

Nothing wrong with the problem, it's the algorithm to blame!

- Partial pivoting (not always stable but standard)
- Complete pivoting (stable but too expensive)
- Rook pivoting

Pivoting - It is noteworthy that A_ε, $0 < \varepsilon << 1$ (Cont.)

In a nutshell, perform permutations to increase numerical stability.
Trivial but telling examples:
For

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

or

$$A_\varepsilon = \begin{pmatrix} \varepsilon & 1 \\ 1 & 0 \end{pmatrix}$$

G.E. will fail (for A) or perform poorly (for A_ε).

Nothing wrong with the problem, it's the algorithm to blame!

- Partial pivoting (not always stable but standard)
- Complete pivoting (stable but too expensive)
- Rook pivoting

Neal and Poole (1992) presented the so-called **Rook pivoting** strategy. In short, this pivoting strategy appears to be intermediate between partial pivoting and complete pivoting in terms of efficiency and stability. For more details see Project 1 !

Special Matrices

- **Symmetric Positive Definite.** A matrix A is symmetric positive definite (SPD) if $A = A^T$ and $\mathbf{x}^T A\mathbf{x} > 0$ for any nonzero vector $\mathbf{x} \neq 0$. (All SPD matrices necessarily have positive eigenvalues.)
  In the context of linear systems – **Cholesky Decomposition**:

$$A = FF^T.$$

  - No pivoting required
  - Half the storage and work. (But still $O(n^2)$ and $O(n^3)$ respectively.)

Special Matrices (Cont.)

- **Narrow Banded.**

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1q} & & & & \\ \vdots & \ddots & \ddots & \ddots & & & \\ a_{p1} & \ddots & \ddots & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \ddots & \ddots & a_{n-q+1,n} \\ & & & \ddots & \ddots & \ddots & \vdots \\ & & & & a_{n,n-p+1} & \cdots & a_{nn} \end{pmatrix}.$$

- Significant savings, if bandwidth is small: $O(n)$ work and storage.

A useful numerical tip

Never invert a matrix explicitly unless your life depends on it.
In MATLAB, choose `backslash` over `inv`.
Reasons:

- More accurate and efficient
- For banded matrices, great saving in storage

LU vs. Gaussian Elimination (why store L?)

If you did all the work, might as well record it!

One good reason: solving linear systems with multiple right hand sides.

Permutations and Reordering Strategies

Permutations and Reordering Strategies

Riddle: Which matrix is better to work with?

$$A = \begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & 0 & 0 & 0 \\ \times & 0 & \times & 0 & 0 \\ \times & 0 & 0 & \times & 0 \\ \times & 0 & 0 & 0 & \times \end{pmatrix}.$$

$$B = \begin{pmatrix} \times & 0 & 0 & 0 & \times \\ 0 & \times & 0 & 0 & \times \\ 0 & 0 & \times & 0 & \times \\ 0 & 0 & 0 & \times & \times \\ \times & \times & \times & \times & \times \end{pmatrix}.$$

B is a matrix obtained by swapping the first and the last row and column of A.

Permutation Matrices

$$(PAP^T)(P\mathbf{x}) = P\mathbf{b}.$$

Look at $P\mathbf{x}$ rather than $\mathbf{x}$, as per the performed permutation.

- If A is symmetric then so is PAP^T. We can define the latter matrix as B and rewrite the linear system as

$$B\mathbf{y} = \mathbf{c},$$

  where $\mathbf{y} = P\mathbf{x}$ and $\mathbf{c} = P\mathbf{b}$.
- In the example $B = PAP^T$ where P is a permutation matrix associated with the vector $\mathbf{p} = (n, 2, 3, 4, \ldots, n-2, n-1, 1)^T$.

At least two possible ways of aiming to reduce the storage and computational work:

- Reduce the bandwidth of the matrix.
- Reduce the expected fill-in in the decomposition stage.

One of the most commonly used tools for doing it is **graph theory**.

Putting in $\times$ to indicate a nonzero element, we have

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

That is, L and U have many more nonzeros than A. These nonzero locations that appear in L and U and not in A are called *fill-in*. On the other hand, if we cyclically permute the rows and columns of A, we have

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

That is, the factorization of PAP^T has *no* fill-in.

Sparse matrices, graphs, and tree elimination (Cont.)

A sparse matrix A can be viewed as an *adjacency matrices* for an associated graphs: make one node for each row, and connect node i to node j if $A_{ij} \neq 0$. The graphs for the two "arrow" matrices above are:

These graphs of both our example matrices are *trees*, and they differ only in how the nodes are labeled. In the original matrix, the root node is assigned the first label; in the second matrix, the root node is labeled after all the children. Clearly, the latter label order is superior for Gaussian elimination.

This turns out to be a general fact: if the graph for a (structurally symmetric) sparse matrix S is a tree, and if the labels are ordered so that each node appears after any children it may have, then there is no fill-in: that is, L and U have nonzeros only where S has nonzeros.

Let us see one more example for a structurally symmetric sparse matrix S. In what follows, we have $S = A$ or $S = \hat{A}$.

More good references ... David S. Watkins, Fundamentals of Matrix Computations, New Jersey: John Wiley & Sons (2 ed., 2002) e (3 ed., 2010).

Timothy A. Davis, Direct methods for sparse linear systems (Fundamentals of algorithms Series), PA, SIAM (2006).

$$A = \begin{pmatrix} 1 & & & 1 & & & 1 \\ & 1 & & 2 & & & 2 \\ & & 1 & 3 & & & 3 \\ 1 & 2 & 3 & 15 & & & 18 \\ & & & & 1 & & 5 \\ & & & & & 1 & 6 \\ 1 & 2 & 3 & 18 & 5 & 6 & 92 \end{pmatrix} \qquad L = \begin{pmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & 1 & & & & \\ 1 & 2 & 3 & 1 & & & \\ & & & & 1 & & \\ & & & & & 1 & \\ 1 & 2 & 3 & 4 & 5 & 6 & 1 \end{pmatrix}$$

$$\hat{A} = \begin{pmatrix} 92 & 6 & 5 & 18 & 3 & 2 & 1 \\ 6 & 1 & & & & & \\ 5 & & 1 & & & & \\ 18 & & & 15 & 3 & 2 & 1 \\ 3 & & & 3 & 1 & & \\ 2 & & & 2 & & 1 & \\ 1 & & & 1 & & & 1 \end{pmatrix} \qquad \hat{L} = \begin{pmatrix} 9.5917 & & & & & & \\ 0.6255 & 0.7802 & & & & & \\ 0.5213 & -0.4180 & 0.7440 & & & & \\ 1.8766 & -1.5047 & -2.1601 & 2.1327 & & & \\ 0.3128 & -0.2508 & -0.3600 & 0.5899 & 0.6014 & & \\ 0.2085 & -0.1672 & -0.2400 & 0.3933 & -0.7075 & 0.4644 & \\ 0.1043 & -0.0836 & -0.1200 & 0.1966 & -0.3538 & -0.8444 & 0.3015 \end{pmatrix}$$

Figure. Top: arrowhead matrix A $\in R^{7\times 7}$ and its Cholesky factor $L = chol(A)$. Bottom: effect of reversing the numbering.

Edges and Vertices: Optimality criteria

- How to assure that the amount of work for determining the ordering does not dominate the computation. As you may already sense, determining an 'optimal' graph may be quite a costly adventure.
- How to deal with 'tie breaking' situations. For example, if we have an algorithm based on the degrees of vertices, what if two or more of them have the same degree: which one should be labeled first?

Edges and Vertices: Optimality criteria (Cont.)

- *Reverse Cuthill McKee* (RCM): aims at minimizing bandwidth.
- *minimum degree* (MMD) or *approximate minimum degree* (AMD): aims at minimizing the expected fill-in.

Edges and Vertices: Optimality criteria (Cont.)

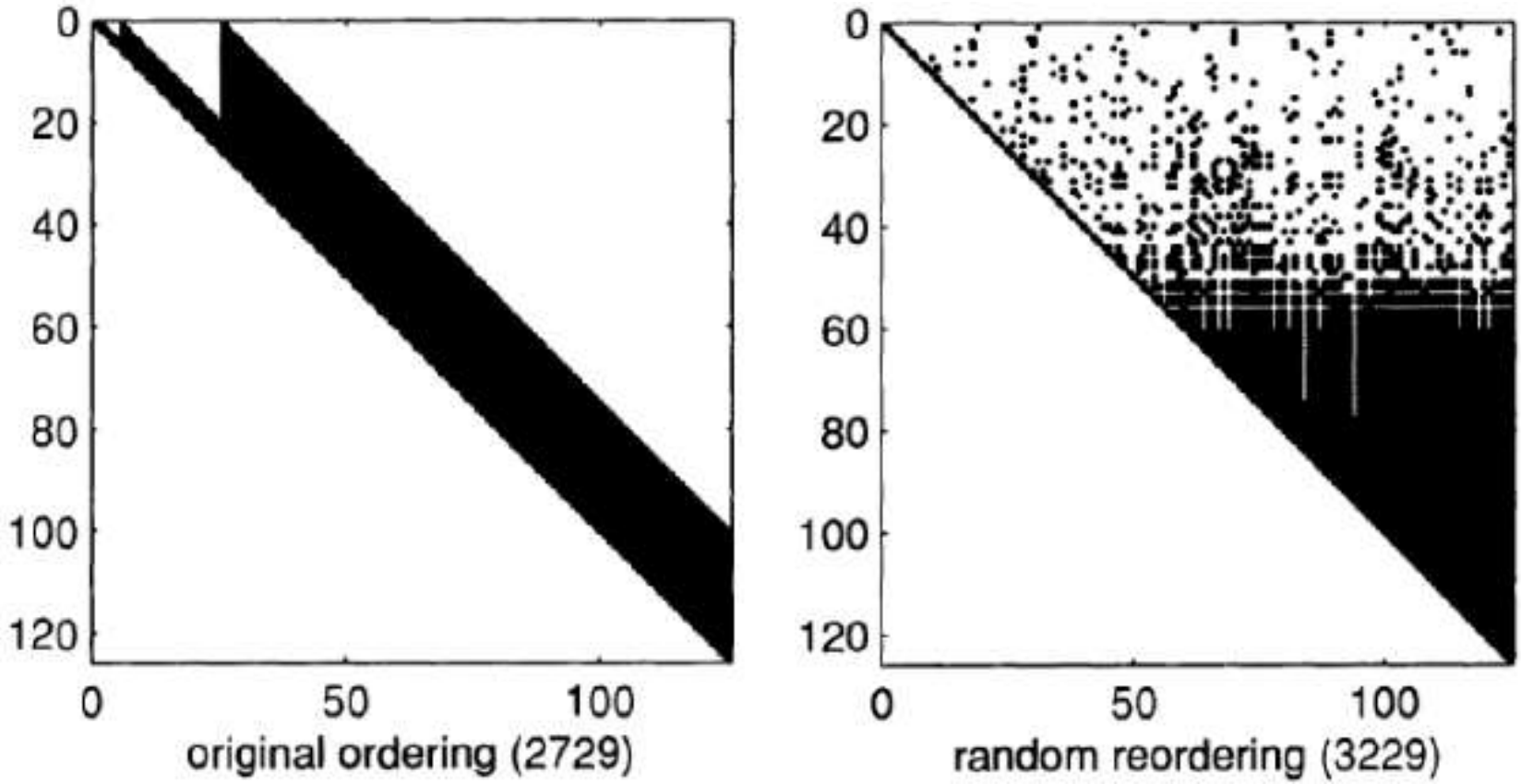


Fig. Spy plots of Cholesky factors of reorderings of a discrete Laplacian matrix. For each ordering, the number of nonzeros is given in parentheses: original (2729) $<$ random (3229).

Edges and Vertices: Optimality criteria (Cont.)

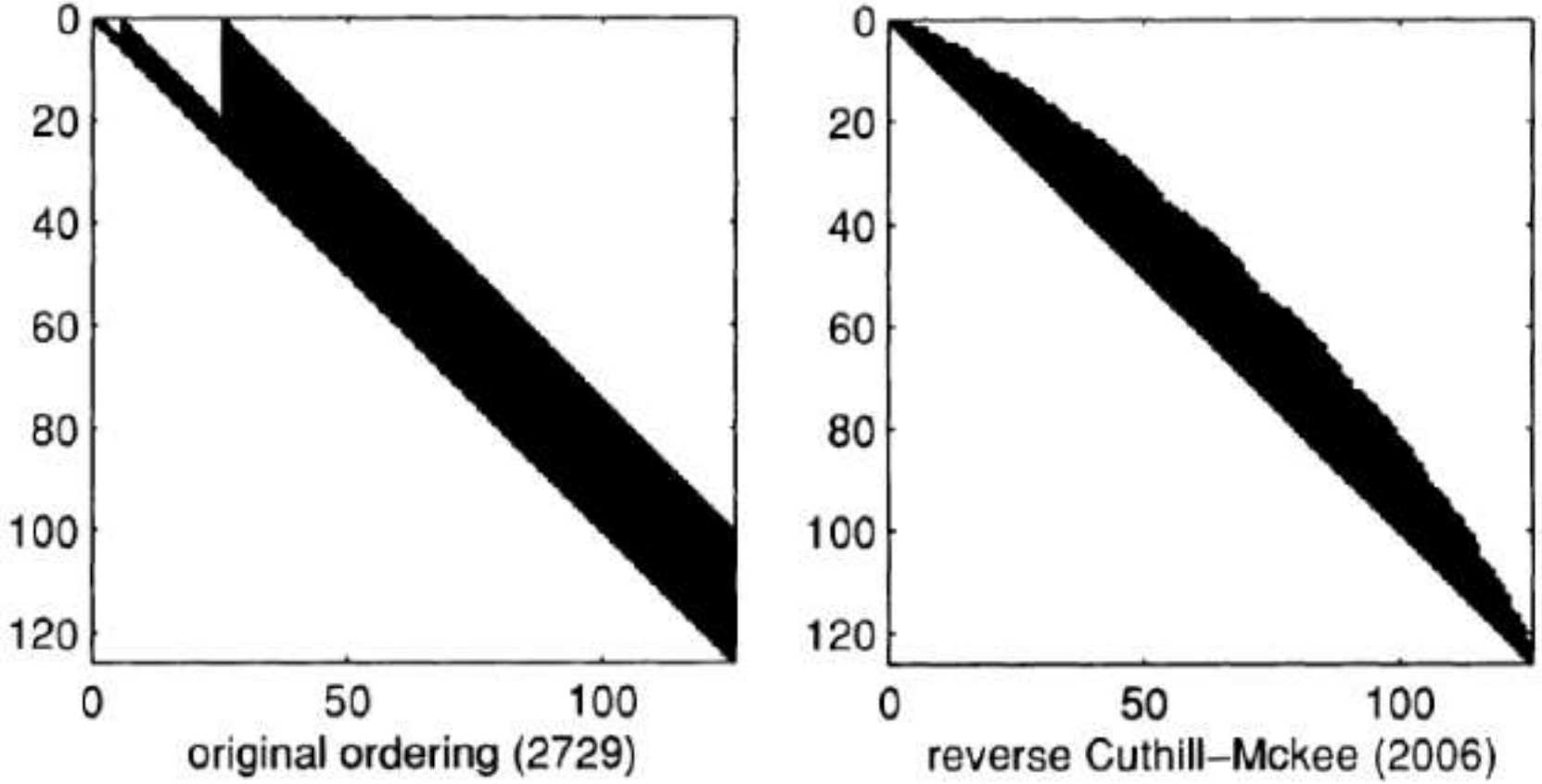


Fig. Spy plots of Cholesky factors of reorderings of a discrete Laplacian matrix. For each ordering, the number of nonzeros is given in parentheses. original (2729) $>$ reverse Cuthill-Mckee (2006).

Edges and Vertices: Optimality criteria (Cont.)

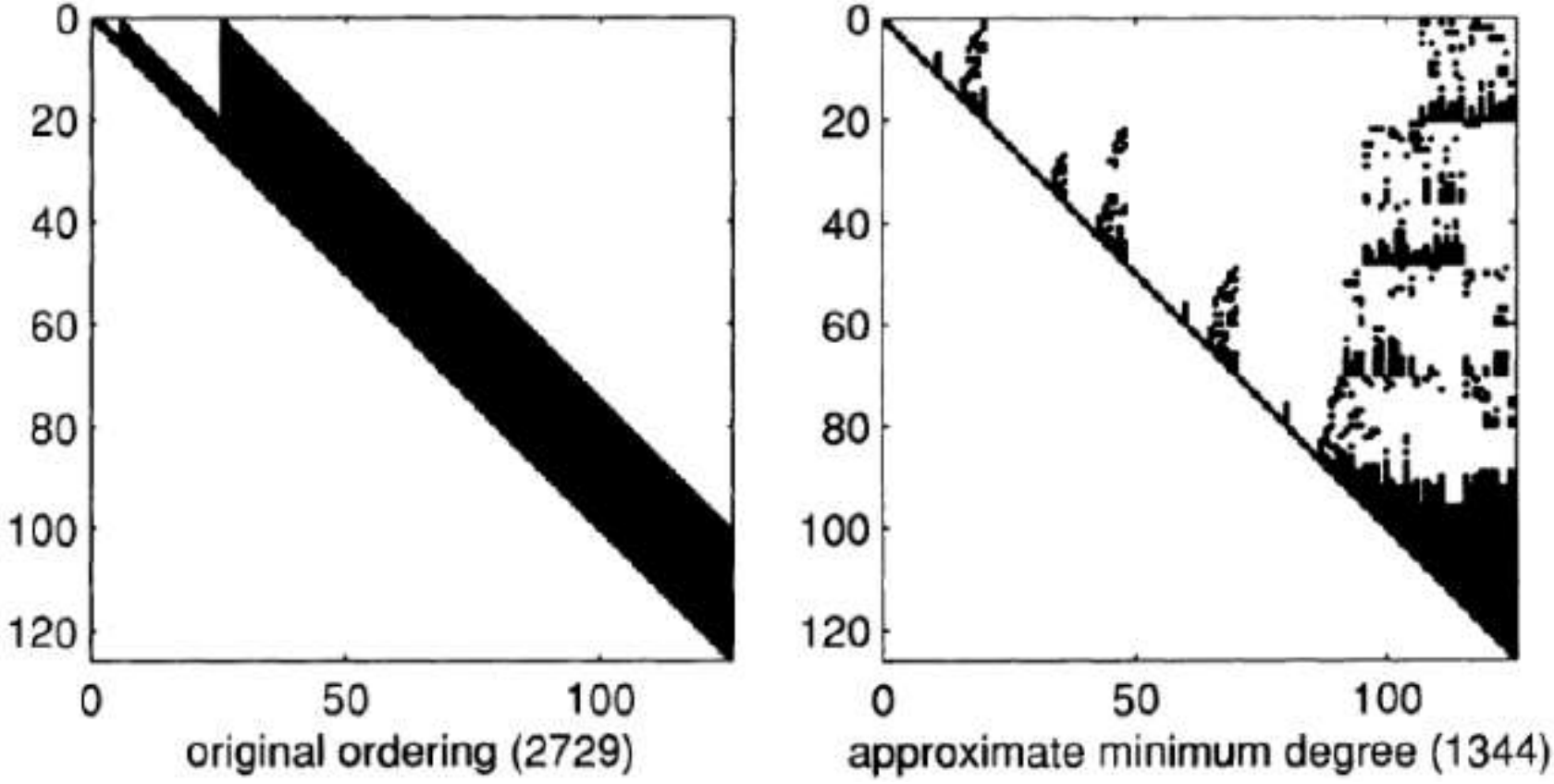


Fig. Spy plots of Cholesky factors of reorderings of a discrete Laplacian matrix. For each ordering, the number of nonzeros is given in parentheses. original (2729) $>$ approximate minimum degree (1344).

Edges and Vertices: Optimality criteria (Cont.)

- One small example proves nothing, but extensive tests on larger matrices have confirmed that the approximate minimum-degree algorithm does significantly better than reverse Cuthill-McKee on a wide variety of problems.

- However, effective exploitation of the good fill properties of the approximate minimum-degree algorithm requires use of a more flexible data structure for sparse matrices since the fill is not restricted to a narrow band.

- In contrast, if we use the reverse Cuthill-McKee algorithm, we can use a simple band or envelope scheme that accommodates the fill automatically.

- **For more details, see Project 1**

- **Conditioning and Accuracy**
  - **Upper bound on the error**
  - **The Condition Number**

Conditioning and Accuracy: Upper bound on the error

Suppose that, using some algorithm, we have computed an approximate solution $\hat{\mathbf{x}}$. We would like to be able to evaluate the absolute error $\|\mathbf{x} - \hat{\mathbf{x}}\|$, or the relative error

$$\frac{\|\mathbf{x} - \hat{\mathbf{x}}\|}{\|\mathbf{x}\|}.$$

- We do not know the error; seek an upper bound, and rely on computable quantities, such as the *residual*

$$\mathbf{r} = \mathbf{b} - A\hat{\mathbf{x}}.$$

- A stable Gaussian elimination variant will deliver a residual with a small norm. The question is, what can we conclude from this about the error in $\mathbf{x}$?

Conditioning and Accuracy: Upper bound on the error

$$\mathbf{r} \quad = \quad \mathbf{b} - A\hat{\mathbf{x}} = A\mathbf{x} - A\hat{\mathbf{x}} = A(\mathbf{x} - \hat{\mathbf{x}}).$$

So

$$\mathbf{x} - \hat{\mathbf{x}} = A^{-1}\mathbf{r}.$$

Then

$$\|\mathbf{x} - \hat{\mathbf{x}}\| = \|A^{-1}\mathbf{r}\| \leq \|A^{-1}\|\|\mathbf{r}\|.$$

This gives a bound on the absolute error in $\hat{\mathbf{x}}$ in terms of $\|A^{-1}\|$.
But usually the relative error is more meaningful. Since
$\|\mathbf{b}\| \leq \|A\|\|\mathbf{x}\|$ implies $\frac{1}{\|\mathbf{x}\|} \leq \frac{\|A\|}{\|\mathbf{b}\|}$, we have

$$\frac{\|\mathbf{x} - \hat{\mathbf{x}}\|}{\|\mathbf{x}\|} \leq \|A^{-1}\|\|\mathbf{r}\| \frac{\|A\|}{\|\mathbf{b}\|}.$$

Conditioning and Accuracy: Condition number

We therefore define the **condition number** of the matrix A as

$$\kappa(A) = \|A\|\|A^{-1}\|$$

and write the bound obtained on the relative error as

$$\frac{\|\mathbf{x} - \hat{\mathbf{x}}\|}{\|\mathbf{x}\|} \leq \kappa(A)\frac{\|\mathbf{r}\|}{\|\mathbf{b}\|}.$$

In words, the relative error in the solution is bounded by the condition number of the matrix A times the relative error in the residual.

Conditioning and Accuracy: Properties and facts!

- Range of values:

$$1 = \|I\| = \|A^{-1}A\| \leq \kappa(A),$$

  (i.e. a matrix is ideally conditioned if its condition number equals 1), and $\kappa(A) = \infty$ for a singular matrix.
- *Orthogonal matrices* are perfectly conditioned.
- If A is SPD, $\kappa_2(A) = \frac{\lambda_1}{\lambda_n}$.
- Condition number is defined for *any* (even non-square) matrices by the singular values of the matrix.
- When something goes wrong with the numerical solution - blame the condition number! (and hope for the best)
- One of the most important areas of research: *preconditioning*. (To be discussed later.)
- **What's a well-conditioned matrix and what's an ill-conditioned matrix?**

Motivation: The previous facts point out to the need an alternative for solving $Ax = b$. Iterative methods!

OVERVIEW OF ITERATIVE METHODS

- **Iterative Methods**
  - **Motivation**
  - **Basic Stationary Methods**
  - **Nonstationary Methods**
  - **Preconditioning**

Motivation

Drawbacks of Direct Solution Methods

The Gaussian elimination algorithm and its variations such as the LU decomposition, the Cholesky method, adaptation to banded systems, etc., is the approach of choice for many problems. There are situations, however, which require a different treatment.

Drawbacks of Direct Solution Methods (Cont.)

- The Gaussian elimination (or LU decomposition) process may introduce **fill-in**, i.e. L and U may have nonzero elements in locations where the original matrix A has zeros. If the amount of fill-in is significant then applying the direct method may become costly. This in fact occurs often, in particular when the matrix is banded and is sparse within the band.
- Sometimes we do not really need to solve the system exactly. (e.g. nonlinear problems.) Direct methods cannot accomplish this because by definition, to obtain a solution the process must be completed; there is no notion of an early termination or an inexact (yet acceptable) solution.

Drawbacks of Direct Solution Methods (Cont.)

- Sometimes we have a pretty good idea of an approximate guess for the solution. For example, in time dependent problems (*warm start* with previous time solution). Direct methods cannot make good use of such information.
- Sometimes only matrix-vector products are given. In other words, the matrix is not available explicitly or is very expensive to compute. For example, in digital signal processing applications it is often the case that only input and output signals are given, without the transformation itself explicitly formulated and available.

MOTIVATION:

FLUID DYNAMICS IN POROUS MEDIA

PDEs, HYPERBOLIC CONSERVATION LAWS AND BALANCE LAWS

A possible motivation for iterative methods

What motivates us to use iterative schemes is the possibility that inverting A may be very difficult, to the extent that it may be worthwhile to invert a much 'easier' matrix several times, rather than inverting A directly only once.

A Canonical Example: Discrete 2D Laplacian

$$A = \begin{pmatrix} J & -I & & & \\ -I & J & -I & & \\ & \ddots & \ddots & \ddots & \\ & & -I & J & -I \\ & & & -I & J \end{pmatrix},$$

where J is a tridiagonal $N \times N$ matrix

$$J = \begin{pmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ & & & -1 & 4 \end{pmatrix}$$

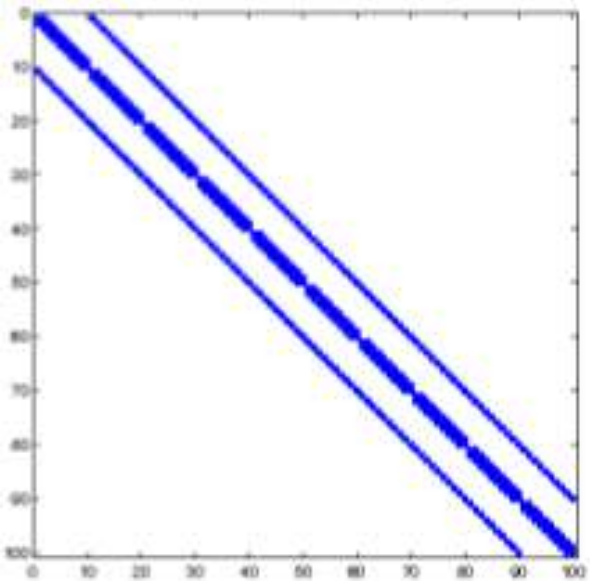
and I denotes the identity matrix of size N.

5-point discretization of the 2D Laplacian (Poisson equation)

A Small Example: Sparsity Pattern

For instance, if $N = 3$ then

$$A = \left(\begin{array}{ccc|ccc|ccc} 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & 0 & 0 & -1 & 0 & 0 & 0 \\ \hline -1 & 0 & 0 & 4 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 4 & 0 & 0 & -1 \\ \hline 0 & 0 & 0 & -1 & 0 & 0 & 4 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 \end{array}\right)$$



5-point discretization of the 2D Laplacian (Poisson equation)

Basic Stationary Methods

Stationary Methods (fixed point iteration)

Given $A\mathbf{x} = \mathbf{b}$, we can rewrite as $\mathbf{x} = (I - A)\mathbf{x} + \mathbf{b}$, which yields the iteration

$$\mathbf{x}_{k+1} = (I - A)\mathbf{x}_k + \mathbf{b}.$$

From this we can generalize: for a given *splitting* $A = M - N$, we have $M\mathbf{x} = N\mathbf{x} + \mathbf{b}$, which leads to the fixed point iteration

$$M\mathbf{x}_{k+1} = N\mathbf{x}_k + \mathbf{b}.$$

Stationary Methods (The Basic Procedure)

Suppose that $A = M - N$ is a splitting, and that $M\mathbf{z} = \mathbf{r}$ is much easier to solve than $A\mathbf{x} = \mathbf{b}$. Given an initial guess $\mathbf{x}_0$,

$$\mathbf{e}_0 = \mathbf{x} - \mathbf{x}_0$$

is the error and

$$A\mathbf{e}_0 = \mathbf{b} - A\mathbf{x}_0 := \mathbf{r}_0.$$

Notice that $\mathbf{r}_0$ is computable whereas $\mathbf{e}_0$ is not, because $\mathbf{x}$ is not available. Since $\mathbf{x} = \mathbf{x}_0 + \mathbf{e}_0 = \mathbf{x}_0 + A^{-1}\mathbf{r}_0$, set

$$M\tilde{\mathbf{e}} = \mathbf{r}_0,$$

and then

$$\mathbf{x}_1 = \mathbf{x}_0 + \tilde{\mathbf{e}}$$

is our new guess. $\mathbf{x}_1$ is hopefully closer to $\mathbf{x}$ than $\mathbf{x}_0$.

Stationary Methods: The hard task of finding a good M

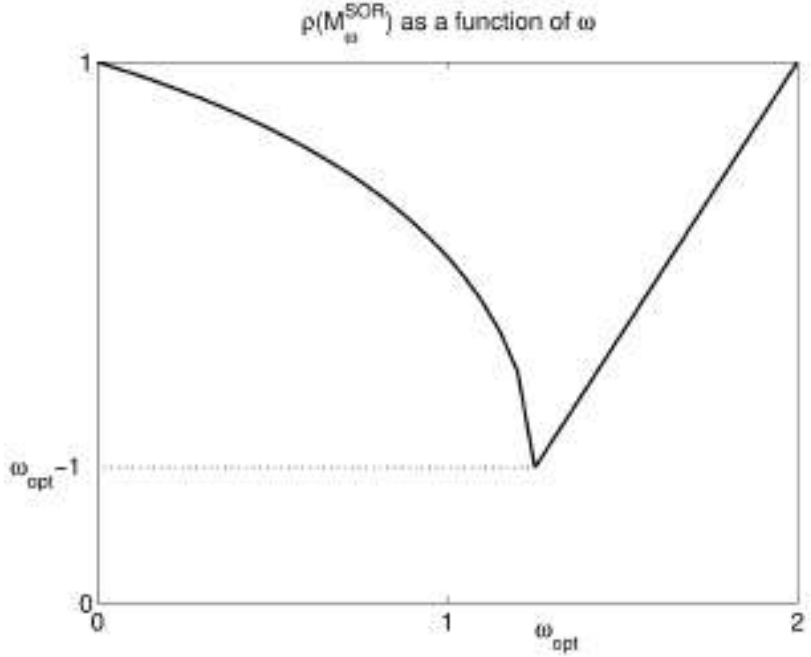
The matrix M should satisfy two contradictory requirements:

- It should be close to A in some sense (or rather, M^{-1} should be close to A^{-1}).
- It should be much easier to invert than A.

Stationary Methods: Jacobi, Gauss-Seidel, SOR and others

It all boils down to the choice of M. If $A = D + E + F$ is split into diagonal D, strictly upper triangular part E and strictly lower triangular part F, then:

- Jacobi: $M = D$.
- Gauss-Seidel: $M = D + E$.
- SOR: a parameter dependent 'improvement' of Gauss-Seidel.



Preliminary Summary: Benefits & Drawbacks

Method	Benefits	Drawbacks
Forward/ backward substitution	Fast (n^2)	Applies only to upper- or lower-triangular matrices
Gaussian elimination	Works for any [non-singular] matrix	$O(n^3)$
LU decomposition	Works for any matrix (singular matrices can still be factored); can re-use L, U for different b values; once factored uses only forward/ backward substitution	$O(n^3)$ initial factorization (same process as Gauss)
Cholesky	$O(n^3)$ but with ½ storage and computation of Gauss	Still $O(n^3)$; only for symmetric positive definite
Band-diagonal elimination	$O(w^2n)$ where w = band width	Only for band diagonal

Exercise: convince yourself about the computational complexity displayed at the columns: benefits and drawbacks !

Preliminary Summary: Benefits & Drawbacks (Cont.)

Method	Benefits	Drawbacks
Sherman-Morrison	Update step is $O(n^2)$	Only for rank-1 changes; degrades with repeated iterations (then use Woodbury instead)
Iterative refinement	Can be applied following any solution method	Requires 2x storage, extra precision for residual
Jacobi	More appropriate than elimination for large/sparse systems; can be parallelized	Can diverge when not diagonally dominant; slow
Gauss-Seidel	More appropriate than elimination for large/sparse; a bit more powerful than Jacobi	Can diverge when not diagonnally dominant or symmetric/positive-definite; slow; can't parallelize
SOR	Potentially faster than Jacobi, Gauss-Seidel for large/sparse systems	Requires parameter tuning
Conjugate gradient	Fast(er) for large/sparse systems; often doesn't require all *n* iterations	Requires symmetric positive definite (otherwise use bi-conjugate)

Nonstationary Methods

Remark: Stationary versus Nonstationary Methods: What is the best (if any) ?

Nonstationary Methods (Krylov subspaces)

The trouble with stationary schemes is that they do not make use of information that has accumulated throughout the iteration. How about trying to optimize something throughout the iteration? For example,

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{r}_k .$$

Adding $\mathbf{b}$ to both sides and subtracting the equations multiplied by A:

$$\mathbf{b} - A\mathbf{x}_{k+1} = \mathbf{b} - A\mathbf{x}_k - \alpha_k A\mathbf{r}_k .$$

It is possible to find α_k that minimizes the residual. Notice:

$$\mathbf{r}_k = p_k(A)\mathbf{r}_0 .$$

Modern method work hard at finding 'the best' p_k.

Remark1: Unlike stationary methods, nonstationary methods do not have an iteration matrix !

Remark2: Looking more closely at this error (residual) r_k, we see that $p_k(A)$ is a k^{th} degree of a polynomial matrix, in which we wish to have small eigenvalues (less than 1 in magnitude).

Nonstationary Methods as an Optimization Problem

Nonstationary methods as an optimization problem

The methods considered here can all be written as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k,$$

where the vector $\mathbf{p}_k$ is the *search direction* and the scalar α_k is the *step size*. The simplest such non-stationary scheme is obtained by setting $\mathbf{p}_k = \mathbf{r}_k$, i.e. $M_k = \alpha_k I$, with I the identity matrix. The resulting method is called **gradient descent**.
The step size α_k may be chosen so as to minimize the ℓ_2 norm of the residual $\mathbf{r}_k$. But there are other options too.

Conjugate Gradients (for SPD matrices)

Our problem $A\mathbf{x} = \mathbf{b}$ is equivalent to the problem of finding a vector $\mathbf{x}$ that minimizes

$$\phi(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T A\mathbf{x} - \mathbf{b}^T\mathbf{x}.$$

The Conjugate Gradient Method defines search directions that are A-conjugate, and minimizes $\|\mathbf{e}_k\|_A = \sqrt{\mathbf{e}_k^T A\mathbf{e}_k}$. Note that this is well defined only if A is SPD.

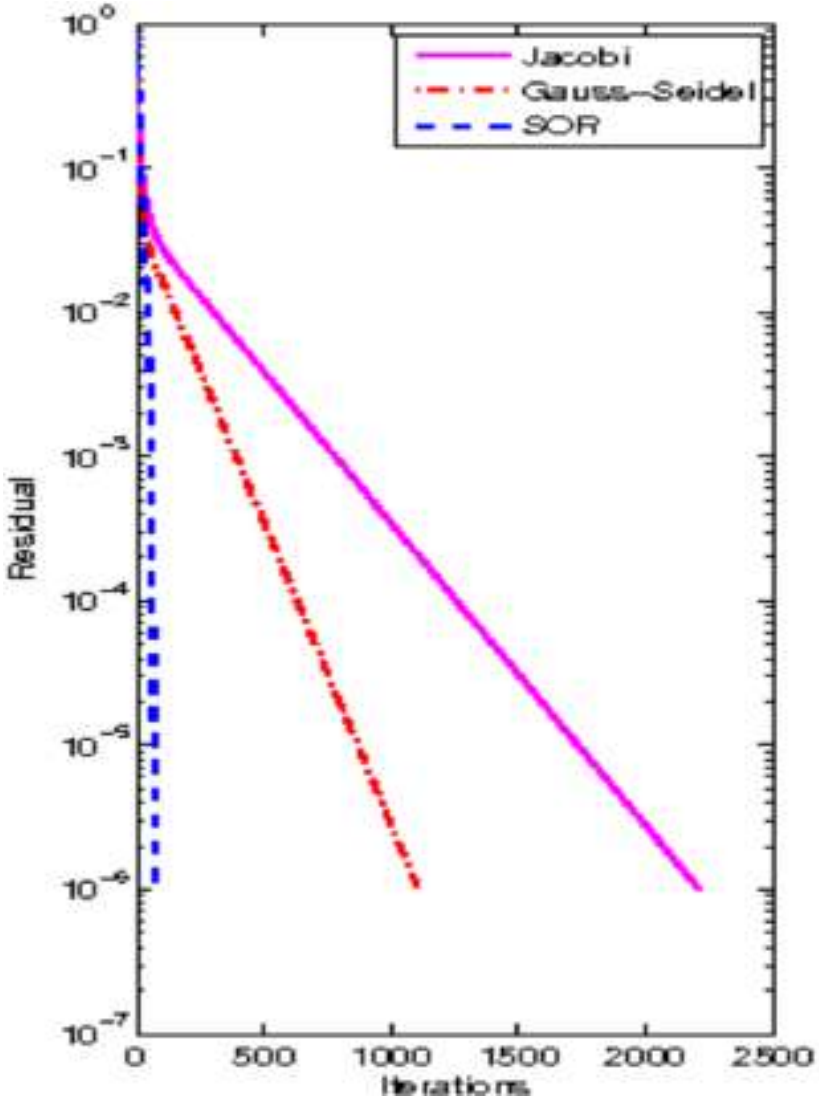
Remark: SPD matrices = **S**ymmetric **P**ositive **D**efinite matrices

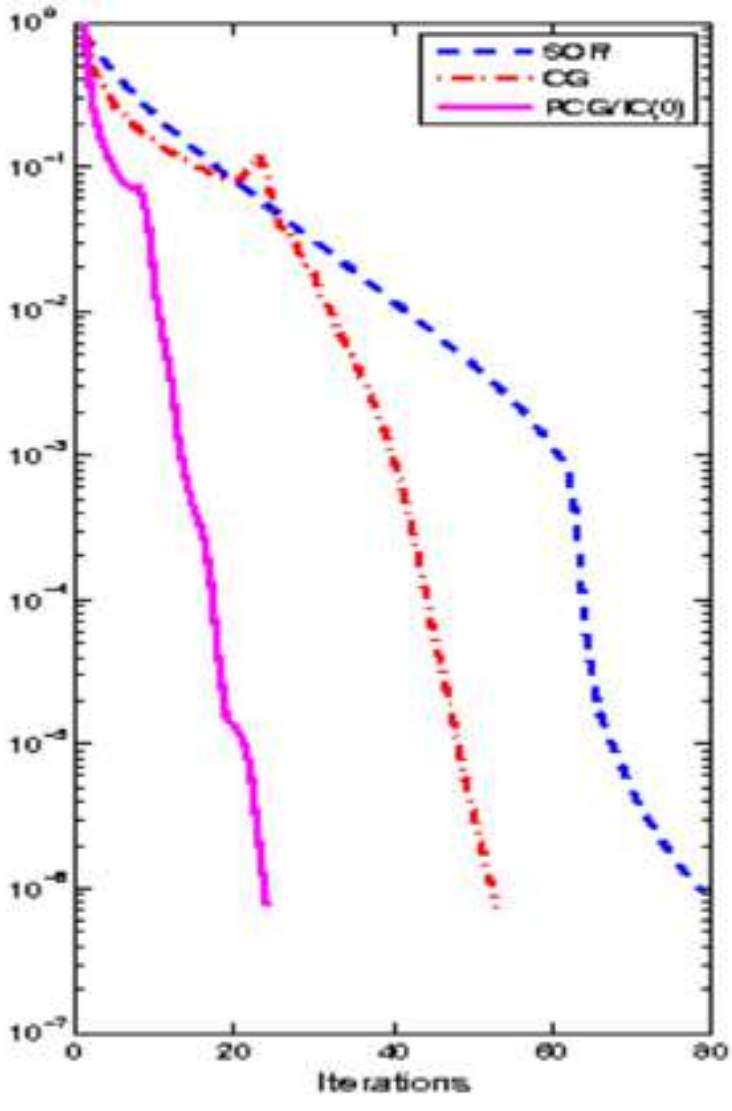
A (first) look at the Conjugate Gradient (CG) algorithm

Given an initial guess $\mathbf{x}_0$ and a tolerance tol, set at first $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$, $\delta_0 = \langle \mathbf{r}_0, \mathbf{r}_0 \rangle$, $b_\delta = \langle \mathbf{b}, \mathbf{b} \rangle$, $k = 0$ and $\mathbf{p}_0 = \mathbf{r}_0$. Then: While $\delta_k > tol^2 \; b_\delta$,

$$
\begin{aligned}
\mathbf{s}_k &= A\mathbf{p}_k \\
\alpha_k &= \frac{\delta_k}{\langle \mathbf{p}_k, \mathbf{s}_k \rangle} \\
\mathbf{x}_{k+1} &= \mathbf{x}_k + \alpha_k \mathbf{p}_k \\
\mathbf{r}_{k+1} &= \mathbf{r}_k - \alpha_k \mathbf{s}_k \\
\delta_{k+1} &= \langle \mathbf{r}_{k+1}, \mathbf{r}_{k+1} \rangle \\
\mathbf{p}_{k+1} &= \mathbf{r}_{k+1} + \frac{\delta_{k+1}}{\delta_k} \mathbf{p}_k \\
k &= k + 1.
\end{aligned}
$$

Numerical convergence behaviour: stationary or nonstationary methods





A link between the Conjugate Gradient method and Krylov subspace methods

- Y. Saad, *Krylov Subspace Methods for Solving Large Unsymmetric Linear Systems*, Mathematics of Computation 37, 105-126 (1981).

- The purpose of the paper of Y. Saad (1981) is to generalize the conjugate gradient method regarded as a projection process onto the **Krylov subspace** $\mathcal{K}^k$.

- We shall say of a method realizing such a process that it belongs to the class of Krylov subspace methods.

- Indeed, it will be seen that these Krylov subspace methods can be efficient for solving large nonsymmetric systems.

Krylov Subspace Methods in a first glance

We are seeking to find a solution within the Krylov subspace

$$\mathbf{x}_k \in \mathbf{x}_0 + \mathcal{K}^k(A; \mathbf{r}_0) \equiv \mathbf{x}_0 + span\{\mathbf{r}_0, A\mathbf{r}_0, A^2\mathbf{r}_0, \ldots, A^{k-1}\mathbf{r}_0\}.$$

- Find a good basis for the space (riddle: 'good' means what??): Lanczos or Arnoldi will help here.
- Optimality condition:
  - Require that the norm of the residual $\|b - Ax_k\|_2$ is minimal over the Krylov subspace.
  - Require that the residual is orthogonal to the subspace.

- C. Lanczos, An iteration method for the solution of the eigenvalue problem of linear differential and integral operators, J. Res. Nat'l Bur. Std. 45 (1950) 255-282.
- W. E. Arnoldi, The principle of minimized iterations in the solution of the matrix eigenvalue problem, Quarterly of Applied Mathematics, 9 (1951) 17-29.

Well known methods Krylov subspace methods

1) Conjugate Gradient method (CG method)
2) Biconjugate Gradient method (BiCG method)
3) Biconjugate Gradient Stabilized (Bi-CGSTAB method)
4) Minimal Residual (MINRES method)
5) General minimal Residual method (GMRES method)
6) Symmetric LQ method(SYMMLQ method)
7) Conjugate Gradient Squared (CGS method)
8) Quasi-Minimal Residual (QMR method)
9) Conjugate Gradients on the Normal Equations (CGNE and CGNR methods)

Remark: We will discuss in more details the Krylov methods: CG, Bi-CGSTAB, MINRES and GMRES.

Well known methods Krylov subspace methods

1) **Conjugate Gradient method (CG method)**

   Magnus R. Hestenes and Eduard Stiefel. Methods of Conjugate Gradients for Solving Linear Systems. Journal of Research of the National Bureau of Standards, 49(6) (**1952**) 409–436.

   Obs.: The matrix A ($Ax = b$) is SPD.

2) **Minimal Residual (MINRES method)**

   Chris Paige Michael Saunders. Solutions of sparse indefinite systems of linear equations, SIAM J. Numer. Anal 12 (**1975**) 617–629.

3) **General minimal Residual method (GMRES method)**

   Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput., 7 (**1986**) 856–869.

4) **Biconjugate Gradient Stabilized (Bi-CGSTAB method)**

   H. A. Van der Vorst. Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems. SIAM J. Sci. and Stat. Comput. 13(2) (**1992**) 631–644.

Preconditioning

Preconditioning

Convergence rate typically depends on two factors:

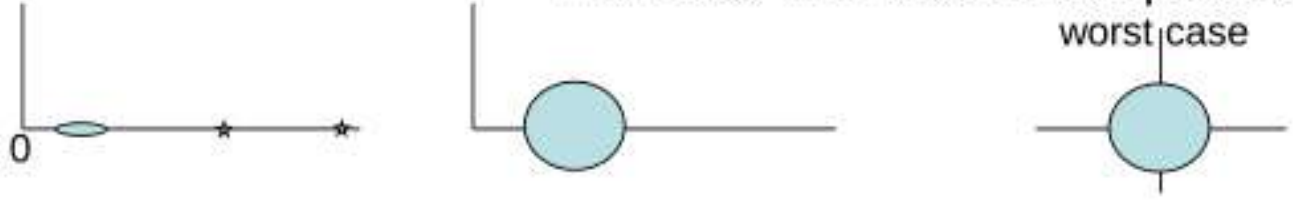
- Distribution/clustering of eigenvalues **(crucial!)**
- Condition number (the **less** important factor!)

Idea: Since A is given and is beyond our control, define a matrix M such that the above properties are better for $M^{-1}A$, and solve $M^{-1}Ax = M^{-1}b$ rather than $Ax = b$.

Requirements: To produce an effective method the preconditioner matrix M must be easily invertible. At the same time it is desirable to have at least one of the following properties hold: $\kappa(M^{-1}A) \ll \kappa(A)$, and/or the eigenvalues of $M^{-1}A$ are much better clustered compared to those of A.

Essential for convergence: position of eigenvalues (singular values)

Fast convergence for: well conditioned problems or matrices with clustered spectrum



Preconditioning (Cont.)

- Algebraic, general purpose (arguably, frequently needed in continuous optimization)
- Specific to the problem (arguably, frequently needed in PDEs)

Remark: We will overview steady-state and dynamic models involving PDEs for a boundary problem and a boundary-initial boundary problem.

Preconditioning (Cont.)

Preconditioning is a combination of art and science...

- Stationary preconditioners, such as Jacobi, Gauss-Seidel, SOR.
- Incomplete factorizations.
- Multigrid and multilevel preconditioners. (Advanced)
- Preconditioners tailored to the problem in hand, that rely for example on the properties of the underlying differential operators.

Remark: Incomplete Factorizations might also be considered as follows: Given the matrix A, construct an LU decomposition or a Cholesky decomposition (if A is symmetric positive definite) that follows precisely the same steps as the usual decomposition algorithms, except that a nonzero entry of a factor is generated only if the matching entry of A is nonzero.

Some basic concepts of consistency, stability and convergence of a numerical method

Topics/prelude

Well-posedness and ill-posedness

Conditioning, stability and sources of error

Forward and backward stability analysis

A priori and a posteriori analysis

Relations between stability and convergence

More on conditioning, stability and sources of error

Well-posedness and ill-posedness

The concept of a well-posed (correct) problem of mathematical physics was formulated by the famous French mathematician Hadamard (1902): Jacques Hadamard (1902). Sur les problèmes aux dérivées partielles et leur signification physique. Princeton University Bulletin. pp. 49-52.

At the present time this concept is widely presented in textbooks on the equations of mathematical physics or partial differential equations (see, e.g., Lawrence C. Evans, PDE book, AMS 1988).

A problem of mathematical physics or a boundary value problem for a partial differential equation is called well-posed if the following conditions are satisfied:

1) a solution of the problem exists;
2) the solution of the problem is unique; and
3) the solution of the problem depends continuously on the data of the problem.

More on conditioning, stability and sources of error

Well-posedness and ill-posedness

The well-posedness conditions just formulated require refinement.

Namely, both **the solution** and **the data** of the problem are considered as elements of some function space, and the conditions for a problem to be well-posed are formulated as follows.

I) A solution of the problem exists for all data belonging to some closed subspace in a normed linear space of the type C^k, L_p, H_p^ℓ, W_p^ℓ, etc... and belongs to a space of the same type. The subspace is most often either the entire space or a part of the space on which a finite collection of linear functionals vanishes.

II) The solution of the problem is unique in some analogous space.

III) To infinitesimal variations of the data of the problem in the data space there correspond infinitesimal variations of the solution in the solution space

Remark: Problems that are not well-posed in the sense of Hadamard are termed **ill-posed**.

More on conditioning, stability and sources of error

Well-posedness and ill-posedness

Having formulated the concept of a well-posed problem, Hadamard presented an example of an ill-posed problem for a differential equation which in his opinion did not correspond to any real physical formulation.

The Cauchy problem for the Laplace equation that is ill-posed (or not well-posed) in the sense of Hadamard, since the solution does not continuously depend on the data of the problem. Such ill-posed problems are not usually satisfactory for physical applications!

Typical examples of well-posed (correct) problems of mathematical physics include the Dirichlet problem for Laplace's equation and the heat equation with specified initial conditions.

For instance, the backwards heat equation, deducing a previous distribution of temperature from final data, is not well-posed in the sense of Hadamard, in that the solution is highly sensitive to changes in the final data, *see, e.g., James V. Beck, Ben Blackwell, Charles R. St. Clair, Jr, Inverse heat conduction: ill-posed problems, NY: John Wiley (1985) – see BAE library*.

More on conditioning, stability and sources of error

Well-posedness and ill-posedness

Continuum models (differential or not) must often be discretized in order to obtain a numerical solution (e.g., in the form $Ax = b$). While solutions may be continuous with respect to the initial conditions, they may suffer from numerical instability when solved with finite precision.

Even if a problem is well-posed, it may still be ill-conditioned, meaning that a small error in the initial data can result in much larger errors in the answers. If the problem is well-posed, then it stands a good chance of solution on a computer using a stable algorithm.

An ill-conditioned problem is indicated by a large condition number.

If it is not well-posed, it needs to be re-formulated for numerical treatment. Typically this involves including additional assumptions, such as smoothness of solution. This process is known as regularization.

Tikhonov regularization is one of the most commonly used for regularization of linear ill-posed problems. (But this is not the subject of this course.)

Now, consider the words of Baxter and Iserles in *B.J.C. Baxter & A. Iserles. "On the foundations of computational mathematics", in Handbook of Numerical Analysis XI (P.G. Ciarlet & F. Cucker, eds), North-Holland, Amsterdam (2003), 3-34.*

"It is a sobering thought that, even when a computational solution to a mathematical problem has been found, often following great intellectual and computational effort, its merit might be devalued by poor stability of the underlying algorithm."

"This state of affairs is sometimes designated as *stability* or *well posedness* or *conditioning* – purists may argue *ad nauseam* over the precise definitions of these concepts, but it is clear that, one way or the other, they play an instrumental role in computational mathematics."

"Another dichotomy, extant in both *computational analysis* and *computational algebra*, is between traditional *forward stability analysis* and the approach of *backward error analysis* (a misnomer: in reality it refers to *backward stability* or *conditioning analysis*) with respect to the numerical method under consideration."

More on conditioning, stability and sources of error

We shall find the terms well posed and stable being used in an interchanging manner. In general, the concept of well-posedness is linked to the original continuum (differential) model and the concept of stability it closely related to conditioning of the underlying model.

For example, a linear system $Ax = b$ of n equations in n unknowns with a nonsingular coefficient matrix A has exactly one solution. Even so, if A is nearly singular then a small perturbation of A can produce a large change in the solution, although not arbitrarily large: the condition number $||A||\ ||A^{-1}||$ bounds the relative change.

Caution: Indeed, it is not appropriate to pretend the numerical method can cure the pathologies of an intrinsically ill-posed problem.

Good references on this subject for more details and rigorous proofs of these facts (all available in our bibimecc):

[1] K. Atkinson. Theoretical numerical analysis: a functional analysis framework, 3rd ed (2010).

[2] Gene H. Golub and Charles F. Van Loan. Matrix computations, 3rd ed., Baltimore, MD; London: Johns Hopkins University Press (1996).

[3] V. A. Morozov ; translation editor Z. Nashed, translated by A. B. Aries. Methods for solving incorrectly posed problems. Springe (1984).

More on conditioning, stability and sources of error

Consider the following problem: find x such that

$$F(x, d) = 0 \tag{1.1}$$

where d is the set of data which the solution depends on and F is the functional relation between x and d. According to the kind of problem that is represented in (1.1), the variables x and d may be real numbers, vectors or functions. Typically, (1.1) is called a *direct* problem if F and d are given and x is the unknown, *inverse* problem if F and x are known and d is the unknown, *identification* problem when x and d are given while the functional relation F is the unknown.

Problem (1.1) is *well posed* if it admits a *unique* solution x which *depends with continuity on the data.*

A problem which does not enjoy the property above is called *ill posed* or *unstable*.

Example A simple instance of an ill-posed problem is finding the number of real roots of a polynomial. For example, the polynomial $p(x) = x^4 - x^2(2a - 1) + a(a - 1)$ exhibits a discontinuous variation of the number of real roots as a continuously varies in the real field. We have, indeed, 4 real roots if $a \geq 1$, 2 if $a \in [0, 1)$ while no real roots exist if $a < 0$.

More on conditioning, stability and sources of error

Well-posedness and Condition Number of a Problem

Continuous dependence on the data means that small perturbations on the data d yield "small" changes in the solution x. Precisely, denoting by δd an admissible perturbation on the data and by δx the consequent change in the solution, in such a way that

$$F(x + \delta x, d + \delta d) = 0, \tag{1.2}$$

then

$$\forall \eta > 0,\ \exists K(\eta, d) :\ \|\delta d\| < \eta \ \Rightarrow\ \|\delta x\| \leq K(\eta, d)\|\delta d\|. \tag{1.3}$$

The norms used for the data and for the solution may not coincide, whenever d and x represent variables of different kinds.
With the aim of making this analysis more quantitative, we introduce the following definition.

More on conditioning, stability and sources of error

Well-conditioned and ill-conditioned depends on the context of the problem

Definition 1.1 For problem (1.1) we define the *relative condition number* to be

$$K(d) = \sup_{\delta d \in D} \frac{\|\delta x\|/\|x\|}{\|\delta d\|/\|d\|}, \tag{1.4}$$

where D is a neighborhood of the origin and denotes the set of admissible perturbations on the data for which the perturbed problem (2.2) still makes sense. Whenever $d = 0$ or $x = 0$, it is necessary to introduce the *absolute condition number*, given by

$$K_{abs}(d) = \sup_{\delta d \in D} \frac{\|\delta x\|}{\|\delta d\|}. \tag{1.5}$$

Problem (1.1) is called *ill-conditioned* if $K(d)$ is "big" for any admissible datum d (the precise meaning of "small" and "big" is going to change depending on the considered problem).

More on conditioning, stability and sources of error

Well-posedness and Ill-posed problems

The property of a problem of being well-conditioned is independent of the numerical method that is being used to solve it. In fact, it is possible to generate stable as well as unstable numerical schemes for solving well-conditioned problems. The concept of stability for an algorithm or for a numerical method is analogous to that used for problem (1.1) and will be made precise in the next section.

Remark (Ill-posed problems) Even in the case in which the condition number does not exist (formally, it is infinite), it is not necessarily true that the problem is ill-posed. In fact there exist well posed problems (for instance, the search of multiple roots of algebraic equations, see next Ex.) that for which the condition number is infinite, but such that they can be reformulated in equivalent problems (that is, having the same solutions) with a finite condition number.

More on conditioning, stability and sources of error

Well-conditioned and ill-conditioned depends on the context of the problem

If problem (1.1) admits a unique solution, then there necessarily exists a mapping G, that we call *resolvent*, between the sets of the data and of the solutions, such that

$$x = G(d), \quad \text{that is} \quad F(G(d), d) = 0. \tag{1.6}$$

According to this definition, (1.2) yields $x + \delta x = G(d + \delta d)$. Assuming that G is differentiable in d and denoting formally by $G'(d)$ its derivative with respect to d (if $G : \mathbb{R}^n \to \mathbb{R}^m$, $G'(d)$ will be the Jacobian matrix of G evaluated at the vector d), a Taylor's expansion of G truncated at first order ensures that

$$G(d + \delta d) - G(d) = G'(d)\delta d + o(\|\delta d\|) \qquad \text{for } \delta d \to 0,$$

where $\|\cdot\|$ is a suitable norm for δd and $o(\cdot)$ is the classical infinitesimal symbol denoting an infinitesimal term of higher order with respect to its argument. Neglecting the infinitesimal of higher order with respect to $\|\delta d\|$, from (1.4) and (1.5) we respectively deduce that

$$K(d) \simeq \|G'(d)\| \frac{\|d\|}{\|G(d)\|}, \qquad K_{abs}(d) \simeq \|G'(d)\|, \tag{1.7}$$

the symbol $\|\cdot\|$ denoting the matrix norm associated with the vector norm

The estimates in (1.7) are of great practical usefulness in the analysis of problems in the form (1.6), as shown in the forthcoming examples.

More on conditioning, stability and sources of error

Well-conditioned and ill-conditioned depends on the context of the problem

Example (Algebraic equations of second degree) The solutions to the algebraic equation $x^2 - 2px + 1 = 0$, with $p \geq 1$, are $x_\pm = p \pm \sqrt{p^2 - 1}$. In this case, $F(x, p) = x^2 - 2px + 1$, the datum d is the coefficient p, while x is the vector of components $\{x_+, x_-\}$. As for the condition number, we notice that (1.6) holds by taking $G : \mathbb{R} \to \mathbb{R}^2$, $G(p) = \{x_+, x_-\}$. Letting $G_\pm(p) = x_\pm$, it follows that $G'_\pm(p) = 1 \pm p/\sqrt{p^2 - 1}$. Using (1.7) with $\|\cdot\| = \|\cdot\|_2$ we get

$$K(p) \simeq \frac{|p|}{\sqrt{p^2 - 1}}, \qquad p > 1. \tag{1.8}$$

From (1.8) it turns out that in the case of separated roots (say, if $p \geq \sqrt{2}$) problem $F(x, p) = 0$ is well conditioned. The behavior dramatically changes in the case of multiple roots, that is when $p = 1$. First of all, one notices that the function $G_\pm(p) = p \pm \sqrt{p^2 - 1}$ is no longer differentiable for $p = 1$, which makes (1.8) meaningless. On the other hand, equation (1.8) shows that, for p close to 1, the problem at hand is *ill conditioned*. However, the problem is not *ill posed*.

Indeed, it is possible to reformulate it in an equivalent manner as $F(x, t) = x^2 - ((1 + t^2)/t)x + 1 = 0$, with $t = p + \sqrt{p^2 - 1}$, whose roots $x_- = t$ and $x_+ = 1/t$ coincide for $t = 1$. The change of parameter thus removes the singularity that is present in the former representation of the roots as functions of p. The two roots $x_- = x_-(t)$ and $x_+ = x_+(t)$ are now indeed regular functions of t in the neighborhood of $t = 1$ and evaluating the condition number by (1.7) yields $K(t) \simeq 1$ for any value of t. The transformed problem is thus well conditioned.

More on conditioning, stability and sources of error

Well-conditioned and ill-conditioned depends on the context of the problem

Example (Systems of linear equations) Consider the linear system $\mathrm{A}\mathbf{x} = \mathbf{b}$, where $\mathbf{x}$ and $\mathbf{b}$ are two vectors in $\mathbb{R}^n$, while A is the matrix $(n \times n)$ of the real coefficients of the system. Suppose that A is nonsingular; in such a case x is the unknown solution $\mathbf{x}$, while the data d are the right-hand side $\mathbf{b}$ and the matrix A, that is, $d = \{b_i,\ a_{ij}, 1 \leq i, j \leq n\}$.

Suppose now that we perturb only the right-hand side $\mathbf{b}$. We have $d = \mathbf{b}$, $\mathbf{x} = G(\mathbf{b}) = \mathrm{A}^{-1}\mathbf{b}$ so that, $G'(\mathbf{b}) = \mathrm{A}^{-1}$, and (1.7) yields

$$K(d) \simeq \frac{\|\mathrm{A}^{-1}\|\ \|\mathbf{b}\|}{\|\mathrm{A}^{-1}\mathbf{b}\|} = \frac{\|\mathrm{A}\mathbf{x}\|}{\|\mathbf{x}\|}\|\mathrm{A}^{-1}\| \leq \|\mathrm{A}\|\ \|\mathrm{A}^{-1}\| = K(\mathrm{A}), \qquad (1.9)$$

where $K(\mathrm{A})$ is the condition number of matrix A (see definition) and the use of a consistent matrix norm is understood. Therefore, if A is well conditioned, solving the linear system $\mathrm{A}\mathbf{x}=\mathbf{b}$ is a stable problem with respect to perturbations of the right-hand side $\mathbf{b}$. Stability with respect to perturbations on the entries of A will be analyzed in the next pages.

More on conditioning, stability and sources of error

Well-conditioned and ill-conditioned depends on the context of the problem: example

Example (Nonlinear equations) Let $f : \mathbb{R} \to \mathbb{R}$ be a function of class C^1 and consider the nonlinear equation

$$F(x, d) = f(x) = \varphi(x) - d = 0,$$

where $\varphi : \mathbb{R} \to \mathbb{R}$ is a suitable function and $d \in \mathbb{R}$ a datum (possibly equal to zero). The problem is well defined only if φ is invertible in a neighborhood of d: in such a case, indeed, $x = \varphi^{-1}(d)$ and the resolvent is $G = \varphi^{-1}$. Since $(\varphi^{-1})'(d) = [\varphi'(x)]^{-1}$, the first relation in (1.7) yields, for $d \neq 0$,

$$K(d) \simeq \frac{|d|}{|x|} |[\varphi'(x)]^{-1}|, \tag{1.10}$$

while if $d = 0$ or $x = 0$ we have

$$K_{abs}(d) \simeq |[\varphi'(x)]^{-1}|. \tag{1.11}$$

The problem is thus ill posed if x is a multiple root of $\varphi(x) - d$; it is ill conditioned when $\varphi'(x)$ is "small", well conditioned when $\varphi'(x)$ is "large".

Stability of Numerical Methods

Consistency, convergence and stability issues

We shall henceforth suppose the problem (1.1) to be well posed. A numerical method for the approximate solution of (1.1) will consist, in general, of a sequence of approximate problems

$$F_n(x_n, d_n) = 0 \qquad n \geq 1 \tag{1.12}$$

depending on a certain parameter n (to be defined case by case). The understood expectation is that $x_n \to x$ as $n \to \infty$, i.e. that the numerical solution converges to the exact solution. For that, it is necessary that $d_n \to d$ and that F_n "approximates" F, as $n \to \infty$.

Stability of Numerical Methods

Consistency, convergence and stability issues

Consistency

Precisely, if the datum d of problem (1.1) is admissible for F_n, we say that (1.12) is *consistent* if

$$F_n(x,d) = F_n(x,d) - F(x,d) \to 0 \quad \text{for } n \to \infty \tag{1.13}$$

where x is the solution to problem (1.1) corresponding to the datum d.

Remark: (Consistency) The meaning of this definition depends on the underlying single class of the considered problems at hand (e.g., initial value problem for ODEs and initial and boundary value problems for PDEs).

Stability of Numerical Methods

Consistency, convergence and stability issues

Strongly consistency (*a good dream*)

A method is said to be *strongly consistent* if $F_n(x, d) = 0$ for *any* value of n and not only for $n \to \infty$.

In some cases (e.g., when iterative methods are used) problem (1.12) could take the following form

$$F_n(x_n, x_{n-1}, \ldots, x_{n-q}, d_n) = 0 \qquad n \geq q \tag{1.14}$$

where $x_0, x_1, \ldots, x_{q-1}$ are given. In such a case, the property of strong consistency becomes $F_n(x, x, \ldots, x, d) = 0$ for all $n \geq q$.

Stability of Numerical Methods

Consistency, convergence and stability issues (example)

Example Let us consider the following iterative method (known as Newton's method and discussed as in elsewhere.) for approximating a simple root α of a function $f : \mathbb{R} \to \mathbb{R}$,

$$\text{given } x_0, \quad x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}, \qquad n \geq 1. \tag{1.15}$$

The method (1.15) can be written in the form (1.14) by setting $F_n(x_n, x_{n-1}, f) = x_n - x_{n-1} + f(x_{n-1})/f'(x_{n-1})$ and is strongly consistent since $F_n(\alpha, \alpha, f) = 0$ for all $n \geq 1$.

Consider now the following numerical method (known as the composite midpoint rule discussed in elsewhere) for approximating $x = \int_a^b f(t)\, dt$,

$$x_n = H \sum_{k=1}^{n} f\left(\frac{t_k + t_{k+1}}{2}\right), \qquad n \geq 1$$

where $H = (b-a)/n$ and $t_k = a + (k-1)H$, $k = 1, \ldots, (n+1)$. This method is consistent; it is also strongly consistent provided thet f is a piecewise linear polynomial.

More generally, all numerical methods obtained from the mathematical problem by truncation of limit operations (such as integrals, derivatives, series, ...) are *not* strongly consistent.

Stability, consistency and convergence issues

Well posed (or stable), uniqueness and continuity w.r.t initial datum

Recalling what has been previously stated about problem (1.1), in order for the numerical method to be *well posed* (or *stable*) we require that for any fixed n, there exists a unique solution x_n corresponding to the datum d_n, that the computation of x_n as a function of d_n is unique and, furthermore, that x_n depends continuously on the data, i.e.

$$\forall \eta > 0,\ \exists K_n(\eta, d_n):\ \|\delta d_n\| < \eta \ \Rightarrow\ \|\delta x_n\| \leq K_n(\eta, d_n)\|\delta d_n\|. \quad (1.16)$$

As done in (1.4), we introduce for each problem in the sequence (1.12) the quantities

$$K_n(d_n) = \sup_{\delta d_n \in D_n} \frac{\|\delta x_n\|/\|x_n\|}{\|\delta d_n\|/\|d_n\|}, \quad K_{abs,n}(d_n) = \sup_{\delta d_n \in D_n} \frac{\|\delta x_n\|}{\|\delta d_n\|}, \quad (1.17)$$

and then define

$$K^{num}(d_n) = \lim_{k\to\infty} \sup_{n\geq k} K_n(d_n), \quad K^{num}_{abs}(d_n) = \lim_{k\to\infty} \sup_{n\geq k} K_{abs,n}(d_n).$$

We call $K^{num}(d_n)$ the *relative asymptotic condition number* of the numerical method (1.12) and $K^{num}_{abs}(d_n)$ *absolute asymptotic condition number*, corresponding to the datum d_n.

Stability, consistency and convergence issues

Relative/absolute asymptotic condition number of the numerical method

The numerical method is said to be well conditioned if K^{num} is "small" for any admissible datum d_n, ill conditioned otherwise. As in (1.6), let us consider the case where, for each n, the functional relation (1.1) defines a mapping G_n between the sets of the numerical data and the solutions

$$x_n = G_n(d_n), \quad \text{that is } F_n(G_n(d_n), d_n) = 0. \tag{1.18}$$

Assuming that G_n is differentiable, we can obtain from (2.17)

$$K_n(d_n) \simeq \|G_n'(d_n)\| \frac{\|d_n\|}{\|G_n(d_n)\|}, \qquad K_{abs,n}(d_n) \simeq \|G_n'(d_n)\|. \tag{1.19}$$

Stability, consistency and convergence issues

Relative/absolute asymptotic condition number of the numerical method

Example (Sum and subtraction) The function $f : \mathbb{R}^2 \to \mathbb{R}$, $f(a,b) = a+b$, is a linear mapping whose gradient is the vector $f'(a,b) = (1,1)^T$. Using

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}, \qquad \text{for } 1 \leq p < \infty,$$

the vector norm $\|\cdot\|_1$ defined in then yields $K(a,b) \simeq (|a|+|b|)/(|a+b|)$, from which it follows that summing two numbers of the same sign is a well conditioned operation, being $K(a,b) \simeq 1$. On the other hand, subtracting two numbers almost equal is ill conditioned, since $|a+b| \ll |a|+|b|$. This fact, already pointed out in before then , leads to the *cancellation of significant digits* whenever numbers can be represented using only a finite number of digits (as in *floating-point* arithmetic)

Stability, consistency and convergence issues

Relative/absolute asymptotic condition number of the numerical method

Example Consider again the problem of computing the roots of a polynomial of second degree analyzed in the above. When $p > 1$ (separated roots), such a problem is well conditioned. However, we generate an *unstable* algorithm if we evaluate the root x_- by the formula $x_- = p - \sqrt{p^2 - 1}$. This formula is indeed subject to errors due to *numerical cancellation* of significant digits (see definition) that are introduced by the finite arithmetic of the computer. A possible remedy to this trouble consists of computing $x_+ = p + \sqrt{p^2 - 1}$ at first, then $x_- = 1/x_+$. Alternatively, one can solve $F(x, p) = x^2 - 2px + 1 = 0$ using Newton's method

$$x_n = x_{n-1} - (x_{n-1}^2 - 2px_{n-1} + 1)/(2x_{n-1} - 2p) = f_n(p), \quad n \geq 1, \quad x_0 \text{ given.}$$

Applying (1.19) for $p > 1$ yields $K_n(p) \simeq |p|/|x_n - p|$. To compute $K^{num}(p)$ we notice that, in the case when the algorithm converges, the solution x_n would converge to one of the roots x_+ or x_-; therefore, $|x_n - p| \to \sqrt{p^2 - 1}$ and thus $K_n(p) \to K^{num}(p) \simeq |p|/\sqrt{p^2 - 1}$, in perfect agreement with the value (1.8) of the condition number of the exact problem.

We can conclude that Newton's method for the search of simple roots of a second order algebraic equation is ill conditioned if $|p|$ is very close to 1, while it is well conditioned in the other cases.

Stability, consistency and convergence issues

Algorithm, numerical approximation and convergence

The final goal of numerical approximation is, of course, to build, through numerical problems of the type (1.12), solutions x_n that "get closer" to the solution of problem (1.1) as much as n gets larger. This concept is made precise in the next definition.

Definition 1.2 The numerical method (1.12) is *convergent* iff

$$\begin{aligned} &\forall \varepsilon > 0 \ \exists n_0(\varepsilon), \ \exists \delta(n_0, \varepsilon) > 0 \ : \\ &\forall n > n_0(\varepsilon), \ \forall \|\delta d_n\| < \delta(n_0, \varepsilon) \qquad \Rightarrow \|x(d) - x_n(d + \delta d_n)\| \leq \varepsilon, \end{aligned} \tag{1.20}$$

where d is an admissible datum for the problem (1.1), $x(d)$ is the corresponding solution and $x_n(d+\delta d_n)$ is the solution of the numerical problem (1.12) with datum $d + \delta d_n$.

Stability, consistency and convergence issues

Algorithm, numerical approximation and convergence

To verify the implication (1.20) it suffices to check that under the same assumptions

$$\|x(d+\delta d_n) - x_n(d+\delta d_n)\| \leq \frac{\varepsilon}{2}. \tag{1.21}$$

Indeed, thanks to (1.3) we have

$$\|x(d) - x_n(d+\delta d_n)\| \leq \|x(d) - x(d+\delta d_n)\|$$

$$+\|x(d+\delta d_n) - x_n(d+\delta d_n)\| \leq K(\delta(n_0,\varepsilon),d)\|\delta d_n\| + \tfrac{\varepsilon}{2}.$$

Choosing δd_n such that $K(\delta(n_0,\varepsilon),d)\|\delta d_n\| < \frac{\varepsilon}{2}$, one obtains (1.20).

Stability, consistency and convergence issues

Measures of the convergence, matrix or vector quantities

Measures of the convergence of x_n to x are given by the *absolute error* or the *relative error*, respectively defined as

$$E(x_n) = |x - x_n|, \qquad E_{rel}(x_n) = \frac{|x - x_n|}{|x|}, \quad (\text{if } x \neq 0). \tag{1.22}$$

In the cases where x and x_n are matrix or vector quantities, in addition to the definitions in (1.22) (where the absolute values are substituted by suitable norms) it is sometimes useful to introduce the *error by component* defined as

$$E^c_{rel}(x_n) = \max_{i,j} \frac{|(x - x_n)_{ij}|}{|x_{ij}|}. \tag{1.23}$$

Relations between Stability and Convergence

Well posed (stability) + consistency linked to convergent numerical methods

The concepts of stability and convergence are strongly connected. First of all, if problem (1.1) is well posed, a *necessary* condition in order for the numerical problem (1.12) to be convergent is that it is stable.

Let us thus assume that the method is convergent, and prove that it is stable by finding a bound for $\|\delta x_n\|$. We have

$$\begin{aligned} \|\delta x_n\| &= \|x_n(d+\delta d_n) - x_n(d)\| \leq \|x_n(d) - x(d)\| \\ &+ \|x(d) - x(d+\delta d_n)\| + \|x(d+\delta d_n) - x_n(d+\delta d_n)\| \quad (1.24) \\ &\leq K(\delta(n_0,\varepsilon),d)\|\delta d_n\| + \varepsilon, \end{aligned}$$

having used (1.3) and (1.21) twice. From (1.24) we can conclude that, for n sufficiently large, $\|\delta x_n\|/\|\delta d_n\|$ can be bounded by a constant of the order of $K(\delta(n_0,\varepsilon),d)$, so that the method is stable. Thus, we are interested in stable numerical methods since only these can be convergent.

Relations between Stability and Convergence

Well posed (stability) + consistency linked to convergent numerical methods

The stability of a numerical method becomes a *sufficient* condition for the numerical problem (1.12) to converge if this latter is also consistent with problem (1.1). Indeed, under these assumptions we have

$$\begin{aligned} \|x(d+\delta d_n) - x_n(d+\delta d_n)\| &\leq \|x(d+\delta d_n) - x(d)\| \\ &+ \|x(d) - x_n(d)\| + \|x_n(d) - x_n(d+\delta d_n)\|. \end{aligned}$$

Thanks to (1.3), the first term at right-hand side can be bounded by $\|\delta d_n\|$ (up to a multiplicative constant independent of δd_n). A similar bound holds for the third term, due to the stability property (1.16). Finally, concerning the remaining term, if F_n is differentiable with respect to the variable x, an expansion in a Taylor series gives

$$F_n(x(d), d) - F_n(x_n(d), d) = \frac{\partial F_n}{\partial x}|_{(\overline{x},d)}(x(d) - x_n(d)),$$

Relations between Stability and Convergence

Well posed (stability) + consistency linked to convergent numerical methods

$$F_n(x(d), d) - F_n(x_n(d), d) = \frac{\partial F_n}{\partial x}|_{(\overline{x},d)}(x(d) - x_n(d)),$$

for a suitable $\overline{x}$ "between" $x(d)$ and $x_n(d)$. Assuming also that $\partial F_n/\partial x$ is invertible, we get

$$x(d) - x_n(d) = \left(\frac{\partial F_n}{\partial x}\right)^{-1}_{|(\overline{x},d)} [F_n(x(d), d) - F_n(x_n(d), d)]. \qquad (1.25)$$

On the other hand, replacing $F_n(x_n(d), d)$ with $F(x(d), d)$ (since both terms are equal to zero) and passing to the norms, we find

$$\|x(d) - x_n(d)\| \leq \left\| \left(\frac{\partial F_n}{\partial x}\right)^{-1}_{|(\overline{x},d)} \right\| \|F_n(x(d), d) - F(x(d), d)\|.$$

Stability, consistency and convergence issues

Well posed (stability) + consistency and convergent numerical methods

On the other hand, replacing $F_n(x_n(d), d)$ with $F(x(d), d)$ (since both terms are equal to zero) and passing to the norms, we find

$$\|x(d) - x_n(d)\| \leq \left\| \left(\frac{\partial F_n}{\partial x} \right)^{-1}_{|(\overline{x},d)} \right\| \|F_n(x(d), d) - F(x(d), d)\|.$$

Thanks to (1.13) we can thus conclude that $\|x(d) - x_n(d)\| \to 0$ for $n \to \infty$. The result that has just been proved, although stated in qualitative terms, is a milestone in numerical analysis, known as *equivalence theorem* (or Lax-Richtmyer theorem): "*for a consistent numerical method, stability is equivalent to convergence*".

A rigorous proof of this theorem is available in Lax-Richtmyer (1956) – see also Dahlquist (1956) – for the case of linear Cauchy problems and in Richtmyer-Morton (1967) for linear well-posed initial value problems.

P. D. Lax and R. Richtmyer (1956), Survey of the stability of linear finite difference equations. Communications on Pure and Applied Mathematics 9(2):267-293.

G. Dahlquist (1956) Convergence and Stability in the Numerical Integration of Ordinary Differential Equations. Math. Scand. 4: 33-53.

R. Richtmyer and K. Morton (1967) Difference Methods for Initial Value Problems. Wiley, New York.

A priori and a posteriori analysis

Forward and backward stability analysis

A priori and a posteriori analysis

Forward and backward stability analysis

The stability analysis of a numerical method can be carried out following different strategies:

1. *forward analysis*, which provides a bound to the variations $\|\delta x_n\|$ on the solution due to both perturbations in the data and to errors that are intrinsic to the numerical method;
2. *backward analysis*, which aims at estimating the perturbations that should be "impressed" to the data of a given problem in order to obtain the results actually computed under the assumption of working in exact arithmetic. Equivalently, given a certain computed solution $\widehat{x}_n$, backward analysis looks for the perturbations δd_n on the data such that $F_n(\widehat{x}_n, d_n + \delta d_n) = 0$. Notice that, when performing such an estimate, *no* account at all is taken into the way $\widehat{x}_n$ has been obtained (that is, which method has been employed to generate it).

A priori and a posteriori analysis

Forward and backward stability analysis

The stability analysis of a numerical method can be carried out following different strategies:

1. *forward analysis*, which provides a bound to the variations $\|\delta x_n\|$ on the solution due to both perturbations in the data and to errors that are intrinsic to the numerical method;
2. *backward analysis*, which aims at estimating the perturbations that should be "impressed" to the data of a given problem in order to obtain the results actually computed under the assumption of working in exact arithmetic. Equivalently, given a certain computed solution $\widehat{x}_n$, backward analysis looks for the perturbations δd_n on the data such that $F_n(\widehat{x}_n, d_n + \delta d_n) = 0$. Notice that, when performing such an estimate, *no* account at all is taken into the way $\widehat{x}_n$ has been obtained (that is, which method has been employed to generate it).

A priori and a posteriori analysis

Forward and backward stability analysis: a first example

Forward and backward analyses are two different instances of the so called *a priori analysis*. This latter can be applied to investigate not only the stability of a numerical method, but also its convergence. In this case it is referred to as *a priori error analysis*, which can again be performed using either a forward or a backward technique.

A priori error analysis is distincted from the so called *a posteriori error analysis*, which aims at producing an estimate of the error on the grounds of quantities that are actually computed by a specific numerical method. Typically, denoting by $\widehat{x}_n$ the computed numerical solution, approximation to the solution x of problem (2.1), the *a posteriori* error analysis aims at evaluating the error $x - \widehat{x}_n$ as a function of the residual $r_n = F(\widehat{x}_n, d)$ by means of constants that are called *stability factors*.

A priori and a posteriori analysis

Forward and backward stability analysis: a second example

Example For the sake of illustration, consider the problem of finding the zeros $\alpha_1, \ldots, \alpha_n$ of a polynomial $p_n(x) = \sum_{k=0}^{n} a_k x^k$ of degree n.

Denoting by $\tilde{p}_n(x) = \sum_{k=0}^{n} \tilde{a}_k x^k$ a perturbed polynomial whose zeros are $\tilde{\alpha}_i$, forward analysis aims at estimating the error between two corresponding zeros α_i and $\tilde{\alpha}_i$, in terms of the variations on the coefficients $a_k - \tilde{a}_k$, $k = 0, 1, \ldots, n$.

On the other hand, let $\{\hat{\alpha}_i\}$ be the approximate zeros of p_n (computed somehow). Backward analysis provides an estimate of the perturbations δa_k which should be impressed to the coefficients so that $\sum_{k=0}^{n}(a_k + \delta a_k)\hat{\alpha}_i^k = 0$, for a fixed $\hat{\alpha}_i$. The goal of *a posteriori* error analysis would rather be to provide an estimate of the error $\alpha_i - \hat{\alpha}_i$ as a function of the residual value $p_n(\hat{\alpha}_i)$.

A priori and a posteriori analysis

Forward and backward stability analysis: a second example

Example Consider the linear system $A\mathbf{x}=\mathbf{b}$, where $A \in \mathbb{R}^{n\times n}$ is a nonsingular matrix.

For the perturbed system $\tilde{A}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$, forward analysis provides an estimate of the error $\mathbf{x} - \tilde{\mathbf{x}}$ in terms of $A - \tilde{A}$ and $\mathbf{b} - \tilde{\mathbf{b}}$, while backward analysis estimates the perturbations $\delta A = (\delta a_{ij})$ and $\delta \mathbf{b} = (\delta b_i)$ which should be impressed to the entries of A and $\mathbf{b}$ in order to get $(A + \delta A)\widehat{\mathbf{x}}_n = \mathbf{b} + \delta \mathbf{b}$, $\widehat{\mathbf{x}}_n$ being the solution of the linear system (computed somehow). Finally, *a posteriori* error analysis looks for an estimate of the error $\mathbf{x} - \widehat{\mathbf{x}}_n$ as a function of the residual $\mathbf{r}_n = \mathbf{b} - A\widehat{\mathbf{x}}_n$.

A priori and a posteriori analysis

It is important to point out the role played by the a posteriori analysis in devising strategies for adaptive error control.

These strategies, by suitably changing the discretization parameters (for instance, the spacing between nodes in the numerical integration of a function or a differential equation), employ the a posteriori analysis in order to ensure that the error does not exceed a fixed tolerance.

A numerical method that makes use of an adaptive error control is called adaptive numerical method. It is also time consuming to use!

In practice, a method of this kind applies in the computational process the idea of feedback, by activating on the grounds of a computed solution a convergence test which ensures the control of error within a fixed tolerance.

In case the convergence test fails, a suitable strategy for modifying the discretization parameters is automatically adopted in order to enhance the accuracy of the solution to be newly computed, and the overall procedure is iterated until the convergence check is passed.

LEAST SQUARE PROBLEM

QR Factorization and Singular Value Decomposition (SVD)

Least Squares Problem

Solution of Least Squares Problems

- More robust approach is to use QR factorization $A = \hat{Q}\hat{R}$
  - b can be projected onto range(A) by $P = \hat{Q}\hat{Q}^T$, and therefore $\hat{Q}\hat{R}x = \hat{Q}\hat{Q}^T b$
  - Left-multiply by $\hat{Q}^T$ and we get $\hat{R}x = \hat{Q}^T b$ (note $A^+ = \hat{R}^{-1}\hat{Q}^T$)

> Least squares via QR Factorization
>
> Compute reduced QR factorization $A = \hat{Q}\hat{R}$
> Compute vector $c = \hat{Q}^T b$
> Solve upper-triangular system $\hat{R}x = c$ for x

- Computation is dominated by QR factorization ($2mn^2 - \frac{2}{3}n^3$)
- Question: If Householder QR is used, how to compute $\hat{Q}^T b$?
- Answer: Compute $Q^T b$ (where Q is from full QR factorization) and then take first n entries of resulting $Q^T b$

Least Squares Problem

Solution of Least Squares Problems

- For a QR factorization $A = QR$ computed by Householder triangularization, the factors $\tilde{Q}$ and $\tilde{R}$ satisfy

$$\tilde{Q}\tilde{R} = A + \delta A, \quad \|\delta A\|/\|A\| = O(\epsilon_{\text{machine}}),$$

  i.e., exact QR factorization of a slightly perturbed A
- $\tilde{R}$ is R computed by algorithm using floating points
- However, $\tilde{Q}$ is product of *exactly orthogonal* reflectors

$$\tilde{Q} = \tilde{Q}_1 \tilde{Q}_2 \dots \tilde{Q}_n$$

  where $\tilde{Q}_k$ is given by computed $\tilde{v}_k$, since Q is not formed explicitly

Backward Stability of Solving $Ax = b$ with QR

Least Squares Problems

Algorithm: Solving $Ax = b$ by QR Factorization

Compute $A = QR$ using Householder, represent Q by reflectors
Compute vector $y = Q^T b$ implicitly using reflectors
Solve upper-triangular system $Rx = y$ for x

- All three steps are backward stable
- Overall, we can show that

$$(A + \Delta A)\tilde{x} = b, \quad \|\Delta A\|/\|A\| = O(\epsilon_{\text{machine}})$$

as we prove next

Proof of Backward Stability

Backward Stability of Solving $Ax = b$ with Householder QR

Proof: Step 2 gives

$$(\tilde{Q} + \delta Q)\tilde{y} = b, \quad \|\delta Q\| = O(\epsilon_{\text{machine}})$$

Step 3 gives

$$(\tilde{R} + \delta R)\tilde{x} = \tilde{y}, \quad \|\delta R\|/\|\tilde{R}\| = O(\epsilon_{\text{machine}})$$

Therefore,

$$b = (\tilde{Q} + \delta Q)(\tilde{R} + \delta R)\tilde{x} = \left[\tilde{Q}\tilde{R} + (\delta Q)\tilde{R} + \tilde{Q}(\delta R) + (\delta Q)(\delta R)\right]\tilde{x}$$

Step 1 gives

$$b = \left[A + \underbrace{\delta A + (\delta Q)\tilde{R} + \tilde{Q}(\delta R) + (\delta Q)(\delta R)}_{\Delta A}\right]\tilde{x}$$

where $\tilde{Q}\tilde{R} = A + \delta A$

Proof of Backward Stability (Cont.)

Backward Stability of Solving $Ax = b$ with Householder QR

$\tilde{Q}\tilde{R} = A + \delta A$ where $\|\delta A\|/\|A\| = O(\epsilon_{\text{machine}})$, and therefore

$$\frac{\|\tilde{R}\|}{\|A\|} \leq \|\tilde{Q}^T\| \frac{\|A + \delta A\|}{\|A\|} = O(1)$$

Now show that each term in ΔA is small

$$\frac{\|(\delta Q)\tilde{R}\|}{\|A\|} \leq \|(\delta Q)\| \frac{\|\tilde{R}\|}{\|A\|} = O(\epsilon_{\text{machine}})$$

$$\frac{\|\tilde{Q}(\delta R)\|}{\|A\|} \leq \|\tilde{Q}\| \frac{\|\delta R\|}{\|\tilde{R}\|} \frac{\|\tilde{R}\|}{\|A\|} = O(\epsilon_{\text{machine}})$$

$$\frac{\|(\delta Q)(\delta R)\|}{\|A\|} \leq \|\delta Q\| \frac{\|\delta R\|}{\|A\|} = O(\epsilon^2_{\text{machine}})$$

Overall,

$$\frac{\|\Delta A\|}{\|A\|} \leq \frac{\|\delta A\|}{\|A\|} + \frac{\|(\delta Q)\tilde{R}\|}{\|A\|} + \frac{\|\tilde{Q}(\delta R)\|}{\|A\|} + \frac{\|(\delta Q)(\delta R)\|}{\|A\|} = O(\epsilon_{\text{machine}})$$

Since the algorithm is backward stable, it is also accurate.

Stability of Gram-Schmidt Orthogonalization

- Gram-Schmidt QR is unstable, due to loss of orthogonality
- Gram-Schmidt can be stabilized using augmented system of equations
  1. Compute QR factorization of augmented matrix: [Q,R1]=mgs([A,b])
  2. Extract R and $\hat{Q}^T b$ from $R1$: R=R1(1:n,1:n); Qb=R1(1:n,n+1)
  3. Back solve: x=R\Qb

Theorem

The solution of the full-rank least squares problem by Gram-Schmidt orthogonality is backward stable in the sense that the computed solution $\tilde{x}$ has the property

$$\|(A+\delta A)\tilde{x}-b)\| = \min, \quad \frac{\|\delta A\|}{\|A\|} = O(\epsilon_{machine})$$

for some $\delta A \in \mathbb{R}^{m\times n}$, provided that $\hat{Q}^T b$ is formed implicitly.

The method of normal equations

- The method of *normal equation* solves $x = (A^T A)^{-1} A^T b$, due to squaring of condition number of A

Theorem

The solution of the full-rank least squares problem via normal equation is unstable. Stability can be achieved, however, by restriction to a class of problems in which $\kappa(A)$ is uniformly bounded above.

- Another method is to SVD (coming up next)

Summary of Algorithms for Least Square problems

- Householder QR (with/without pivoting, explicit or implicit Q): **Backward stable**
- Classical Gram-Schmidt: **Unstable**
- Modified Gram-Schmidt with explicit Q: **Unstable**
- Modified Gram-Schmidt with augmented system of equations with implicit Q: **Backward stable**
- Normal equations (solve $A^T Ax = A^T b$): **Very unstable**
- Singular value decomposition: **Backward stable**

Singular Value Decomposition (SVD)

- The image of unit sphere under any $m \times n$ matrix is a *hyperellipsoid*
- Give a unit sphere S in $\mathbb{R}^n$, let AS denote the shape after transformation
- SVD is

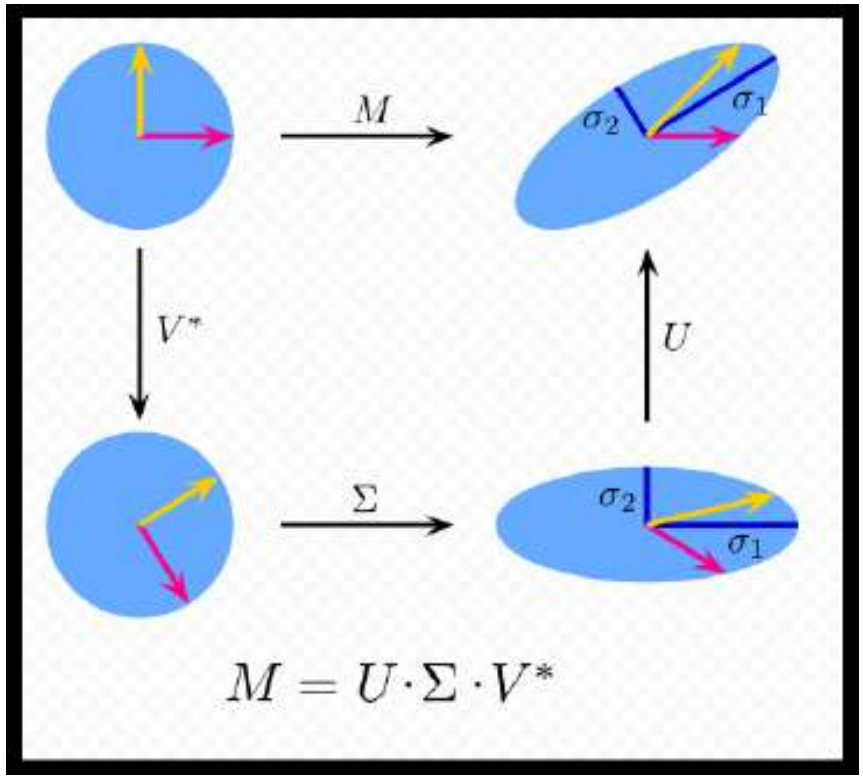
$$A = U\Sigma V^T$$

  where $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ is orthogonal and $\Sigma \in \mathbb{R}^{m \times n}$ is diagonal
- *Singular values* are diagonal entries of Σ, correspond to the principal semiaxes, with entries $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$.
- *Left singular vectors* of A are column vectors of U and are oriented in the directions of the principal semiaxes of AS
- *Right singular vectors* of A are column vectors of V and are the preimages of the principal semiaxes of AS
- $Av_j = \sigma_j u_j$ for $1 \leq j \leq n$

Some remarks:

- $A \in \mathbb{C}^{n \times n}$, $U \in \mathbb{C}^{n \times n}$, $\Sigma \in \mathbb{C}^{n \times n}$, $V^* \in \Re^{n \times n}$.
- Note that the diagonal matrix Σ has the same shape as A even when A is not square, but U and V^* are always square unitary matrices.
- The singular values of a matrix A are precisely the lengths of the semi-axes of the hyperellipsoid E defined by $E = \{Ax : ||x||_2 = 1\}$.

SVD (Geometric Observation)



The image[2] shows:

Upper Left: The unit disc with the two canonical unit vectors. It is clear that the image of the unit sphere in $\Re^n$ under a map $A = U\Sigma V^*$ must be a hyperellipse in $\Re^n$.

Lower Left: The action of V^* on the unit disc. This is just a rotation. The unitary map V^* preserves the sphere.

Lower Right: The action of ΣV^* on the unit disc. Sigma scales in vertically and horizontally. The diagonal matrix Σ stretches the sphere into a hyperellipse aligned with the canonical basis.

Upper Right: Unit disc transformed with M and singular Values σ_1 and σ_2 indicate. Finally, the latter unitary map U rotates or reflects the hyperellipse without changing its shape.

[2] thanks to wikipedia

Two Different Types of SVD

- **Full SVD**: $U \in \mathbb{R}^{m \times m}$, $\Sigma \in \mathbb{R}^{m \times n}$, and $V \in \mathbb{R}^{n \times n}$ is

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

- **Reduced SVD**: $\hat{U} \in \mathbb{R}^{m \times n}$, $\hat{\Sigma} \in \mathbb{R}^{n \times n}$ (assume $m \geq n$)

$$A = \hat{U} \hat{\Sigma} V^T$$

- Furthermore, notice that

$$A = \sum_{i=1}^{\min\{m,n\}} \sigma_i u_i v_i^T$$

so we can keep only entries of U and V corresponding to nonzero σ_i.

SVD versus Eigenvalue Decomposition

- Eigenvalue decomposition of nondefective matrix A is $A = X\Lambda X^{-1}$
- Differences between SVD and Eigenvalue Decomposition
  - Not every matrix has eigenvalue decomposition, but every matrix has singular value decomposition
  - Eigenvalues may not always be real numbers, but singular values are always non-negative real numbers
  - Eigenvectors are not always orthogonal to each other (orthogonal for symmetric matrices), but left (or right) singular vectors are orthogonal to each other
- Similarities
  - Singular values of A are square roots of eigenvalues of AA^T and A^TA, and their eigenvectors are left and right singular vectors, respectively
  - Singular values of symmetric matrices are absolute values of eigenvalues, and eigenvectors are singular vectors
  - This relationship can be used to compute singular values by hand

Existence of SVD (sketch of the proof)

Theorem

(Existence) Every matrix $A \in \mathbb{R}^{m \times n}$ has an SVD.

Proof: Let $\sigma_1 = \|A\|_2$. There exists $v_1 \in \mathbb{R}^n$ with $\|v_1\|_2 = 1$ and $\|Av_1\|_2 = \sigma_1$. Let U_1 and V_1 be orthogonal matrices whose first columns are $u_1 = Av_1/\sigma_1$ (or any unit-length vector if $\sigma_1 = 0$) and v_1, respectively. Note that

$$U_1^T A V_1 = S = \begin{bmatrix} \sigma_1 & \omega^T \\ 0 & B \end{bmatrix}. \tag{1}$$

Furthermore, $\omega = 0$ because $\|S\|_2 = \sigma_1$, and

$$\left\| \begin{bmatrix} \sigma_1 & \omega^T \\ 0 & B \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \omega \end{bmatrix} \right\|_2 \geq \sigma_1^2 + \omega^T\omega = \sqrt{\sigma_1^2 + \omega^T\omega} \left\| \begin{bmatrix} \sigma_1 \\ \omega \end{bmatrix} \right\|_2,$$

implying that $\sigma_1 \geq \sqrt{\sigma_1^2 + \omega^T\omega}$ and $\omega = 0$.

Existence of SVD: Based on 2-norm and prove by induction

We then prove by induction using (1). If $m = 1$ or $n = 1$, then B is empty and we have $A = U_1 S V_1^T$. Otherwise, suppose $B = U_2 \Sigma_2 V_2^T$, and then

$$A = \underbrace{U_1 \begin{bmatrix} 1 & 0^T \\ 0 & U_2 \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} \sigma_1 & 0^T \\ 0 & \Sigma_2 \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} 1 & 0^T \\ 0 & V_2^T \end{bmatrix} V_1^T}_{V^T},$$

where U and V are orthogonal.

Existence and Uniqueness (Theorem): Every matrix $A \in \mathcal{C}^{n \times n}$ has a singular value decomposition $A = U \Sigma V^*$. Furthermore, the singular values $\{\sigma_i\}$ are uniquely determined, and, if A is square and the σ_i are distinct, the left and right singular vectors $\{u_i\}$ and $\{v_i\}$ are uniquely determined up to complex signs (i.e., complex scalar factors of absolute value 1). *Gene H. Golub and Charles F. Van Loan. Matrix computations, 3rd ed., Johns Hopkins University Press (1996)*.

Numerical methods for the SVD are based on the *QR* (Francis) iterative algorithms and its variants; Golub & Van Loan(1996).

BACKGROUND IN VECTORS, MATRICES AND NORMS

ORIENTED TO NUMERICAL LINEAR ALGEBRA

Obs.: For more details on these topics see list of references at the course syllabus MS993/MT404

Matrices

- Multiplication by another matrix:

$$C = AB,$$

where $A \in \mathbb{C}^{n\times m}, B \in \mathbb{C}^{m\times p}, C \in \mathbb{C}^{n\times p}$, and

$$c_{ij} = \sum_{k=1}^{m} a_{ik} b_{kj}.$$

Sometimes, a notation with column vectors and row vectors is used. The column vector a_{*j} is the vector consisting of the j-th column of A,

$$a_{*j} = \begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{nj} \end{pmatrix}.$$

Similarly, the notation a_{i*} will denote the i-th row of the matrix A

$$a_{i*} = (a_{i1}, a_{i2}, \ldots, a_{im}).$$

Matrices (Cont.)

For example, the following could be written

$$A = (a_{*1}, a_{*2}, \ldots, a_{*m}) ,$$

or

$$A = \begin{pmatrix} a_{1*} \\ a_{2*} \\ . \\ . \\ a_{n*} \end{pmatrix} .$$

The *transpose* of a matrix A in $\mathbb{C}^{n\times m}$ is a matrix C in $\mathbb{C}^{m\times n}$ whose elements are defined by $c_{ij} = a_{ji}, i = 1, \ldots, m,\ j = 1, \ldots, n$. It is denoted by A^T. It is often more relevant to use the *transpose conjugate* matrix denoted by A^H and defined by

$$A^H = \bar{A}^T = \overline{A^T},$$

in which the bar denotes the (element-wise) complex conjugation.

Matrices are strongly related to linear mappings between vector spaces of finite dimension. This is because they represent these mappings with respect to two given bases: one for the initial vector space and the other for the image vector space, or *range* of A.

Square Matrices and Eigenvalues

A matrix is *square* if it has the same number of columns and rows, i.e., if $m = n$. An important square matrix is the identity matrix

$$I = \{\delta_{ij}\}_{i,j=1,\ldots,n},$$

where δ_{ij} is the Kronecker symbol. The identity matrix satisfies the equality $AI = IA = A$ for every matrix A of size n. The inverse of a matrix, when it exists, is a matrix C such that

$$CA = AC = I.$$

The inverse of A is denoted by A^{-1}.

Square Matrices and Eigenvalues (Cont.)

The *determinant* of a matrix may be defined in several ways. For simplicity, the following recursive definition is used here. The determinant of a 1×1 matrix (a) is defined as the scalar a. Then the determinant of an $n \times n$ matrix is given by

$$\det(A) = \sum_{j=1}^{n} (-1)^{j+1} a_{1j} \det(A_{1j}),$$

where A_{1j} is an $(n-1) \times (n-1)$ matrix obtained by deleting the first row and the j-th column of A. A matrix is said to be *singular* when $\det(A) = 0$ and *nonsingular* otherwise. We have the following simple properties:

- $\det(AB) = \det(A)\det(B)$.
- $\det(A^T) = \det(A)$.
- $\det(\alpha A) = \alpha^n \det(A)$.
- $\det(\bar{A}) = \overline{\det(A)}$.
- $\det(I) = 1$.

Square Matrices and Eigenvalues (Cont.)

From the above definition of determinants it can be shown by induction that the function that maps a given complex value λ to the value $p_A(\lambda) = \det(A - \lambda I)$ is a polynomial of degree n; see Exercise 8. This is known as the *characteristic polynomial* of the matrix A.

Definition 1.1 *A complex scalar λ is called an eigenvalue of the square matrix A if a nonzero vector u of $\mathbb{C}^n$ exists such that $Au = \lambda u$. The vector u is called an eigenvector of A associated with λ. The set of all the eigenvalues of A is called the spectrum of A and is denoted by $\sigma(A)$.*

A scalar λ is an eigenvalue of A if and only if $\det(A - \lambda I) \equiv p_A(\lambda) = 0$. That is true *if and only if* (*iff* thereafter) λ is a root of the characteristic polynomial. In particular, there are at most n distinct eigenvalues.

It is clear that a matrix is singular if and only if it admits zero as an eigenvalue. A well known result in linear algebra is stated in the following proposition.

Square Matrices and Eigenvalues (Cont.)

Proposition 1.2 *A matrix A is nonsingular if and only if it admits an inverse.*

Thus, the determinant of a matrix determines whether or not the matrix admits an inverse.

The maximum modulus of the eigenvalues is called *spectral radius* and is denoted by $\rho(A)$

$$\rho(A) = \max_{\lambda \in \sigma(A)} |\lambda|.$$

The *trace* of a matrix is equal to the sum of all its diagonal elements

$$\mathrm{tr}(A) = \sum_{i=1}^{n} a_{ii}.$$

It can be easily shown that the trace of A is also equal to the sum of the eigenvalues of A counted with their multiplicities as roots of the characteristic polynomial.

Square Matrices and Eigenvalues (Cont.)

Proposition 1.3 *If λ is an eigenvalue of A, then $\bar{\lambda}$ is an eigenvalue of A^H. An eigenvector v of A^H associated with the eigenvalue $\bar{\lambda}$ is called a left eigenvector of A.*

When a distinction is necessary, an eigenvector of A is often called a right eigenvector. Therefore, the eigenvalue λ as well as the right and left eigenvectors, u and v, satisfy the relations

$$Au = \lambda u, \quad v^H A = \lambda v^H,$$

or, equivalently,

$$u^H A^H = \bar{\lambda} u^H, \quad A^H v = \bar{\lambda} v.$$

Types of Matrices

The choice of a method for solving linear systems will often depend on the structure of the matrix A. One of the most important properties of matrices is symmetry, because of its impact on the eigenstructure of A. A number of other classes of matrices also have particular eigenstructures. The most important ones are listed below:

- *Symmetric matrices:* $A^T = A$.
- *Hermitian matrices:* $A^H = A$.
- *Skew-symmetric matrices:* $A^T = -A$.
- *Skew-Hermitian matrices:* $A^H = -A$.
- *Normal matrices:* $A^H A = AA^H$.
- *Nonnegative matrices:* $a_{ij} \geq 0,\ i, j = 1, \ldots, n$ (similar definition for nonpositive, positive, and negative matrices).

Types of Matrices (Cont.)

- *Unitary matrices:* $Q^H Q = I$.

It is worth noting that a unitary matrix Q is a matrix whose inverse is its transpose conjugate Q^H, since

$$Q^H Q = I \quad \rightarrow \quad Q^{-1} = Q^H. \tag{1.1}$$

A matrix Q such that $Q^H Q$ is diagonal is often called orthogonal.

Some matrices have particular structures that are often convenient for computational purposes. The following list, though incomplete, gives an idea of these special matrices which play an important role in numerical analysis and scientific computing applications.

- *Diagonal matrices:* $a_{ij} = 0$ for $j \neq i$. Notation:

$$A = \text{diag}\ (a_{11}, a_{22}, \ldots, a_{nn}).$$

Types of Matrices (Cont.)

- *Upper triangular matrices:* $a_{ij} = 0$ for $i > j$.
- *Lower triangular matrices:* $a_{ij} = 0$ for $i < j$.
- *Upper bidiagonal matrices:* $a_{ij} = 0$ for $j \neq i$ or $j \neq i + 1$.
- *Lower bidiagonal matrices:* $a_{ij} = 0$ for $j \neq i$ or $j \neq i - 1$.
- *Tridiagonal matrices:* $a_{ij} = 0$ for any pair i, j such that $|j - i| > 1$. Notation:

$$A = \text{tridiag } (a_{i,i-1}, a_{ii}, a_{i,i+1}) .$$

- *Banded matrices:* $a_{ij} \neq 0$ only if $i - m_l \leq j \leq i + m_u$, where m_l and m_u are two nonnegative integers. The number $m_l + m_u + 1$ is called the bandwidth of A.
- *Upper Hessenberg matrices:* $a_{ij} = 0$ for any pair i, j such that $i > j + 1$. Lower Hessenberg matrices can be defined similarly.

Types of Matrices (Cont.)

- *Outer product matrices:* $A = uv^H$, where both u and v are vectors.
- *Permutation matrices:* the columns of A are a permutation of the columns of the identity matrix.
- *Block diagonal matrices:* generalizes the diagonal matrix by replacing each diagonal entry by a matrix. Notation:

$$A = \text{diag}\ (A_{11}, A_{22}, \ldots, A_{nn})\,.$$

- *Block tridiagonal matrices:* generalizes the tridiagonal matrix by replacing each nonzero entry by a square matrix. Notation:

$$A = \text{tridiag}\ (A_{i,i-1}, A_{ii}, A_{i,i+1})\,.$$

The above properties emphasize structure, i.e., positions of the nonzero elements with respect to the zeros. Also, they assume that there are many zero elements or that the matrix is of low rank. This is in contrast with the classifications listed earlier, such as symmetry or normality.

Vector Inner Products and Norms

An inner product on a (complex) vector space $\mathbb{X}$ is any mapping s from $\mathbb{X} \times \mathbb{X}$ into $\mathbb{C}$,

$$x \in \mathbb{X}, y \in \mathbb{X} \quad \rightarrow \quad s(x,y) \in \mathbb{C},$$

which satisfies the following conditions:

1. $s(x,y)$ is linear with respect to x, i.e.,

$$s(\lambda_1 x_1 + \lambda_2 x_2, y) = \lambda_1 s(x_1, y) + \lambda_2 s(x_2, y), \quad \forall\, x_1, x_2 \in \mathbb{X}, \forall\, \lambda_1, \lambda_2 \in \mathbb{C}.$$

2. $s(x,y)$ is *Hermitian*, i.e.,

$$s(y,x) = \overline{s(x,y)}, \quad \forall\, x, y \in \mathbb{X}.$$

3. $s(x,y)$ is *positive definite*, i.e.,

$$s(x,x) > 0, \quad \forall\, x \neq 0.$$

Vector Inner Products and Norms (Cont.)

Note that (2) implies that $s(x, x)$ is real and therefore, (3) adds the constraint that $s(x, x)$ must also be positive for any nonzero x. For any x and y,

$$s(x, 0) = s(x, 0.y) = 0.s(x, y) = 0.$$

Similarly, $s(0, y) = 0$ for any y. Hence, $s(0, y) = s(x, 0) = 0$ for any x and y. In particular the condition (3) can be rewritten as

$$s(x, x) \geq 0 \quad \text{and} \quad s(x, x) = 0 \quad \text{iff} \quad x = 0,$$

as can be readily shown. A useful relation satisfied by any inner product is the so-called Cauchy-Schwartz inequality:

$$|s(x, y)|^2 \leq s(x, x)\, s(y, y). \tag{1.2}$$

Vector Inner Products and Norms (Cont.)

In the particular case of the vector space $\mathbb{X} = \mathbb{C}^n$, a "canonical" inner product is the *Euclidean inner product*. The Euclidean inner product of two vectors $x = (x_i)_{i=1,\ldots,n}$ and $y = (y_i)_{i=1,\ldots,n}$ of $\mathbb{C}^n$ is defined by

$$(x, y) = \sum_{i=1}^{n} x_i \bar{y}_i, \tag{1.3}$$

which is often rewritten in matrix notation as

$$(x, y) = y^H x. \tag{1.4}$$

It is easy to verify that this mapping does indeed satisfy the three conditions required for inner products, listed above. A fundamental property of the Euclidean inner product in matrix computations is the simple relation

$$(Ax, y) = (x, A^H y), \quad \forall\, x, y \in \mathbb{C}^n. \tag{1.5}$$

Vector Inner Products and Norms (Cont.)

The proof of this is straightforward. The *adjoint* of A *with respect to an arbitrary inner product* is a matrix B such that $(Ax, y) = (x, By)$ for all pairs of vectors x and y. A matrix is *self-adjoint*, or Hermitian with respect to this inner product, if it is equal to its adjoint. The following proposition is a consequence of the equality (1.5).

Proposition 1.4 *Unitary matrices preserve the Euclidean inner product, i.e.,*

$$(Qx, Qy) = (x, y)$$

for any unitary matrix Q and any vectors x and y.

Proof. Indeed, $(Qx, Qy) = (x, Q^H Qy) = (x, y)$. □

Vector Inner Products and Norms (Cont.)

A vector norm on a vector space $\mathbb{X}$ is a real-valued function $x \rightarrow \|x\|$ on $\mathbb{X}$, which satisfies the following three conditions:

1. $\|x\| \geq 0, \quad \forall\, x \in \mathbb{X}, \quad$ and $\quad \|x\| = 0$ iff $x = 0$.
2. $\|\alpha x\| = |\alpha| \|x\|, \quad \forall\, x \in \mathbb{X}, \quad \forall \alpha \in \mathbb{C}$.
3. $\|x + y\| \leq \|x\| + \|y\|, \quad \forall\, x, y \in \mathbb{X}$.

For the particular case when $\mathbb{X} = \mathbb{C}^n$, we can associate with the inner product (1.3) the *Euclidean norm* of a complex vector defined by

$$\|x\|_2 = (x, x)^{1/2}.$$

It follows from Proposition 1.4 that a unitary matrix preserves the Euclidean norm metric, i.e.,

$$\|Qx\|_2 = \|x\|_2, \ \forall\, x.$$

Matrix Norms

The linear transformation associated with a unitary matrix Q is therefore an *isometry*.

The most commonly used vector norms in numerical linear algebra are special cases of the Hölder norms

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}. \tag{1.6}$$

Note that the limit of $\|x\|_p$ when p tends to infinity exists and is equal to the maximum modulus of the x_i's. This defines a norm denoted by $\|.\|_\infty$. The cases $p = 1$, $p = 2$, and $p = \infty$ lead to the most important norms in practice,

$$\begin{aligned}
\|x\|_1 &= |x_1| + |x_2| + \cdots + |x_n|, \\
\|x\|_2 &= \left[|x_1|^2 + |x_2|^2 + \cdots + |x_n|^2 \right]^{1/2}, \\
\|x\|_\infty &= \max_{i=1,\ldots,n} |x_i|.
\end{aligned}$$

The Cauchy-Schwartz inequality of (1.2) becomes

$$|(x, y)| \leq \|x\|_2 \|y\|_2.$$

Matrix Norms (Cont.)

For a general matrix A in $\mathbb{C}^{n\times m}$, we define the following special set of norms

$$\|A\|_{pq} = \max_{x\in\mathbb{C}^m,\, x\neq 0} \frac{\|Ax\|_p}{\|x\|_q}. \tag{1.7}$$

The norm $\|.\|_{pq}$ is *induced* by the two norms $\|.\|_p$ and $\|.\|_q$. These norms satisfy the usual properties of norms, i.e.,

$$\|A\| \geq 0, \quad \forall\, A \in \mathbb{C}^{n\times m}, \quad \text{and} \quad \|A\| = 0 \quad \text{iff} \quad A = 0 \tag{1.8}$$

$$\|\alpha A\| = |\alpha|\|A\|, \forall\, A \in \mathbb{C}^{n\times m}, \quad \forall\, \alpha \in \mathbb{C} \tag{1.9}$$

$$\|A + B\| \leq \|A\| + \|B\|, \quad \forall\, A, B \in \mathbb{C}^{n\times m}. \tag{1.10}$$

$$\tag{1.11}$$

A norm which satisfies the above three properties is nothing but a *vector norm* applied to the matrix considered as a vector consisting of the m columns stacked into a vector of size nm.

Matrix Norms (Cont.)

The most important cases are again those associated with $p, q = 1, 2, \infty$. The case $q = p$ is of particular interest and the associated norm $\|.\|_{pq}$ is simply denoted by $\|.\|_p$ and called a "p-norm." A fundamental property of a p-norm is that

$$\|AB\|_p \leq \|A\|_p \|B\|_p,$$

an immediate consequence of the definition (1.7). Matrix norms that satisfy the above property are sometimes called *consistent*. Often a norm satisfying the properties (1.8–1.10) and which is consistent is called a *matrix norm*. A result of consistency is that for any square matrix A,

$$\|A^k\|_p \leq \|A\|_p^k.$$

In particular the matrix A^k converges to zero if *any* of its p-norms is less than 1.

Matrix Norms - The Frobenius norm of a matrix

The Frobenius norm of a matrix is defined by

$$\|A\|_F = \left(\sum_{j=1}^{m} \sum_{i=1}^{n} |a_{ij}|^2 \right)^{1/2} . \tag{1.12}$$

This can be viewed as the 2-norm of the column (or row) vector in $\mathbb{C}^{n^2}$ consisting of all the columns (respectively rows) of A listed from 1 to m (respectively 1 to n.) It can be shown that this norm is also consistent, in spite of the fact that it is not induced by a pair of vector norms, i.e., it is not derived from a formula of the form (1.7); However, it does not satisfy some of the other properties of the p-norms. For example, the Frobenius norm of the identity matrix is not equal to one. To avoid these difficulties, *we will only use the term matrix norm for a norm that is induced by two norms as in the definition (1.7).* Thus, we will not consider the Frobenius norm to be a proper matrix norm, according to our conventions, even though it is consistent.

Matrix Norms (Cont.)

The following equalities satisfied by the matrix norms defined above lead to alternative definitions that are often easier to work with:

$$\|A\|_1 = \max_{j=1,\ldots,m} \sum_{i=1}^{n} |a_{ij}|, \tag{1.13}$$

$$\|A\|_\infty = \max_{i=1,\ldots,n} \sum_{j=1}^{m} |a_{ij}|, \tag{1.14}$$

$$\|A\|_2 = \left[\rho(A^H A)\right]^{1/2} = \left[\rho(AA^H)\right]^{1/2}, \tag{1.15}$$

$$\|A\|_F = \left[\mathrm{tr}(A^H A)\right]^{1/2} = \left[\mathrm{tr}(AA^H)\right]^{1/2}. \tag{1.16}$$

As will be shown later, the eigenvalues of $A^H A$ are nonnegative. Their square roots are called *singular values* of A and are denoted by $\sigma_i, i = 1, \ldots, m$. Thus, the relation (1.15) states that $\|A\|_2$ is equal to σ_1, the largest singular value of A.

Matrix Norms (Cont.)

Example 1.1. From the relation (1.15), it is clear that the spectral radius $\rho(A)$ is equal to the 2-norm of a matrix when the matrix is Hermitian. However, it is not a matrix norm in general. For example, the first property of norms is not satisfied, since for

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

we have $\rho(A) = 0$ while $A \neq 0$. Also, the triangle inequality is not satisfied for the pair A, and $B = A^T$ where A is defined above. Indeed,

$$\rho(A + B) = 1 \quad \text{while} \quad \rho(A) + \rho(B) = 0.$$

Subspaces, Range, and Kernel

A subspace of $\mathbb{C}^n$ is a subset of $\mathbb{C}^n$ that is also a complex vector space. The set of all linear combinations of a set of vectors $G = \{a_1, a_2, \ldots, a_q\}$ of $\mathbb{C}^n$ is a vector subspace called the linear span of G,

$$\begin{aligned} \text{span}\{G\} &= \text{span}\{a_1, a_2, \ldots, a_q\} \\ &= \left\{ z \in \mathbb{C}^n \;\middle|\; z = \sum_{i=1}^{q} \alpha_i a_i;\ \{\alpha_i\}_{i=1,\ldots,q} \in \mathbb{C}^q \right\}. \end{aligned}$$

If the a_i's are linearly independent, then each vector of $\text{span}\{G\}$ admits a unique expression as a linear combination of the a_i's. The set G is then called a *basis* of the subspace $\text{span}\{G\}$.

Given two vector subspaces S_1 and S_2, their *sum* S is a subspace defined as the set of all vectors that are equal to the sum of a vector of S_1 and a vector of S_2. The intersection of two subspaces is also a subspace. If the intersection of S_1 and S_2 is reduced to $\{0\}$, then the sum of S_1 and S_2 is called their direct sum and is denoted by $S = S_1 \bigoplus S_2$. When S is equal to $\mathbb{C}^n$, then every vector x of $\mathbb{C}^n$ can be written in a unique way as the sum of an element x_1 of S_1 and an element x_2 of S_2. The transformation P that maps x into x_1 is a linear transformation that is *idempotent*, i.e., such that $P^2 = P$. It is called a *projector* onto S_1 along S_2.

Subspaces, Range, and Kernel (Cont.)

Two important subspaces that are associated with a matrix A of $\mathbb{C}^{n\times m}$ are its *range*, defined by

$$\mathrm{Ran}(A) = \{Ax \mid x \in \mathbb{C}^m\}, \tag{1.17}$$

and its *kernel* or *null space*

$$\mathrm{Null}(A) = \{x \in \mathbb{C}^m \mid Ax = 0\}.$$

The range of A is clearly equal to the linear *span* of its columns. The *rank* of a matrix is equal to the dimension of the range of A, i.e., to the number of linearly independent columns. This *column rank* is equal to the *row rank*, the number of linearly independent rows of A. A matrix in $\mathbb{C}^{n\times m}$ is of *full rank* when its rank is equal to the smallest of m and n. A fundamental result of linear algebra is stated by the following relation

$$\mathbb{C}^n = \mathrm{Ran}(A) \oplus \mathrm{Null}(A^T)\ . \tag{1.18}$$

The same result applied to the transpose of A yields: $\mathbb{C}^m = \mathrm{Ran}(A^T) \oplus \mathrm{Null}(A)$.

A subspace S is said to be *invariant* under a (square) matrix A whenever $AS \subset S$. In particular for any eigenvalue λ of A the subspace $\mathrm{Null}(A - \lambda I)$ is invariant under A. The subspace $\mathrm{Null}(A - \lambda I)$ is called the eigenspace associated with λ and consists of all the eigenvectors of A associated with λ, in addition to the zero-vector.

Orthogonal Vectors and Subspaces

A set of vectors $G = \{a_1, a_2, \ldots, a_r\}$ is said to be *orthogonal* if

$$(a_i, a_j) = 0 \quad \text{when} \quad i \neq j.$$

It is *orthonormal* if, in addition, every vector of G has a 2-norm equal to unity. A vector that is orthogonal to all the vectors of a subspace S is said to be orthogonal to this subspace. The set of all the vectors that are orthogonal to S is a vector subspace called the *orthogonal complement* of S and denoted by $S^{\perp}$. The space $\mathbb{C}^n$ is the direct sum of S and its orthogonal complement. Thus, any vector x can be written in a unique fashion as the sum of a vector in S and a vector in $S^{\perp}$. The operator which maps x into its component in the subspace S is the *orthogonal projector* onto S.

Orthogonal Vectors and Subspaces (Cont.)

The Gram-Schmidt process

A set of vectors $G = \{a_1, a_2, \ldots, a_r\}$ is said to be *orthogonal* if

$$(a_i, a_j) = 0 \quad \text{when} \quad i \neq j.$$

It is *orthonormal* if, in addition, every vector of G has a 2-norm equal to unity. A vector that is orthogonal to all the vectors of a subspace S is said to be orthogonal to this subspace. The set of all the vectors that are orthogonal to S is a vector subspace called the *orthogonal complement* of S and denoted by $S^{\perp}$. The space $\mathbb{C}^n$ is the direct sum of S and its orthogonal complement. Thus, any vector x can be written in a unique fashion as the sum of a vector in S and a vector in $S^{\perp}$. The operator which maps x into its component in the subspace S is the *orthogonal projector* onto S.

Orthogonal Vectors and Subspaces (Cont.)

The Gram-Schmidt process

ALGORITHM **1.1** *Gram-Schmidt*

1. *Compute* $r_{11} := \|x_1\|_2$. *If* $r_{11} = 0$ *Stop, else compute* $q_1 := x_1/r_{11}$.
2. *For* $j = 2, \ldots, r$ *Do:*
3. *Compute* $r_{ij} := (x_j, q_i)$, *for* $i = 1, 2, \ldots, j-1$
4. $\hat{q} := x_j - \sum_{i=1}^{j-1} r_{ij} q_i$
5. $r_{jj} := \|\hat{q}\|_2$,
6. *If* $r_{jj} = 0$ *then Stop, else* $q_j := \hat{q}/r_{jj}$
7. *EndDo*

It is easy to prove that the above algorithm will not break down, i.e., all r steps will be completed if and only if the set of vectors $x_1, x_2, \ldots, x_r$ is linearly independent. From lines 4 and 5, it is clear that at every step of the algorithm the following relation holds:

$$x_j = \sum_{i=1}^{j} r_{ij} q_i.$$

Orthogonal Vectors and Subspaces (Cont.)

The Modified Gram-Schmidt process

If $X = [x_1, x_2, \ldots, x_r]$, $Q = [q_1, q_2, \ldots, q_r]$, and if R denotes the $r \times r$ upper triangular matrix whose nonzero elements are the r_{ij} defined in the algorithm, then the above relation can be written as

$$X = QR. \tag{1.19}$$

This is called the QR decomposition of the $n \times r$ matrix X. From what was said above, the QR decomposition of a matrix exists whenever the column vectors of X form a linearly independent set of vectors.

The above algorithm is the standard Gram-Schmidt process. There are alternative formulations of the algorithm which have better numerical properties. The best known of these is the Modified Gram-Schmidt (MGS) algorithm.

Orthogonal Vectors and Subspaces (Cont.)

The Modified Gram-Schmidt process

ALGORITHM **1.2** *Modified Gram-Schmidt*

1. *Define* $r_{11} := \|x_1\|_2$. *If* $r_{11} = 0$ *Stop, else* $q_1 := x_1/r_{11}$.
2. *For* $j = 2, \ldots, r$ *Do:*
3. *Define* $\hat{q} := x_j$
4. *For* $i = 1, \ldots, j-1$, *Do:*
5. $r_{ij} := (\hat{q}, q_i)$
6. $\hat{q} := \hat{q} - r_{ij} q_i$
7. *EndDo*
8. *Compute* $r_{jj} := \|\hat{q}\|_2$,
9. *If* $r_{jj} = 0$ *then Stop, else* $q_j := \hat{q}/r_{jj}$
10. *EndDo*

Orthogonal Vectors and Subspaces (Cont.)

The Householder *reflectors* algorithm

Yet another alternative for orthogonalizing a sequence of vectors is the Householder algorithm. This technique uses Householder *reflectors*, i.e., matrices of the form

$$P = I - 2ww^T, \tag{1.20}$$

in which w is a vector of 2-norm unity. Geometrically, the vector Px represents a mirror image of x with respect to the hyperplane $\text{span}\{w\}^\perp$.

To describe the Householder orthogonalization process, the problem can be formulated as that of finding a QR factorization of a given $n \times m$ matrix X. For any vector x, the vector w for the Householder transformation (1.20) is selected in such a way that

$$Px = \alpha e_1,$$

where α is a scalar. Writing $(I - 2ww^T)x = \alpha e_1$ yields

$$2w^T x \, w = x - \alpha e_1. \tag{1.21}$$

This shows that the desired w is a multiple of the vector $x - \alpha e_1$,

$$w = \pm \frac{x - \alpha e_1}{\|x - \alpha e_1\|_2}.$$

Orthogonal Vectors and Subspaces (Cont.)

The Householder *reflectors* algorithm

For (1.21) to be satisfied, we must impose the condition

$$2(x - \alpha e_1)^T x = \|x - \alpha e_1\|_2^2$$

which gives $2(\|x\|_1^2 - \alpha\xi_1) = \|x\|_2^2 - 2\alpha\xi_1 + \alpha^2$, where $\xi_1 \equiv e_1^T x$ is the first component of the vector x. Therefore, it is necessary that

$$\alpha = \pm\|x\|_2.$$

In order to avoid that the resulting vector w be small, it is customary to take

$$\alpha = -\text{sign}(\xi_1)\|x\|_2,$$

which yields

$$w = \frac{x + \text{sign}(\xi_1)\|x\|_2 e_1}{\|x + \text{sign}(\xi_1)\|x\|_2 e_1\|_2}. \qquad (1.22)$$

Orthogonal Vectors and Subspaces (Cont.)

The Householder *reflectors* algorithm

Given an $n \times m$ matrix, its first column can be transformed to a multiple of the column e_1, by premultiplying it by a Householder matrix P_1,

$$X_1 \equiv P_1 X, \qquad X_1 e_1 = \alpha e_1.$$

Assume, inductively, that the matrix X has been transformed in $k-1$ successive steps into the partially upper triangular form

$$X_k \equiv P_{k-1} \dots P_1 X_1 = \begin{pmatrix} x_{11} & x_{12} & x_{13} & \cdots & \cdots & \cdots & x_{1m} \\ & x_{22} & x_{23} & \cdots & \cdots & \cdots & x_{2m} \\ & & x_{33} & \cdots & \cdots & \cdots & x_{3m} \\ & & & \ddots & \cdots & \cdots & \vdots \\ & & & & x_{kk} & \cdots & \vdots \\ & & & & x_{k+1,k} & \cdots & x_{k+1,m} \\ & & & & \vdots & \vdots & \vdots \\ & & & & x_{n,k} & \cdots & x_{n,m} \end{pmatrix}.$$

Orthogonal Vectors and Subspaces (Cont.)

The Householder *reflectors* algorithm

This matrix is upper triangular up to column number $k-1$. To advance by one step, it must be transformed into one which is upper triangular up the k-th column, leaving the previous columns in the same form. To leave the first $k-1$ columns unchanged, select a w vector which has zeros in positions 1 through $k-1$. So the next Householder reflector matrix is defined as

$$P_k = I - 2w_k w_k^T, \tag{1.23}$$

in which the vector w_k is defined as

$$w_k = \frac{z}{\|z\|_2}, \tag{1.24}$$

where the components of the vector z are given by

$$z_i = \begin{cases} 0 & \text{if} \quad i < k \\ \beta + x_{ii} & \text{if} \quad i = k \\ x_{ik} & \text{if} \quad i > k \end{cases} \tag{1.25}$$

with

$$\beta = \text{sign}(x_{kk}) \times \left(\sum_{i=k}^{n} x_{ik}^2 \right)^{1/2}. \tag{1.26}$$

Orthogonal Vectors and Subspaces (Cont.)

The Householder *reflectors* algorithm

We note in passing that the premultiplication of a matrix X by a Householder transform requires only a rank-one update since,

$$(I - 2ww^T)X = X - wv^T \quad \text{where} \quad v = 2X^T w.$$

Therefore, the Householder matrices need not, and should not, be explicitly formed. In addition, the vectors w need not be explicitly scaled.

Assume now that $m-1$ Householder transforms have been applied to a certain matrix X of dimension $n \times m$, to reduce it into the upper triangular form,

$$X_m \equiv P_{m-1}P_{m-2}\ldots P_1 X = \begin{pmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1m} \\ & x_{22} & x_{23} & \cdots & x_{2m} \\ & & x_{33} & \cdots & x_{3m} \\ & & & \ddots & \vdots \\ & & & & x_{m,m} \\ & & & & 0 \\ & & & & \vdots \\ & & & & \vdots \end{pmatrix}. \qquad (1.27)$$

Orthogonal Vectors and Subspaces (Cont.)

The Householder *reflectors* algorithm

Recall that our initial goal was to obtain a QR factorization of X. We now wish to recover the Q and R matrices from the P_k's and the above matrix. If we denote by P the product of the P_i on the left-side of (1.27), then (1.27) becomes

$$PX = \begin{pmatrix} R \\ O \end{pmatrix}, \tag{1.28}$$

in which R is an $m \times m$ upper triangular matrix, and O is an $(n-m) \times m$ zero block. Since P is unitary, its inverse is equal to its transpose and, as a result,

$$X = P^T \begin{pmatrix} R \\ O \end{pmatrix} = P_1 P_2 \dots P_{m-1} \begin{pmatrix} R \\ O \end{pmatrix}.$$

If E_m is the matrix of size $n \times m$ which consists of the first m columns of the identity matrix, then the above equality translates into

$$X = P^T E_m R.$$

Orthogonal Vectors and Subspaces (Cont.)

The Householder *reflectors* algorithm

The matrix $Q = P^T E_m$ represents the m first columns of P^T. Since

$$Q^T Q = E_m^T P P^T E_m = I,$$

Q and R are the matrices sought. In summary,

$$X = QR,$$

in which R is the triangular matrix obtained from the Householder reduction of X (see (1.27) and (1.28)) and

$$Qe_j = P_1 P_2 \dots P_{m-1} e_j.$$

Orthogonal Vectors and Subspaces (Cont.)

The Householder *reflectors* algorithm

ALGORITHM **1.3** *Householder Orthogonalization*

1. *Define* $X = [x_1, \ldots, x_m]$
2. *For* $k = 1, \ldots, m$ *Do:*
3. *If* $k > 1$ *compute* $r_k := P_{k-1} P_{k-2} \ldots P_1 x_k$
4. *Compute* w_k *using (1.24), (1.25), (1.26)*
5. *Compute* $r_k := P_k r_k$ *with* $P_k = I - 2 w_k w_k^T$
6. *Compute* $q_k = P_1 P_2 \ldots P_k e_k$
7. *EndDo*

Note that line 6 can be omitted since the q_i are not needed in the execution of the next steps. It must be executed only when the matrix Q is needed at the completion of the algorithm. Also, the operation in line 5 consists only of zeroing the components $k+1, \ldots, n$ and updating the k-th component of r_k. In practice, a work vector can be used for r_k and its nonzero components after this step can be saved into an upper triangular matrix. Since the components 1 through k of the vector w_k are zero, the upper triangular matrix R can be saved in those zero locations which would otherwise be unused.

Canonical Forms of Matrices

This section discusses the reduction of square matrices into matrices that have simpler forms, such as diagonal, bidiagonal, or triangular. Reduction means a transformation that preserves the eigenvalues of a matrix.

Definition 1.5 *Two matrices A and B are said to be similar if there is a nonsingular matrix X such that*

$$A = XBX^{-1}.$$

The mapping $B \rightarrow A$ is called a similarity transformation.

It is clear that *similarity* is an equivalence relation. Similarity transformations preserve the eigenvalues of matrices. An eigenvector u_B of B is transformed into the eigenvector $u_A = Xu_B$ of A. In effect, a similarity transformation amounts to representing the matrix B in a different basis.

We now introduce some terminology.

1. An eigenvalue λ of A has *algebraic multiplicity* μ, if it is a root of multiplicity μ of the characteristic polynomial.

2. If an eigenvalue is of algebraic multiplicity one, it is said to be *simple*. A nonsimple eigenvalue is *multiple*.

Canonical Forms of Matrices

3. The *geometric multiplicity* γ of an eigenvalue λ of A is the maximum number of independent eigenvectors associated with it. In other words, the geometric multiplicity γ is the dimension of the eigenspace Null $(A - \lambda I)$.

4. A matrix is *derogatory* if the geometric multiplicity of at least one of its eigenvalues is larger than one.

5. An eigenvalue is *semisimple* if its algebraic multiplicity is equal to its geometric multiplicity. An eigenvalue that is not semisimple is called *defective*.

Often, $\lambda_1, \lambda_2, \ldots, \lambda_p$ $(p \leq n)$ are used to denote the *distinct* eigenvalues of A. It is easy to show that the characteristic polynomials of two similar matrices are identical; see Exercise 9. Therefore, the eigenvalues of two similar matrices are equal and so are their algebraic multiplicities. Moreover, if v is an eigenvector of B, then Xv is an eigenvector of A and, conversely, if y is an eigenvector of A then $X^{-1}y$ is an eigenvector of B. As a result the number of independent eigenvectors associated with a given eigenvalue is the same for two similar matrices, i.e., their geometric multiplicity is also the same.

Linear Independence of Eigenvectors

Most matrices have an ample supply of eigenvectors

Theorem *Let $A \in \mathbb{C}^{n\times n}$, let $\lambda_1, \ldots, \lambda_k$ be distinct eigenvalues of A, and let $v_1, \ldots, v_k$ be eigenvectors associated with $\lambda_1, \ldots, \lambda_k$, respectively. Then $v_1, \ldots, v_k$ are linearly independent.*

Corollary *If $A \in \mathbb{C}^{n\times n}$ has n distinct eigenvalues, then A has a set of n linearly independent eigenvectors $v_1, \ldots, v_n$. In other words, there is a basis of $\mathbb{C}^n$ consisting of eigenvectors of A.*

A matrix $A \in \mathbb{C}^{n\times n}$ that has n linearly independent eigenvectors is called *semisimple*. A synonym for semisimple is *diagonalizable*.

Corollary states that every matrix that has distinct eigenvalues is semisimple. The converse is false; a matrix can have repeated eigenvalues and still be semisimple. A good example is the matrix I, which has the eigenvalue 1, repeated n times. Since every nonzero vector is an eigenvector of I, any basis of $\mathbb{C}^n$ is a set of n linearly independent eigenvectors of I. Thus I is semisimple. Another good example is the matrix 0.

A matrix that is not semisimple is called *defective*. An example is

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

which is upper triangular and has the eigenvalues 0 and 0. The eigenspace associated with 0 is one-dimensional, so A does not have two linearly independent eigenvectors.

REDUCTION TO THE DIAGONAL FORM

DIAGONALIZABLE MATRICES

Reduction to the Diagonal Form

Diagonalizable matrices

The simplest form in which a matrix can be reduced is undoubtedly the diagonal form. Unfortunately, this reduction is not always possible. A matrix that can be reduced to the diagonal form is called *diagonalizable*. The following theorem characterizes such matrices.

Theorem 1.6 *A matrix of dimension n is diagonalizable if and only if it has n linearly independent eigenvectors.*

Proof. A matrix A is diagonalizable if and only if there exists a nonsingular matrix X and a diagonal matrix D such that $A = XDX^{-1}$, or equivalently $AX = XD$, where D is a diagonal matrix. This is equivalent to saying that n linearly independent vectors exist — the n column-vectors of X — such that $Ax_i = d_i x_i$. Each of these column-vectors is an eigenvector of A. □

Reduction to the Diagonal Form

Diagonalizable matrices

A matrix that is diagonalizable has only semisimple eigenvalues. Conversely, if all the eigenvalues of a matrix A are semisimple, then A has n eigenvectors. It can be easily shown that these eigenvectors are linearly independent.

We have the following proposition.

Proposition 1.7 *A matrix is diagonalizable if and only if all its eigenvalues are semisimple.*

Since every simple eigenvalue is semisimple, an immediate corollary of the above result is: When A has n distinct eigenvalues, then it is diagonalizable.

Remark: *What are the eigenvectors of the* 2×2 *zero matrix ? The eigenvectors are clearly* $[1\ 0]^\top$ *and* $[0\ 1]^\top$ *(and any multiple of these). The particular* 2×2 *zero matrix is considered for simplicity.*

The Jordan Canonical Form

From the theoretical viewpoint, one of the most important canonical forms of matrices is the well known Jordan form. A full development of the steps leading to the Jordan form is beyond the scope of this book. Only the main theorem is stated. Details, including the proof, can be found in standard books of linear algebra.

In the following, m_i refers to the algebraic multiplicity of the individual eigenvalue λ_i and l_i is the *index* of the eigenvalue, i.e., the smallest integer for which $\text{Null}(A - \lambda_i I)^{l_i+1} = \text{Null}(A - \lambda_i I)^{l_i}$.

Theorem 1.8 *Any matrix A can be reduced to a block diagonal matrix consisting of p diagonal blocks, each associated with a distinct eigenvalue λ_i. Each of these diagonal blocks has itself a block diagonal structure consisting of γ_i sub-blocks, where γ_i is the geometric multiplicity of the eigenvalue λ_i. Each of the sub-blocks, referred to as a Jordan block, is an upper bidiagonal matrix of size not exceeding $l_i \leq m_i$, with the constant λ_i on the diagonal and the constant one on the super diagonal.*

Remark: For more details on this topic, see e.g., [**1**] *Paul R. Halmos, Finite-Dimensional Vector Spaces, Springer Verlag, New York, 1958.* or [**2**] *Kenneth Hoffman and Ray Kunze. Linear algebra, 2nd ed, Englewood Cliffs, NJ, Prentice-Hall (1971).*

The Jordan Canonical Form (Cont.)

The i-th diagonal block, $i = 1, \ldots, p$, is known as the i-th Jordan submatrix (sometimes "Jordan Box"). The Jordan submatrix number i starts in column $j_i \equiv m_1 + m_2 + \cdots + m_{i-1} + 1$. Thus,

$$X^{-1}AX = J = \begin{pmatrix} J_1 & & & & & \\ & J_2 & & & & \\ & & \ddots & & & \\ & & & J_i & & \\ & & & & \ddots & \\ & & & & & J_p \end{pmatrix},$$

where each J_i is associated with λ_i and is of size m_i the algebraic multiplicity of λ_i. It has itself the following structure,

$$J_i = \begin{pmatrix} J_{i1} & & & \\ & J_{i2} & & \\ & & \ddots & \\ & & & J_{i\gamma_i} \end{pmatrix} \text{ with } J_{ik} = \begin{pmatrix} \lambda_i & 1 & & \\ & \ddots & \ddots & \\ & & \lambda_i & 1 \\ & & & \lambda_i \end{pmatrix}.$$

Each of the blocks J_{ik} corresponds to a different eigenvector associated with the eigenvalue λ_i. Its size l_i is the index of λ_i.

The Schur Canonical Form

Here, it will be shown that any matrix is unitarily similar to an upper triangular matrix. The only result needed to prove the following theorem is that any vector of 2-norm one can be completed by $n-1$ additional vectors to form an orthonormal basis of $\mathbb{C}^n$.

Theorem 1.9 *For any square matrix A, there exists a unitary matrix Q such that*

$$Q^H A Q = R$$

is upper triangular.

Remark: Not all matrices are diagonalizable, but we can transform any square matrix into triangular form by means of a unitary (or orthogonal) similarity. This is the consequence of the Schur theorem.

The Schur Canonical Form

Theorem 1.9 *For any square matrix A, there exists a unitary matrix Q such that*

$$Q^H A Q = R$$

is upper triangular.

Some Remarks on Schur canonical form

1) Notice that the Schur form is not unique, because the eigenvalues may appear on the diagonal of R in any order.
2) This introduces complex numbers even when A is real. When A is real, we prefer a canonical form that uses only real numbers, because it will be cheaper to compute.
3) This means that we will have to sacrifice a triangular canonical form and settle for a block-triangular canonical form.

The Schur Canonical Form (proof)

Proof. The proof is by induction over the dimension n. The result is trivial for $n = 1$. Assume that it is true for $n - 1$ and consider any matrix A of size n. The matrix admits at least one eigenvector u that is associated with an eigenvalue λ. Also assume without loss of generality that $\|u\|_2 = 1$. First, complete the vector u into an orthonormal set, i.e., find an $n \times (n-1)$ matrix V such that the $n \times n$ matrix $U = [u, V]$ is unitary. Then $AU = [\lambda u, AV]$ and hence,

$$U^H A U = \begin{bmatrix} u^H \\ V^H \end{bmatrix} [\lambda u, AV] = \begin{pmatrix} \lambda & u^H A V \\ 0 & V^H A V \end{pmatrix}. \tag{1.29}$$

Now use the induction hypothesis for the $(n-1) \times (n-1)$ matrix $B = V^H A V$: There exists an $(n-1) \times (n-1)$ unitary matrix Q_1 such that $Q_1^H B Q_1 = R_1$ is upper triangular. Define the $n \times n$ matrix

$$\hat{Q}_1 = \begin{pmatrix} 1 & 0 \\ 0 & Q_1 \end{pmatrix}$$

and multiply both members of (1.29) by $\hat{Q}_1^H$ from the left and $\hat{Q}_1$ from the right. The resulting matrix is clearly upper triangular and this shows that the result is true for A, with $Q = \hat{Q}_1 U$ which is a unitary $n \times n$ matrix. □

The Schur Canonical Form (proof)

How people actually find it ? (Francis 1961, Kublanovskaja 1962), See book David S. Watkins (3ed,2010)

1) First, notice that this proof is not constructive as we assume the knowledge of an eigenpair (λ, u)
2) It bears noting that in practice, people do not use repeated Gram-Schmidt to find this Schur decomposition!
3) Schur decomposition of a given matrix is known to be numerically computed by QR algorithm or its variants.
4) In other words, the roots of the characteristic polynomial corresponding to the matrix are not necessarily computed ahead in order to obtain its Schur decomposition.
5) Conversely, *QR* algorithm can be used to compute the roots of any given characteristic polynomial by finding the Schur decomposition of its companion matrix.
6) We conclude by pointing out that the Schur (and the quasi-Schur) form of a given matrix are in no way unique!

LEAST SQUARE PROBLEM

QR Factorization and Singular Value Decomposition (SVD)

Least Squares Problem

Solution of Least Squares Problems

- More robust approach is to use QR factorization $A = \hat{Q}\hat{R}$
  - b can be projected onto range(A) by $P = \hat{Q}\hat{Q}^T$, and therefore $\hat{Q}\hat{R}x = \hat{Q}\hat{Q}^T b$
  - Left-multiply by $\hat{Q}^T$ and we get $\hat{R}x = \hat{Q}^T b$ (note $A^+ = \hat{R}^{-1}\hat{Q}^T$)

> Least squares via QR Factorization
>
> Compute reduced QR factorization $A = \hat{Q}\hat{R}$
> Compute vector $c = \hat{Q}^T b$
> Solve upper-triangular system $\hat{R}x = c$ for x

- Computation is dominated by QR factorization ($2mn^2 - \frac{2}{3}n^3$)
- Question: If Householder QR is used, how to compute $\hat{Q}^T b$?
- Answer: Compute $Q^T b$ (where Q is from full QR factorization) and then take first n entries of resulting $Q^T b$

Least Squares Problem

Solution of Least Squares Problems

- For a QR factorization $A = QR$ computed by Householder triangularization, the factors $\tilde{Q}$ and $\tilde{R}$ satisfy

$$\tilde{Q}\tilde{R} = A + \delta A, \quad \|\delta A\|/\|A\| = O(\epsilon_{\text{machine}}),$$

  i.e., exact QR factorization of a slightly perturbed A
- $\tilde{R}$ is R computed by algorithm using floating points
- However, $\tilde{Q}$ is product of *exactly orthogonal* reflectors

$$\tilde{Q} = \tilde{Q}_1 \tilde{Q}_2 \dots \tilde{Q}_n$$

  where $\tilde{Q}_k$ is given by computed $\tilde{v}_k$, since Q is not formed explicitly

Backward Stability of Solving $Ax = b$ with QR

Least Squares Problems

Algorithm: Solving $Ax = b$ by QR Factorization

Compute $A = QR$ using Householder, represent Q by reflectors
Compute vector $y = Q^T b$ implicitly using reflectors
Solve upper-triangular system $Rx = y$ for x

- All three steps are backward stable
- Overall, we can show that

$$(A + \Delta A)\tilde{x} = b, \quad \|\Delta A\|/\|A\| = O(\epsilon_{\text{machine}})$$

as we prove next

Proof of Backward Stability

Backward Stability of Solving $Ax = b$ with Householder QR

Proof: Step 2 gives

$$(\tilde{Q} + \delta Q)\tilde{y} = b, \quad \|\delta Q\| = O(\epsilon_{\text{machine}})$$

Step 3 gives

$$(\tilde{R} + \delta R)\tilde{x} = \tilde{y}, \quad \|\delta R\|/\|\tilde{R}\| = O(\epsilon_{\text{machine}})$$

Therefore,

$$b = (\tilde{Q} + \delta Q)(\tilde{R} + \delta R)\tilde{x} = \left[\tilde{Q}\tilde{R} + (\delta Q)\tilde{R} + \tilde{Q}(\delta R) + (\delta Q)(\delta R)\right]\tilde{x}$$

Step 1 gives

$$b = \left[A + \underbrace{\delta A + (\delta Q)\tilde{R} + \tilde{Q}(\delta R) + (\delta Q)(\delta R)}_{\Delta A}\right]\tilde{x}$$

where $\tilde{Q}\tilde{R} = A + \delta A$

Proof of Backward Stability (Cont.)

Backward Stability of Solving $Ax = b$ with Householder QR

$\tilde{Q}\tilde{R} = A + \delta A$ where $\|\delta A\|/\|A\| = O(\epsilon_{\text{machine}})$, and therefore

$$\frac{\|\tilde{R}\|}{\|A\|} \leq \|\tilde{Q}^T\| \frac{\|A + \delta A\|}{\|A\|} = O(1)$$

Now show that each term in ΔA is small

$$\frac{\|(\delta Q)\tilde{R}\|}{\|A\|} \leq \|(\delta Q)\| \frac{\|\tilde{R}\|}{\|A\|} = O(\epsilon_{\text{machine}})$$

$$\frac{\|\tilde{Q}(\delta R)\|}{\|A\|} \leq \|\tilde{Q}\| \frac{\|\delta R\|}{\|\tilde{R}\|} \frac{\|\tilde{R}\|}{\|A\|} = O(\epsilon_{\text{machine}})$$

$$\frac{\|(\delta Q)(\delta R)\|}{\|A\|} \leq \|\delta Q\| \frac{\|\delta R\|}{\|A\|} = O(\epsilon^2_{\text{machine}})$$

Overall,

$$\frac{\|\Delta A\|}{\|A\|} \leq \frac{\|\delta A\|}{\|A\|} + \frac{\|(\delta Q)\tilde{R}\|}{\|A\|} + \frac{\|\tilde{Q}(\delta R)\|}{\|A\|} + \frac{\|(\delta Q)(\delta R)\|}{\|A\|} = O(\epsilon_{\text{machine}})$$

Since the algorithm is backward stable, it is also accurate.

Stability of Gram-Schmidt Orthogonalization

- Gram-Schmidt QR is unstable, due to loss of orthogonality
- Gram-Schmidt can be stabilized using augmented system of equations
  1. Compute QR factorization of augmented matrix: [Q,R1]=mgs([A,b])
  2. Extract R and $\hat{Q}^T b$ from $R1$: R=R1(1:n,1:n); Qb=R1(1:n,n+1)
  3. Back solve: x=R\Qb

Theorem

The solution of the full-rank least squares problem by Gram-Schmidt orthogonality is backward stable in the sense that the computed solution $\tilde{x}$ has the property

$$\|(A+\delta A)\tilde{x}-b)\| = \min, \quad \frac{\|\delta A\|}{\|A\|} = O(\epsilon_{machine})$$

for some $\delta A \in \mathbb{R}^{m\times n}$, provided that $\hat{Q}^T b$ is formed implicitly.

The method of normal equations

- The method of *normal equation* solves $x = (A^T A)^{-1} A^T b$, due to squaring of condition number of A

Theorem

The solution of the full-rank least squares problem via normal equation is unstable. Stability can be achieved, however, by restriction to a class of problems in which $\kappa(A)$ is uniformly bounded above.

- Another method is to SVD (coming up next)

Summary of Algorithms for Least Square problems

- Householder QR (with/without pivoting, explicit or implicit Q): **Backward stable**
- Classical Gram-Schmidt: **Unstable**
- Modified Gram-Schmidt with explicit Q: **Unstable**
- Modified Gram-Schmidt with augmented system of equations with implicit Q: **Backward stable**
- Normal equations (solve $A^T Ax = A^T b$): **Very unstable**
- Singular value decomposition: **Backward stable**

Singular Value Decomposition (SVD)

- The image of unit sphere under any $m \times n$ matrix is a *hyperellipsoid*
- Give a unit sphere S in $\mathbb{R}^n$, let AS denote the shape after transformation
- SVD is

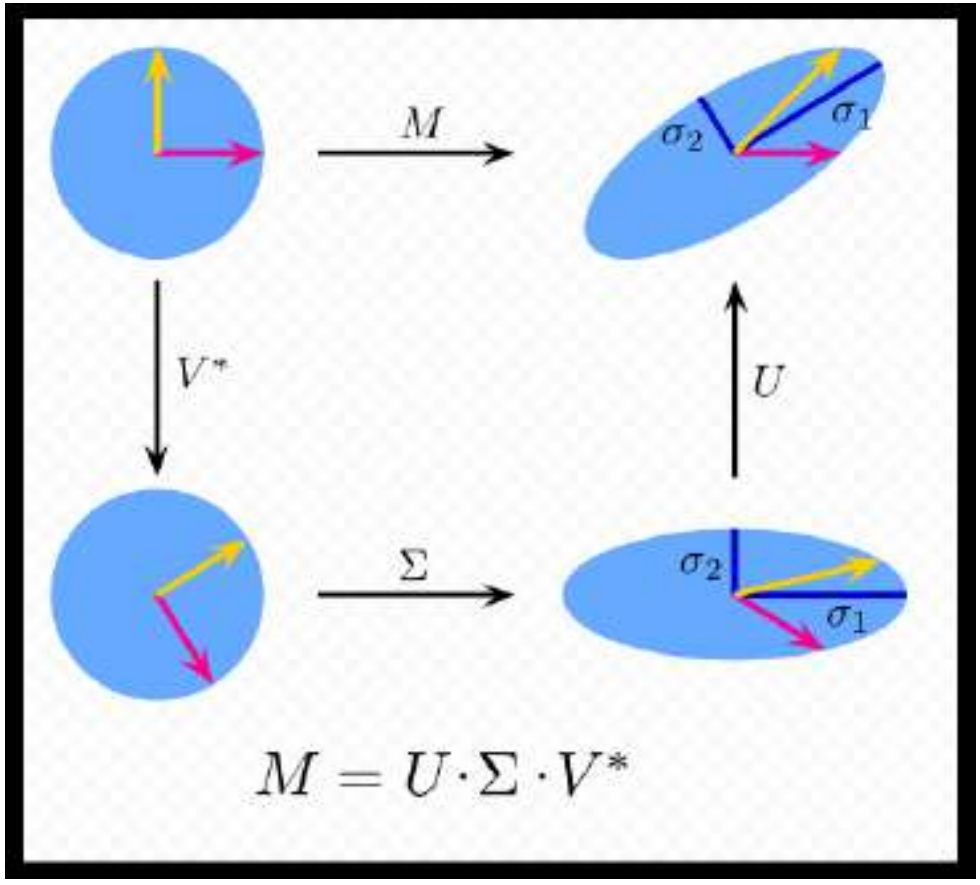
$$A = U\Sigma V^T$$

  where $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ is orthogonal and $\Sigma \in \mathbb{R}^{m \times n}$ is diagonal
- *Singular values* are diagonal entries of Σ, correspond to the principal semiaxes, with entries $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$.
- *Left singular vectors* of A are column vectors of U and are oriented in the directions of the principal semiaxes of AS
- *Right singular vectors* of A are column vectors of V and are the preimages of the principal semiaxes of AS
- $Av_j = \sigma_j u_j$ for $1 \leq j \leq n$

Some remarks:

- $A \in \mathbb{C}^{n \times n}$, $U \in \mathbb{C}^{n \times n}$, $\Sigma \in \mathbb{C}^{n \times n}$, $V^* \in \Re^{n \times n}$.
- Note that the diagonal matrix Σ has the same shape as A even when A is not square, but U and V^* are always square unitary matrices.
- The singular values of a matrix A are precisely the lengths of the semi-axes of the hyperellipsoid E defined by $E = \{Ax : ||x||_2 = 1\}$.

SVD[3] (Geometric Observation)



The image shows:

Upper Left: The unit disc with the two canonical unit vectors.

Upper Right: Unit disc transformed with M and singular Values σ_1 and σ_2 indicated

Lower Left: The action of V^* on the unit disc. This is just a rotation.

Lower Right: The action of ΣV^* on the unit disc. Sigma scales in vertically and horizontally.

[3] thanks to wikipedia (image)

Two Different Types of SVD

- **Full SVD**: $U \in \mathbb{R}^{m \times m}$, $\Sigma \in \mathbb{R}^{m \times n}$, and $V \in \mathbb{R}^{n \times n}$ is

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

- **Reduced SVD**: $\hat{U} \in \mathbb{R}^{m \times n}$, $\hat{\Sigma} \in \mathbb{R}^{n \times n}$ (assume $m \geq n$)

$$A = \hat{U} \hat{\Sigma} V^T$$

- Furthermore, notice that

$$A = \sum_{i=1}^{\min\{m,n\}} \sigma_i u_i v_i^T$$

so we can keep only entries of U and V corresponding to nonzero σ_i.

SVD versus Eigenvalue Decomposition

- Eigenvalue decomposition of nondefective matrix A is $A = X\Lambda X^{-1}$
- Differences between SVD and Eigenvalue Decomposition
  - Not every matrix has eigenvalue decomposition, but every matrix has singular value decomposition
  - Eigenvalues may not always be real numbers, but singular values are always non-negative real numbers
  - Eigenvectors are not always orthogonal to each other (orthogonal for symmetric matrices), but left (or right) singular vectors are orthogonal to each other
- Similarities
  - Singular values of A are square roots of eigenvalues of AA^T and A^TA, and their eigenvectors are left and right singular vectors, respectively
  - Singular values of symmetric matrices are absolute values of eigenvalues, and eigenvectors are singular vectors
  - This relationship can be used to compute singular values by hand

Existence of SVD (sketch of the proof)

Theorem

(Existence) Every matrix $A \in \mathbb{R}^{m \times n}$ has an SVD.

Proof: Let $\sigma_1 = \|A\|_2$. There exists $v_1 \in \mathbb{R}^n$ with $\|v_1\|_2 = 1$ and $\|Av_1\|_2 = \sigma_1$. Let U_1 and V_1 be orthogonal matrices whose first columns are $u_1 = Av_1/\sigma_1$ (or any unit-length vector if $\sigma_1 = 0$) and v_1, respectively. Note that

$$U_1^T A V_1 = S = \begin{bmatrix} \sigma_1 & \omega^T \\ 0 & B \end{bmatrix}. \tag{1}$$

Furthermore, $\omega = 0$ because $\|S\|_2 = \sigma_1$, and

$$\left\| \begin{bmatrix} \sigma_1 & \omega^T \\ 0 & B \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \omega \end{bmatrix} \right\|_2 \geq \sigma_1^2 + \omega^T\omega = \sqrt{\sigma_1^2 + \omega^T\omega} \left\| \begin{bmatrix} \sigma_1 \\ \omega \end{bmatrix} \right\|_2,$$

implying that $\sigma_1 \geq \sqrt{\sigma_1^2 + \omega^T\omega}$ and $\omega = 0$.

Existence of SVD (Cont.)

We then prove by induction using (1). If $m = 1$ or $n = 1$, then B is empty and we have $A = U_1 S V_1^T$. Otherwise, suppose $B = U_2 \Sigma_2 V_2^T$, and then

$$A = \underbrace{U_1 \begin{bmatrix} 1 & 0^T \\ 0 & U_2 \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} \sigma_1 & 0^T \\ 0 & \Sigma_2 \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} 1 & 0^T \\ 0 & V_2^T \end{bmatrix} V_1^T}_{V^T},$$

where U and V are orthogonal.

Existence and Uniqueness (Theorem): Every matrix $A \in \mathcal{C}^{n \times n}$ has a singular value decomposition $A = U \Sigma V^*$. Furthermore, the singular values $\{\sigma_i\}$ are uniquely determined, and, if A is square and the σ_i are distinct, the left and right singular vectors $\{u_i\}$ and $\{v_i\}$ are uniquely determined up to complex signs (i.e., complex scalar factors of absolute value 1). See detilas ain *Gene H. Golub and Charles F. Van Loan. Matrix computations, 3rd ed., Johns Hopkins University Press (1996)*.

Some results of linear algebra behind the iterative methods

Some results of linear algebra behind the iterative methods

In what follows we will briefly highlight some results that are important in the study of convergence of iterative methods.

The interested reader is referred to the below list of references for further information linked to the subject of numerical linear algebra as well as detailed proofs.

Powers of Matrices

Theorem 1.10 *The sequence A^k, $k = 0, 1, \ldots$, converges to zero if and only if $\rho(A) < 1$.*

Proof. To prove the necessary condition, assume that $A^k \rightarrow 0$ and consider u_1 a unit eigenvector associated with an eigenvalue λ_1 of maximum modulus. We have

$$A^k u_1 = \lambda_1^k u_1,$$

which implies, by taking the 2-norms of both sides,

$$|\lambda_1^k| = \|A^k u_1\|_2 \rightarrow 0.$$

This shows that $\rho(A) = |\lambda_1| < 1$.

The Jordan canonical form must be used to show the sufficient condition. Assume that $\rho(A) < 1$. Start with the equality

$$A^k = X J^k X^{-1}.$$

Powers of Matrices

To prove that A^k converges to zero, it is sufficient to show that J^k converges to zero. An important observation is that J^k preserves its block form. Therefore, it is sufficient to prove that each of the Jordan blocks converges to zero. Each block is of the form

$$J_i = \lambda_i I + E_i$$

where E_i is a nilpotent matrix of index l_i, i.e., $E_i^{l_i} = 0$. Therefore, for $k \geq l_i$,

$$J_i^k = \sum_{j=0}^{l_i-1} \frac{k!}{j!(k-j)!} \lambda_i^{k-j} E_i^j.$$

Using the triangle inequality for any norm and taking $k \geq l_i$ yields

$$\|J_i^k\| \leq \sum_{j=0}^{l_i-1} \frac{k!}{j!(k-j)!} |\lambda_i|^{k-j} \|E_i^j\|.$$

Since $|\lambda_i| < 1$, each of the terms in this *finite* sum converges to zero as $k \to \infty$. Therefore, the matrix J_i^k converges to zero. □

An equally important result is stated in the following theorem.

Powers of Matrices

Theorem 1.11 *The series*

$$\sum_{k=0}^{\infty} A^k$$

converges if and only if $\rho(A) < 1$. *Under this condition,* $I - A$ *is nonsingular and the limit of the series is equal to* $(I - A)^{-1}$.

Proof. The first part of the theorem is an immediate consequence of Theorem 1.10. Indeed, if the series converges, then $\|A^k\| \to 0$. By the previous theorem, this implies that $\rho(A) < 1$. To show that the converse is also true, use the equality

$$I - A^{k+1} = (I - A)(I + A + A^2 + \ldots + A^k)$$

and exploit the fact that since $\rho(A) < 1$, then $I - A$ is nonsingular, and therefore,

$$(I - A)^{-1}(I - A^{k+1}) = I + A + A^2 + \ldots + A^k.$$

This shows that the series converges since the left-hand side will converge to $(I - A)^{-1}$. In addition, it also shows the second part of the theorem. □

Powers of Matrices - Jordan canonical form

Another important consequence of the Jordan canonical form is a result that relates the spectral radius of a matrix to its matrix norm.

Theorem 1.12 *For any matrix norm* $\|.\|$, *we have*

$$\lim_{k\to\infty} \|A^k\|^{1/k} = \rho(A).$$

Proof. The proof is a direct application of the Jordan canonical form.

Normal and Hermitian Matrices

Normal matrices

Proof. The proof is by induction over the dimension n. The result is trivial for $n = 1$. Assume that it is true for $n - 1$ and consider any matrix A of size n. The matrix admits at least one eigenvector u that is associated with an eigenvalue λ. Also assume without loss of generality that $\|u\|_2 = 1$. First, complete the vector u into an orthonormal set, i.e., find an $n \times (n-1)$ matrix V such that the $n \times n$ matrix $U = [u, V]$ is unitary. Then $AU = [\lambda u, AV]$ and hence,

$$U^H AU = \begin{bmatrix} u^H \\ V^H \end{bmatrix} [\lambda u, AV] = \begin{pmatrix} \lambda & u^H AV \\ 0 & V^H AV \end{pmatrix}. \tag{1.29}$$

Now use the induction hypothesis for the $(n-1) \times (n-1)$ matrix $B = V^H AV$: There exists an $(n-1) \times (n-1)$ unitary matrix Q_1 such that $Q_1^H B Q_1 = R_1$ is upper triangular. Define the $n \times n$ matrix

$$\hat{Q}_1 = \begin{pmatrix} 1 & 0 \\ 0 & Q_1 \end{pmatrix}$$

and multiply both members of (1.29) by $\hat{Q}_1^H$ from the left and $\hat{Q}_1$ from the right. The resulting matrix is clearly upper triangular and this shows that the result is true for A, with $Q = \hat{Q}_1 U$ which is a unitary $n \times n$ matrix. □

Normal and Hermitian Matrices

Normal Matrices

This section examines specific properties of normal matrices and Hermitian matrices, including some optimality properties related to their spectra. The most common normal matrices that arise in practice are Hermitian or skew-Hermitian.

1.9.1 Normal Matrices

By definition, a matrix is said to be normal if it commutes with its transpose conjugate, i.e., if it satisfies the relation

$$A^H A = A A^H. \tag{1.30}$$

An immediate property of normal matrices is stated in the following lemma.

Normal and Hermitian Matrices

Normal Matrices

Lemma 1.13 *If a normal matrix is triangular, then it is a diagonal matrix.*

Proof. Assume, for example, that A is upper triangular and normal. Compare the first diagonal element of the left-hand side matrix of (1.30) with the corresponding element of the matrix on the right-hand side. We obtain that

$$|a_{11}|^2 = \sum_{j=1}^{n} |a_{1j}|^2,$$

which shows that the elements of the first row are zeros except for the diagonal one. The same argument can now be used for the second row, the third row, and so on to the last row, to show that $a_{ij} = 0$ for $i \neq j$. □

A consequence of this lemma is the following important result.

Normal and Hermitian Matrices

Normal Matrices

Theorem 1.14 *A matrix is normal if and only if it is unitarily similar to a diagonal matrix.*

Proof. It is straightforward to verify that a matrix which is unitarily similar to a diagonal matrix is normal. We now prove that any normal matrix A is unitarily similar to a diagonal matrix. Let $A = QRQ^H$ be the Schur canonical form of A where Q is unitary and R is upper triangular. By the normality of A,

$$QR^HQ^HQRQ^H = QRQ^HQR^HQ^H$$

or,

$$QR^HRQ^H = QRR^HQ^H.$$

Upon multiplication by Q^H on the left and Q on the right, this leads to the equality $R^HR = RR^H$ which means that R is normal, and according to the previous lemma this is only possible if R is diagonal. □

Normal and Hermitian Matrices

Normal Matrices

Thus, any normal matrix is diagonalizable and admits an orthonormal basis of eigenvectors, namely, the column vectors of Q.

The following result will be used in a later chapter. The question that is asked is: Assuming that any eigenvector of a matrix A is also an eigenvector of A^H, is A normal? If A had a full set of eigenvectors, then the result is true and easy to prove. Indeed, if V is the $n \times n$ matrix of common eigenvectors, then $AV = VD_1$ and $A^H V = VD_2$, with D_1 and D_2 diagonal. Then, $AA^H V = VD_1D_2$ and $A^H AV = VD_2D_1$ and, therefore, $AA^H = A^H A$. It turns out that the result is true in general, i.e., independently of the number of eigenvectors that A admits.

Normal and Hermitian Matrices

Normal Matrices

Lemma 1.15 *A matrix A is normal if and only if each of its eigenvectors is also an eigenvector of A^H.*

Proof. If A is normal, then its left and right eigenvectors are identical, so the sufficient condition is trivial. Assume now that a matrix A is such that each of its eigenvectors v_i, $i = 1, \ldots, k$, with $k \leq n$ is an eigenvector of A^H. For each eigenvector v_i of A, $Av_i = \lambda_i v_i$, and since v_i is also an eigenvector of A^H, then $A^H v_i = \mu v_i$. Observe that $(A^H v_i, v_i) = \mu(v_i, v_i)$ and because $(A^H v_i, v_i) = (v_i, Av_i) = \bar{\lambda}_i(v_i, v_i)$, it follows that $\mu = \bar{\lambda}_i$. Next, it is proved by contradiction that there are no elementary divisors. Assume that the contrary is true for λ_i. Then, the first principal vector u_i associated with λ_i is defined by

$$(A - \lambda_i I)u_i = v_i.$$

Taking the inner product of the above relation with v_i, we obtain

$$(Au_i, v_i) = \lambda_i(u_i, v_i) + (v_i, v_i). \tag{1.31}$$

On the other hand, it is also true that

$$(Au_i, v_i) = (u_i, A^H v_i) = (u_i, \bar{\lambda}_i v_i) = \lambda_i(u_i, v_i). \tag{1.32}$$

Clearly, Hermitian matrices are a particular case of normal matrices. Since a normal matrix satisfies the relation $A = QDQ^H$, with D diagonal and Q unitary, the eigenvalues of A are the diagonal entries of D. Therefore, if these entries are real it is clear that $A^H = A$. This is restated in the following corollary.

Corollary 1.16 *A normal matrix whose eigenvalues are real is Hermitian.*

As will be seen shortly, the converse is also true, i.e., a Hermitian matrix has real eigenvalues.

An eigenvalue λ of any matrix satisfies the relation

$$\lambda = \frac{(Au, u)}{(u, u)},$$

where u is an associated eigenvector. Generally, one might consider the complex scalars

$$\mu(x) = \frac{(Ax, x)}{(x, x)}, \tag{1.33}$$

defined for any nonzero vector in $\mathbb{C}^n$. These ratios are known as *Rayleigh quotients* and are important both for theoretical and practical purposes. The set of all possible Rayleigh quotients as x runs over $\mathbb{C}^n$ is called the *field of values* of A. This set is clearly bounded since each $|\mu(x)|$ is bounded by the the 2-norm of A, i.e., $|\mu(x)| \leq \|A\|_2$ for all x.

Normal and Hermitian Matrices

Normal Matrices

If a matrix is normal, then any vector x in $\mathbb{C}^n$ can be expressed as

$$\sum_{i=1}^{n} \xi_i q_i,$$

where the vectors q_i form an orthogonal basis of eigenvectors, and the expression for $\mu(x)$ becomes

$$\mu(x) = \frac{(Ax,x)}{(x,x)} = \frac{\sum_{k=1}^{n} \lambda_k |\xi_k|^2}{\sum_{k=1}^{n} |\xi_k|^2} \equiv \sum_{k=1}^{n} \beta_k \lambda_k, \qquad (1.34)$$

where

$$0 \leq \beta_i = \frac{|\xi_i|^2}{\sum_{k=1}^{n} |\xi_k|^2} \leq 1, \quad \text{and} \quad \sum_{i=1}^{n} \beta_i = 1.$$

From a well known characterization of convex hulls established by Hausdorff (Hausdorff's convex hull theorem), this means that the set of all possible Rayleigh quotients as x runs over all of $\mathbb{C}^n$ is equal to the convex hull of the λ_i's. This leads to the following theorem which is stated without proof.

Normal and Hermitian Matrices

Normal Matrices

Theorem 1.17 *The field of values of a normal matrix is equal to the convex hull of its spectrum.*

The next question is whether or not this is also true for nonnormal matrices and the answer is no: The convex hull of the eigenvalues and the field of values of a nonnormal matrix are different in general. As a generic example, one can take any nonsymmetric real matrix which has real eigenvalues only. In this case, the convex hull of the spectrum is a real interval but its field of values will contain imaginary values. See Exercise 12 for another example. It has been shown (Hausdorff) that the field of values of a matrix is a convex set. Since the eigenvalues are members of the field of values, their convex hull is contained in the field of values. This is summarized in the following proposition.

Proposition 1.18 *The field of values of an arbitrary matrix is a convex set which contains the convex hull of its spectrum. It is equal to the convex hull of the spectrum when the matrix is normal.*

Normal and Hermitian Matrices

Normal Matrices

Theorem 1.17 *The field of values of a normal matrix is equal to the convex hull of its spectrum.*

The next question is whether or not this is also true for nonnormal matrices and the answer is no: The convex hull of the eigenvalues and the field of values of a nonnormal matrix are different in general. As a generic example, one can take any nonsymmetric real matrix which has real eigenvalues only. In this case, the convex hull of the spectrum is a real interval but its field of values will contain imaginary values. See Exercise 12 for another example. It has been shown (Hausdorff) that the field of values of a matrix is a convex set. Since the eigenvalues are members of the field of values, their convex hull is contained in the field of values. This is summarized in the following proposition.

Proposition 1.18 *The field of values of an arbitrary matrix is a convex set which contains the convex hull of its spectrum. It is equal to the convex hull of the spectrum when the matrix is normal.*

Normal and Hermitian Matrices

Normal Matrices

A useful definition based on field of values is that of the *numerical radius*. The numerical radius $\nu(A)$ of an arbitrary matrix A is the radius of the smallest disk containing the field of values, i.e.,

$$\nu(A) = \max_{x \in \mathbb{C}^n} |\mu(x)| .$$

It is easy to see that

$$\rho(A) \leq \nu(A) \leq \|A\|_2 .$$

The spectral radius and numerical radius are identical for normal matrices. It can also be easily shown (see Exercise 21) that $\nu(A) \geq \|A\|_2/2$, which means that

$$\frac{\|A\|_2}{2} \leq \nu(A) \leq \|A\|_2. \tag{1.35}$$

The numerical radius is a vector norm, i.e., it satisfies (1.8–1.10), but it is not consistent. However, it satisfies the power inequality (See HORN AND JOHNSON, 1985):

$$\nu(A^k) \leq \nu(A)^k . \tag{1.36}$$

Normal and Hermitian Matrices

Hermitian Matrices

A first result on Hermitian matrices is the following.

Theorem 1.19 *The eigenvalues of a Hermitian matrix are real, i.e.,* $\sigma(A) \subset \mathbb{R}$.

Proof. Let λ be an eigenvalue of A and u an associated eigenvector of 2-norm unity. Then

$$\lambda = (Au, u) = (u, Au) = \overline{(Au, u)} = \overline{\lambda},$$

which is the stated result. □

Remark: *If, in addition, the matrix is real, then the eigenvectors can be chosen to be real. Since a Hermitian matrix is normal in the above, the following result is a consequence of Theorem 1.14.*

Normal and Hermitian Matrices

Hermitian Matrices

Theorem 1.20 *Any Hermitian matrix is unitarily similar to a real diagonal matrix.*

In particular a Hermitian matrix admits a set of orthonormal eigenvectors that form a basis of $\mathbb{C}^n$.

In the proof of Theorem 1.17 we used the fact that the inner products (Au, u) are real. Generally, it is clear that any Hermitian matrix is such that (Ax, x) is real for any vector $x \in \mathbb{C}^n$. It turns out that the converse is also true, i.e., it can be shown that if (Az, z) is real for all vectors z in $\mathbb{C}^n$, then the matrix A is Hermitian.

Eigenvalues of Hermitian matrices can be characterized by optimality properties of the Rayleigh quotients (1.33). The best known of these is the min-max principle. We now label all the eigenvalues of A in descending order:

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n.$$

Here, the eigenvalues are not necessarily distinct and they are repeated, each according to its multiplicity. In the following theorem, known as the *Min-Max Theorem*, S represents a generic subspace of $\mathbb{C}^n$.

Theorem 1.21 *The eigenvalues of a Hermitian matrix A are characterized by the relation*

$$\lambda_k = \min_{S,\ \dim(S)=n-k+1} \ \max_{x\in S, x\neq 0} \ \frac{(Ax,x)}{(x,x)}. \tag{1.37}$$

Proof. Let $\{q_i\}_{i=1,\ldots,n}$ be an orthonormal basis of $\mathbb{C}^n$ consisting of eigenvectors of A associated with $\lambda_1,\ldots,\lambda_n$ respectively. Let S_k be the subspace spanned by the first k of these vectors and denote by $\mu(S)$ the maximum of $(Ax,x)/(x,x)$ over all nonzero vectors of a subspace S. Since the dimension of S_k is k, a well known theorem of linear algebra shows that its intersection with any subspace S of dimension $n-k+1$ is not reduced to $\{0\}$, i.e., there is vector x in $S\bigcap S_k$. For this $x=\sum_{i=1}^{k}\xi_i q_i$, we have

$$\frac{(Ax,x)}{(x,x)} = \frac{\sum_{i=1}^{k}\lambda_i|\xi_i|^2}{\sum_{i=1}^{k}|\xi_i|^2} \geq \lambda_k \qquad \text{so that } \mu(S) \geq \lambda_k.$$

Consider, on the other hand, the particular subspace S_0 of dimension $n-k+1$ which is spanned by $q_k,\ldots,q_n$. For each vector x in this subspace, we have

$$\frac{(Ax,x)}{(x,x)} = \frac{\sum_{i=k}^{n}\lambda_i|\xi_i|^2}{\sum_{i=k}^{n}|\xi_i|^2} \leq \lambda_k$$

so that $\mu(S_0)\leq\lambda_k$. In other words, as S runs over all the $(n-k+1)$-dimensional subspaces, $\mu(S)$ is always $\geq \lambda_k$ and there is at least one subspace S_0 for which $\mu(S_0)\leq\lambda_k$. This shows the desired result. □

The above result is often called the Courant-Fisher min-max principle or theorem. As a particular case, the largest eigenvalue of A satisfies

$$\lambda_1 = \max_{x \neq 0} \frac{(Ax, x)}{(x, x)}. \tag{1.38}$$

Actually, there are four different ways of rewriting the above characterization. The second formulation is

$$\lambda_k = \max_{S,\ \dim\,(S)=k} \ \min_{x \in S, x \neq 0} \frac{(Ax, x)}{(x, x)} \tag{1.39}$$

and the two other ones can be obtained from (1.37) and (1.39) by simply relabeling the eigenvalues increasingly instead of decreasingly. Thus, with our labeling of the eigenvalues in descending order, (1.39) tells us that the smallest eigenvalue satisfies

$$\lambda_n = \min_{x \neq 0} \frac{(Ax, x)}{(x, x)}, \tag{1.40}$$

with λ_n replaced by λ_1 if the eigenvalues are relabeled increasingly.

In order for all the eigenvalues of a Hermitian matrix to be positive, it is necessary and sufficient that

$$(Ax, x) > 0, \quad \forall\, x \in \mathbb{C}^n, \quad x \neq 0.$$

Such a matrix is called *positive definite*. A matrix which satisfies $(Ax, x) \geq 0$ for any x is said to be *positive semidefinite*. In particular, the matrix $A^H A$ is semipositive definite for any rectangular matrix, since

$$(A^H Ax, x) = (Ax, Ax) \geq 0, \quad \forall\, x.$$

Similarly, AA^H is also a Hermitian semipositive definite matrix. The square roots of the eigenvalues of $A^H A$ for a general rectangular matrix A are called the *singular values* of A and are denoted by σ_i. This is now an obvious fact, because

$$\|A\|_2^2 = \max_{x \neq 0} \frac{\|Ax\|_2^2}{\|x\|_2^2} = \max_{x \neq 0} \frac{(Ax, Ax)}{(x, x)} = \max_{x \neq 0} \frac{(A^H Ax, x)}{(x, x)} = \sigma_1^2$$

which results from (1.38).

Another characterization of eigenvalues, known as the Courant characterization, is stated in the next theorem. In contrast with the min-max theorem, this property is recursive in nature.

Theorem 1.22 *The eigenvalue λ_i and the corresponding eigenvector q_i of a Hermitian matrix are such that*

$$\lambda_1 = \frac{(Aq_1, q_1)}{(q_1, q_1)} = \max_{x \in \mathbb{C}^n, x \neq 0} \frac{(Ax, x)}{(x, x)}$$

and for $k > 1$,

$$\lambda_k = \frac{(Aq_k, q_k)}{(q_k, q_k)} = \max_{x \neq 0, q_1^H x = \ldots = q_{k-1}^H x = 0} \frac{(Ax, x)}{(x, x)}. \tag{1.41}$$

In other words, the maximum of the Rayleigh quotient over a subspace that is orthogonal to the first $k-1$ eigenvectors is equal to λ_k and is achieved for the eigenvector q_k associated with λ_k. The proof follows easily from the expansion (1.34) of the Rayleigh quotient.

Nonnegative Matrices, M-Matrices

Nonnegative matrices play a crucial role in the theory of matrices. They are important in the study of convergence of iterative methods and arise in many applications including economics, queuing theory, and chemical engineering.

A *nonnegative matrix* is simply a matrix whose entries are nonnegative. More generally, a partial order relation can be defined on the set of matrices.

Definition 1.23 *Let A and B be two $n \times m$ matrices. Then*

$$A \leq B$$

if by definition, $a_{ij} \leq b_{ij}$ for $1 \leq i \leq n$, $1 \leq j \leq m$. If O denotes the $n \times m$ zero matrix, then A is nonnegative if $A \geq O$, and positive if $A > O$. Similar definitions hold in which "positive" is replaced by "negative".

The binary relation "$\leq$" imposes only a *partial* order on $\mathbb{R}^{n \times m}$ since two arbitrary matrices in $\mathbb{R}^{n \times m}$ are not necessarily comparable by this relation. For the remainder of this section, we now assume that only square matrices are involved. The next proposition lists a number of rather trivial properties regarding the partial order relation just defined.

Nonnegative Matrices, M-Matrices

Proposition 1.24 *The following properties hold.*

1. *The relation $\leq$ for matrices is reflexive ($A \leq A$), antisymmetric (if $A \leq B$ and $B \leq A$, then $A = B$), and transitive (if $A \leq B$ and $B \leq C$, then $A \leq C$).*
2. *If A and B are nonnegative, then so is their product AB and their sum $A + B$.*
3. *If A is nonnegative, then so is A^k.*
4. *If $A \leq B$, then $A^T \leq B^T$.*
5. *If $O \leq A \leq B$, then $\|A\|_1 \leq \|B\|_1$ and similarly $\|A\|_\infty \leq \|B\|_\infty$.*

Remark: For a proof of the itens in the Proposition 1.24 see *Roger A. Horn and Charles R. Johnson. Matrix analysis, Cambridge, MA, Cambridge University Press (1985).*

Nonnegative Matrices, M-Matrices

A matrix is said to be *reducible* if there is a permutation matrix P such that PAP^T is block upper triangular. Otherwise, it is *irreducible*. An important result concerning nonnegative matrices is the following theorem known as the Perron-Frobenius theorem.

Theorem 1.25 *Let A be a real $n \times n$ nonnegative irreducible matrix. Then $\lambda \equiv \rho(A)$, the spectral radius of A, is a simple eigenvalue of A. Moreover, there exists an eigenvector u with positive elements associated with this eigenvalue.*

A relaxed version of this theorem allows the matrix to be reducible but the conclusion is somewhat weakened in the sense that the elements of the eigenvectors are only guaranteed to be *nonnegative*.

Next, a useful property is established.

Nonnegative Matrices, M-Matrices

Proposition 1.26 *Let A, B, C be nonnegative matrices, with $A \leq B$. Then*

$$AC \leq BC \quad \textit{and} \quad CA \leq CB.$$

Proof. Consider the first inequality only, since the proof for the second is identical. The result that is claimed translates into

$$\sum_{k=1}^{n} a_{ik} c_{kj} \leq \sum_{k=1}^{n} b_{ik} c_{kj}, \quad 1 \leq i, j \leq n,$$

which is clearly true by the assumptions. □

A consequence of the proposition is the following corollary.

Nonnegative Matrices, M-Matrices

Corollary 1.27 *Let A and B be two nonnegative matrices, with $A \leq B$. Then*

$$A^k \leq B^k, \quad \forall\, k \geq 0. \tag{1.42}$$

Proof. The proof is by induction. The inequality is clearly true for $k = 0$. Assume that (1.42) is true for k. According to the previous proposition, multiplying (1.42) from the left by A results in

$$A^{k+1} \leq AB^k. \tag{1.43}$$

Now, it is clear that if $B \geq 0$, then also $B^k \geq 0$, by Proposition 1.24. We now multiply both sides of the inequality $A \leq B$ by B^k to the right, and obtain

$$AB^k \leq B^{k+1}. \tag{1.44}$$

The inequalities (1.43) and (1.44) show that $A^{k+1} \leq B^{k+1}$, which completes the induction proof. □

Proof. The proof is by induction. The inequality is clearly true for $k = 0$. Assume that (1.42) is true for k. According to the previous proposition, multiplying (1.42) from the left by A results in

$$A^{k+1} \leq AB^k. \tag{1.43}$$

Now, it is clear that if $B \geq 0$, then also $B^k \geq 0$, by Proposition 1.24. We now multiply both sides of the inequality $A \leq B$ by B^k to the right, and obtain

$$AB^k \leq B^{k+1}. \tag{1.44}$$

The inequalities (1.43) and (1.44) show that $A^{k+1} \leq B^{k+1}$, which completes the induction proof. □

Nonnegative Matrices, M-Matrices

A theorem which has important consequences on the analysis of iterative methods (e.g. stationary methods linked to boundary value problem, as such the Poisson problem) as well as in the mathematical sciences and applications (e.g., Economics) will now be stated in what follows (see, e.g., also the references below and cited therein):

- Mohamed Abd El Aziz, Wael Khidr, *Nonnegative matrix factorization based on projected hybrid conjugate gradient algorithm*, Signal, Image and Video Processing 9(8) (2015) 1825-1831.
- Abraham Berman and Robert J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences (Classics in Applied Mathematics), SIAM, 1994*.

Proof. The proof is by induction. The inequality is clearly true for $k = 0$. Assume that (1.42) is true for k. According to the previous proposition, multiplying (1.42) from the left by A results in

$$A^{k+1} \leq AB^k. \tag{1.43}$$

Now, it is clear that if $B \geq 0$, then also $B^k \geq 0$, by Proposition 1.24. We now multiply both sides of the inequality $A \leq B$ by B^k to the right, and obtain

$$AB^k \leq B^{k+1}. \tag{1.44}$$

The inequalities (1.43) and (1.44) show that $A^{k+1} \leq B^{k+1}$, which completes the induction proof. □

Nonnegative Matrices, M-Matrices

Theorem 1.28 *Let A and B be two square matrices that satisfy the inequalities*

$$O \leq A \leq B. \tag{1.45}$$

Then

$$\rho(A) \leq \rho(B). \tag{1.46}$$

Proof. The proof is based on the following equality stated in Theorem 1.12

$$\rho(X) = \lim_{k \to \infty} \|X^k\|^{1/k}$$

for any matrix norm. Choosing the $1-$norm, for example, we have from the last property in Proposition 1.24

$$\rho(A) = \lim_{k \to \infty} \|A^k\|_1^{1/k} \leq \lim_{k \to \infty} \|B^k\|_1^{1/k} = \rho(B)$$

which completes the proof. □

Nonnegative Matrices, M-Matrices

Theorem 1.29 *Let B be a nonnegative matrix. Then $\rho(B) < 1$ if and only if $I - B$ is nonsingular and $(I - B)^{-1}$ is nonnegative.*

Proof. Define $C = I - B$. If it is assumed that $\rho(B) < 1$, then by Theorem 1.11, $C = I - B$ is nonsingular and

$$C^{-1} = (I - B)^{-1} = \sum_{i=0}^{\infty} B^i. \tag{1.47}$$

In addition, since $B \geq 0$, all the powers of B as well as their sum in (1.47) are also nonnegative.

To prove the sufficient condition, assume that C is nonsingular and that its inverse is nonnegative. By the Perron-Frobenius theorem, there is a nonnegative eigenvector u associated with $\rho(B)$, which is an eigenvalue, i.e.,

$$Bu = \rho(B)u \qquad \text{or, equivalently,} \qquad C^{-1}u = \frac{1}{1 - \rho(B)}u.$$

Since u and C^{-1} are nonnegative, and $I - B$ is nonsingular, this shows that $1 - \rho(B) > 0$, which is the desired result. □

Nonnegative Matrices, M-Matrices *(M-Matrices)*

Definition 1.30 *A matrix is said to be an M-matrix if it satisfies the following four properties:*

1. $a_{i,i} > 0$ *for* $i = 1, \ldots, n$.
2. $a_{i,j} \leq 0$ *for* $i \neq j,\ i, j = 1, \ldots, n$.
3. A *is nonsingular.*
4. $A^{-1} \geq 0$.

In reality, the four conditions in the above definition are somewhat redundant and equivalent conditions that are more rigorous will be given later. Let A be any matrix which satisfies properties (1) and (2) in the above definition and let D be the diagonal of A. Since $D > 0$,

$$A = D - (D - A) = D\left(I - (I - D^{-1}A)\right).$$

Now define

$$B \equiv I - D^{-1}A.$$

Using the previous theorem, $I - B = D^{-1}A$ is nonsingular and $(I - B)^{-1} = A^{-1}D \geq 0$ if and only if $\rho(B) < 1$. It is now easy to see that conditions (3) and (4) of Definition 1.30 can be replaced by the condition $\rho(B) < 1$.

Theorem 1.31 *Let a matrix A be given such that*

1. $a_{i,i} > 0$ *for* $i = 1, \ldots, n$.
2. $a_{i,j} \leq 0$ *for* $i \neq j$, $i, j = 1, \ldots, n$.

Then A is an M-matrix if and only if

3. $\rho(B) < 1$, *where* $B = I - D^{-1}A$.

Proof. From the above argument, an immediate application of Theorem 1.29 shows that properties (3) and (4) of the above definition are equivalent to $\rho(B) < 1$, where $B = I - C$ and $C = D^{-1}A$. In addition, C is nonsingular iff A is and C^{-1} is nonnegative iff A is. □

The next theorem shows that the condition (1) in Definition 1.30 is implied by the other three.

Theorem 1.32 *Let a matrix A be given such that*

1. $a_{i,j} \leq 0$ *for* $i \neq j$, $i, j = 1, \ldots, n$.
2. A *is nonsingular.*
3. $A^{-1} \geq 0$.

Then

4. $a_{i,i} > 0$ *for* $i = 1, \ldots, n$, *i.e.,* A *is an* M*-matrix.*
5. $\rho(B) < 1$ *where* $B = I - D^{-1}A$.

5. $\rho(B) < 1$ *where* $B = I - D^{-1}A$.

Proof. Define $C \equiv A^{-1}$. Writing that $(AC)_{ii} = 1$ yields

$$\sum_{k=1}^{n} a_{ik}c_{ki} = 1 \qquad \text{which gives} \qquad a_{ii}c_{ii} = 1 - \sum_{\substack{k=1 \\ k \neq i}}^{n} a_{ik}c_{ki}.$$

Since $a_{ik}c_{ki} \leq 0$ for all k, the right-hand side is ≥ 1 and since $c_{ii} \geq 0$, then $a_{ii} > 0$. The second part of the result now follows immediately from an application of the previous theorem.

Finally, this useful result follows.

Theorem 1.33 *Let A, B be two matrices which satisfy*

1. $A \leq B$.
2. $b_{ij} \leq 0$ *for all* $i \neq j$.

Then if A is an M-matrix, so is the matrix B.

Proof. Assume that A is an M-matrix and let D_X denote the diagonal of a matrix X. The matrix D_B is positive because

$$D_B \geq D_A > 0.$$

Consider now the matrix $I - D_B^{-1}B$. Since $A \leq B$, then

$$D_A - A \geq D_B - B \geq O$$

which, upon multiplying through by D_A^{-1}, yields

$$I - D_A^{-1}A \geq D_A^{-1}(D_B - B) \geq D_B^{-1}(D_B - B) = I - D_B^{-1}B \geq O.$$

Since the matrices $I - D_B^{-1}B$ and $I - D_A^{-1}A$ are nonnegative, Theorems 1.28 and 1.31 imply that

$$\rho(I - D_B^{-1}B) \leq \rho(I - D_A^{-1}A) < 1.$$

This establishes the result by using Theorem 1.31 once again. □

Positive-Definite Matrices

A *real* matrix is said to be *positive definite* or *positive real* if

$$(Au, u) > 0, \quad \forall\, u \in \mathbb{R}^n,\ u \neq 0. \tag{1.48}$$

It must be emphasized that this definition is only useful when formulated entirely for real variables. Indeed, if u were not restricted to be real, then assuming that (Au, u) is real for all u complex would imply that A is Hermitian. If, in addition to the definition stated by 1.48, A is symmetric (real), then A is said to be *Symmetric Positive Definite* (SPD). Similarly, if A is Hermitian, then A is said to be *Hermitian Positive Definite* (HPD). Some properties of HPD matrices were seen in the above, in particular with regards to their eigenvalues. Now the more general case where A is non-Hermitian and positive definite is considered.

Positive-Definite Matrices

We begin with the observation that any square matrix (real or complex) can be decomposed as

in which

$$A = H + iS, \tag{1.49}$$

$$H = \frac{1}{2}(A + A^H) \tag{1.50}$$

$$S = \frac{1}{2i}(A - A^H). \tag{1.51}$$

Note that both H and S are Hermitian while the matrix iS in the decomposition (1.49) is skew-Hermitian. The matrix H in the decomposition is called the *Hermitian part* of A, while the matrix iS is the *skew-Hermitian part* of A. The above decomposition is the analogue of the decomposition of a complex number z into $z = x + iy$,

$$x = \Re e(z) = \frac{1}{2}(z + \bar{z}), \; y = \Im m(z) = \frac{1}{2i}(z - \bar{z}).$$

When A is real and u is a real vector then (Au, u) is real and, as a result, the decomposition (1.49) immediately gives the equality

$$(Au, u) = (Hu, u). \tag{1.52}$$

This results in the following theorem.

Positive-Definite Matrices

Theorem 1.34 *Let A be a real positive definite matrix. Then A is nonsingular. In addition, there exists a scalar $\alpha > 0$ such that*

$$(Au, u) \geq \alpha \|u\|_2^2, \tag{1.53}$$

for any real vector u.

Proof. The first statement is an immediate consequence of the definition of positive definiteness. Indeed, if A were singular, then there would be a nonzero vector such that $Au = 0$ and as a result $(Au, u) = 0$ for this vector, which would contradict (1.48). We now prove the second part of the theorem. From (1.52) and the fact that A is positive definite, we conclude that H is HPD. Hence, from (1.40) based on the min-max theorem, we get

$$\min_{u \neq 0} \frac{(Au, u)}{(u, u)} = \min_{u \neq 0} \frac{(Hu, u)}{(u, u)} \geq \lambda_{min}(H) > 0.$$

Taking $\alpha \equiv \lambda_{min}(H)$ yields the desired inequality (1.53). □

Positive-Definite Matrices

A simple yet important result which locates the eigenvalues of A in terms of the spectra of H and S can now be proved.

Theorem 1.35 *Let A be any square (possibly complex) matrix and let $H = \frac{1}{2}(A + A^H)$ and $S = \frac{1}{2i}(A - A^H)$. Then any eigenvalue λ_j of A is such that*

$$\lambda_{min}(H) \leq \Re e(\lambda_j) \leq \lambda_{max}(H) \tag{1.54}$$

$$\lambda_{min}(S) \leq \Im m(\lambda_j) \leq \lambda_{max}(S). \tag{1.55}$$

Proof. When the decomposition (1.49) is applied to the Rayleigh quotient of the eigenvector u_j associated with λ_j, we obtain

$$\lambda_j = (Au_j, u_j) = (Hu_j, u_j) + i(Su_j, u_j), \tag{1.56}$$

assuming that $\|u_j\|_2 = 1$. This leads to

$$\begin{aligned} \Re e(\lambda_j) &= (Hu_j, u_j) \\ \Im m(\lambda_j) &= (Su_j, u_j). \end{aligned}$$

The result follows using properties established in Section 1.9. □

Positive-Definite Matrices

Thus, the eigenvalues of a matrix are contained in a rectangle defined by the eigenvalues of its Hermitian part and its non-Hermitian part. In the particular case where A is real, then iS is skew-Hermitian and its eigenvalues form a set that is symmetric with respect to the real axis in the complex plane. Indeed, in this case, iS is real and its eigenvalues come in conjugate pairs.

Note that all the arguments herein are based on the field of values and, therefore, they provide ways to localize the eigenvalues of A from knowledge of the field of values. However, this approximation can be inaccurate in some cases.

Example 1.3. Consider the matrix

$$A = \begin{pmatrix} 1 & 1 \\ 10^4 & 1 \end{pmatrix}.$$

The eigenvalues of A are -99 and 101. Those of H are $1 \pm (10^4 + 1)/2$ and those of iS are $\pm i(10^4 - 1)/2$. □

Positive-Definite Matrices

When a matrix B is Symmetric Positive Definite, the mapping

$$x, y \quad \rightarrow \quad (x, y)_B \equiv (Bx, y) \tag{1.57}$$

from $\mathbb{C}^n \times \mathbb{C}^n$ to $\mathbb{C}$ is a proper inner product on $\mathbb{C}^n$, in the sense defined in Section 1.4. The associated norm is often referred to as the *energy norm* or *A-norm*. Sometimes, it is possible to find an appropriate HPD matrix B which makes a given matrix A Hermitian, i.e., such that

$$(Ax, y)_B = (x, Ay)_B, \quad \forall\, x, y$$

although A is a non-Hermitian matrix with respect to the Euclidean inner product. The simplest examples are $A = B^{-1}C$ and $A = CB$, where C is Hermitian and B is Hermitian Positive Definite.

Projection Operators

Projection operators or *projectors* play an important role in numerical linear algebra, particularly in iterative methods for solving various matrix problems. See the following the references (also used in this course) for more details:
James W. Demmel. Applied numerical linear algebra, Philadelphia, PA, SIAM (1997).
Lloyd N. Trefethen, David Bau III. Numerical linear algebra, Philadelphia, PA, SIAM (1997).
However, in what follows, we will introduce these operators from a purely algebraic point of view and gives a few of their important properties.

Range and Null Space of a Projector

A projector P is any linear mapping from $\mathbb{C}^n$ to itself which is idempotent, i.e., such that

$$P^2 = P.$$

A few simple properties follow from this definition. First, if P is a projector, then so is $(I - P)$, and the following relation holds,

$$\text{Null}(P) = \text{Ran}(I - P). \tag{1.58}$$

In addition, the two subspaces $\text{Null}(P)$ and $\text{Ran}(P)$ intersect only at the element zero. Indeed, if a vector x belongs to $\text{Ran}(P)$, then $Px = x$, by the idempotence property. If it is also in $\text{Null}(P)$, then $Px = 0$. Hence, $x = Px = 0$ which proves the result. Moreover, every element of $\mathbb{C}^n$ can be written as $x = Px + (I - P)x$. Therefore, the space $\mathbb{C}^n$ can be decomposed as the direct sum

$$\mathbb{C}^n = \text{Null}(P) \oplus \text{Ran}(P).$$

Range and Null Space of a Projector

Conversely, every pair of subspaces M and S which forms a direct sum of $\mathbb{C}^n$ defines a unique projector such that $\text{Ran}(P) = M$ and $\text{Null}(P) = S$. This associated projector P maps an element x of $\mathbb{C}^n$ into the component x_1, where x_1 is the M-component in the unique decomposition $x = x_1 + x_2$ associated with the direct sum.

In fact, this association is unique, that is, an arbitrary projector P can be entirely determined by two subspaces: (1) The range M of P, and (2) its null space S which is also the range of $I - P$. For any x, the vector Px satisfies the conditions,

$$\begin{aligned} Px &\in M \\ x - Px &\in S. \end{aligned}$$

The linear mapping P is said to project x *onto* M and *along* or *parallel to* the subspace S. If P is of rank m, then the range of $I - P$ is of dimension $n - m$. Therefore, it is natural to define S through its orthogonal complement $L = S^{\perp}$ which has dimension m. The above conditions that define $u = Px$ for any x become

$$u \in M \tag{1.59}$$

$$x - u \perp L. \tag{1.60}$$

Range and Null Space of a Projector

These equations define a projector P *onto* M and *orthogonal* to the subspace L. The first statement, (1.59), establishes the m degrees of freedom, while the second, (1.60), gives the m constraints that define Px from these degrees of freedom. The general definition of projectors is illustrated in Figure 1.1.

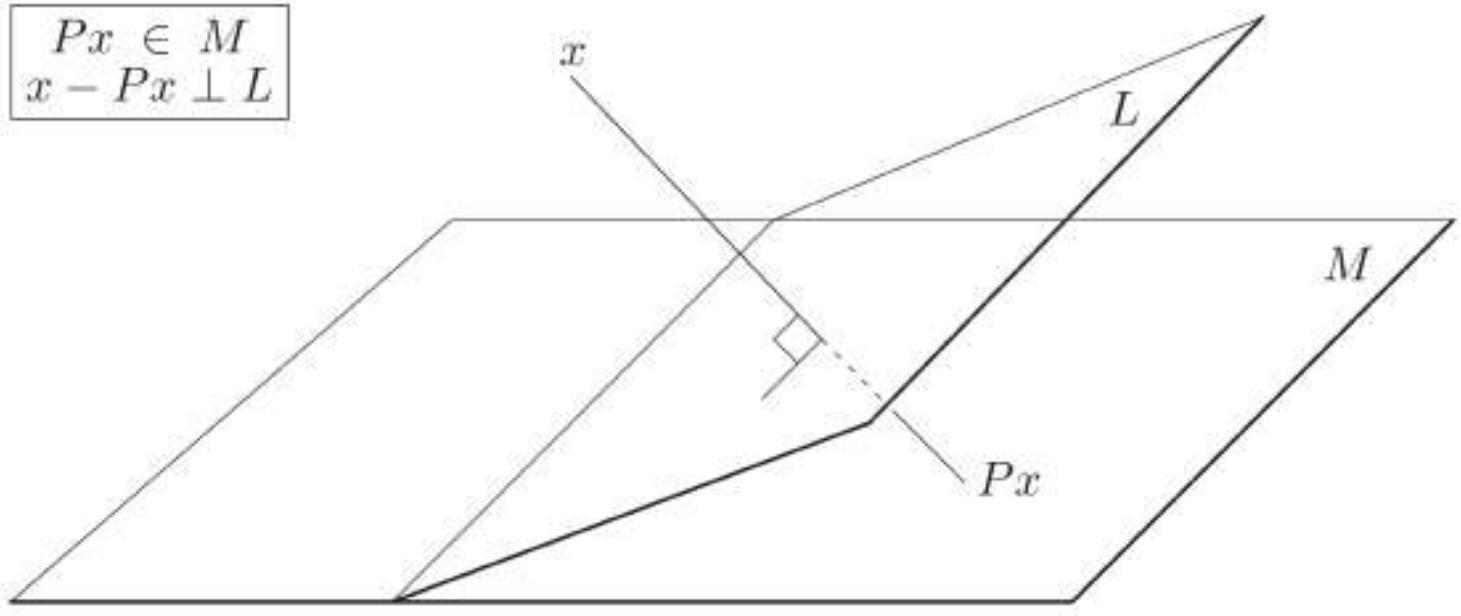


Figure 1.1 *Projection of x onto M and orthogonal to L.*

The question now is: Given two arbitrary subspaces, M and L both of dimension m, is it always possible to define a projector onto M orthogonal to L through the conditions (1.59) and (1.60)? The following lemma answers this question.

Range and Null Space of a Projector

Lemma 1.36 *Given two subspaces M and L of the same dimension m, the following two conditions are mathematically equivalent.*

i. No nonzero vector of M is orthogonal to L;

ii. For any x in $\mathbb{C}^n$ there is a unique vector u which satisfies the conditions

Proof. The first condition states that any vector which is in M and also orthogonal to L must be the zero vector. It is equivalent to the condition

$$M \cap L^{\perp} = \{0\}.$$

Since L is of dimension m, $L^{\perp}$ is of dimension $n - m$ and the above condition is equivalent to the condition that

$$\mathbb{C}^n = M \oplus L^{\perp}. \tag{1.61}$$

This in turn is equivalent to the statement that for any x, there exists a unique pair of vectors u, w such that

$$x = u + w,$$

where u belongs to M, and $w = x - u$ belongs to $L^{\perp}$, a statement which is identical with *ii*. □

Range and Null Space of a Projector

In summary, given two subspaces M and L, satisfying the condition $M \cap L^{\perp} = \{0\}$, there is a projector P onto M orthogonal to L, which defines the projected vector u of any vector x from equations (1.59) and (1.60). This projector is such that

$$\text{Ran}(P) = M, \qquad \text{Null}(P) = L^{\perp}.$$

In particular, the condition $Px = 0$ translates into $x \in \text{Null}(P)$ which means that $x \in L^{\perp}$. The converse is also true. Hence, the following useful property,

$$Px = 0 \qquad \text{iff} \qquad x \perp L. \tag{1.62}$$

Matrix Representations

Two bases are required to obtain a matrix representation of a general projector: a basis $V = [v_1, \ldots, v_m]$ for the subspace $M = \mathrm{Ran}(P)$ and a second one $W = [w_1, \ldots, w_m]$ for the subspace L. These two bases are *biorthogonal* when

$$(v_i, w_j) = \delta_{ij}. \tag{1.63}$$

In matrix form this means $W^H V = I$. Since Px belongs to M, let Vy be its representation in the V basis. The constraint $x - Px \perp L$ is equivalent to the condition,

$$((x - Vy), w_j) = 0 \quad \text{for } j = 1, \ldots, m.$$

In matrix form, this can be rewritten as

$$W^H(x - Vy) = 0. \tag{1.64}$$

If the two bases are biorthogonal, then it follows that $y = W^H x$. Therefore, in this case, $Px = VW^H x$, which yields the matrix representation of P,

If the two bases are biorthogonal, then it follows that $y = W^H x$. Therefore, in this case, $Px = VW^H x$, which yields the matrix representation of P,

$$P = VW^H. \tag{1.65}$$

In case the bases V and W are not biorthogonal, then it is easily seen from the condition (1.64) that

$$P = V(W^H V)^{-1} W^H. \tag{1.66}$$

If we assume that no vector of M is orthogonal to L, then it can be shown that the $m \times m$ matrix $W^H V$ is nonsingular.

Orthogonal and Oblique Projectors

An important class of projectors is obtained in the case when the subspace L is equal to M, i.e., when

$$\text{Null}(P) = \text{Ran}(P)^{\perp}.$$

Then, the projector P is said to be the *orthogonal projector* onto M. A projector that is not orthogonal is *oblique*. Thus, an orthogonal projector is defined through the following requirements satisfied for any vector x,

$$Px \in M \quad \text{and} \quad (I - P)\, x \perp M \tag{1.67}$$

or equivalently,

Orthogonal and Oblique Projectors

$$Px \in M \quad \text{and} \quad ((I-P)x, y) = 0 \quad \forall\, y \in M.$$

$$\boxed{\begin{array}{c} Px \in M \\ x - Px \perp M \end{array}}$$

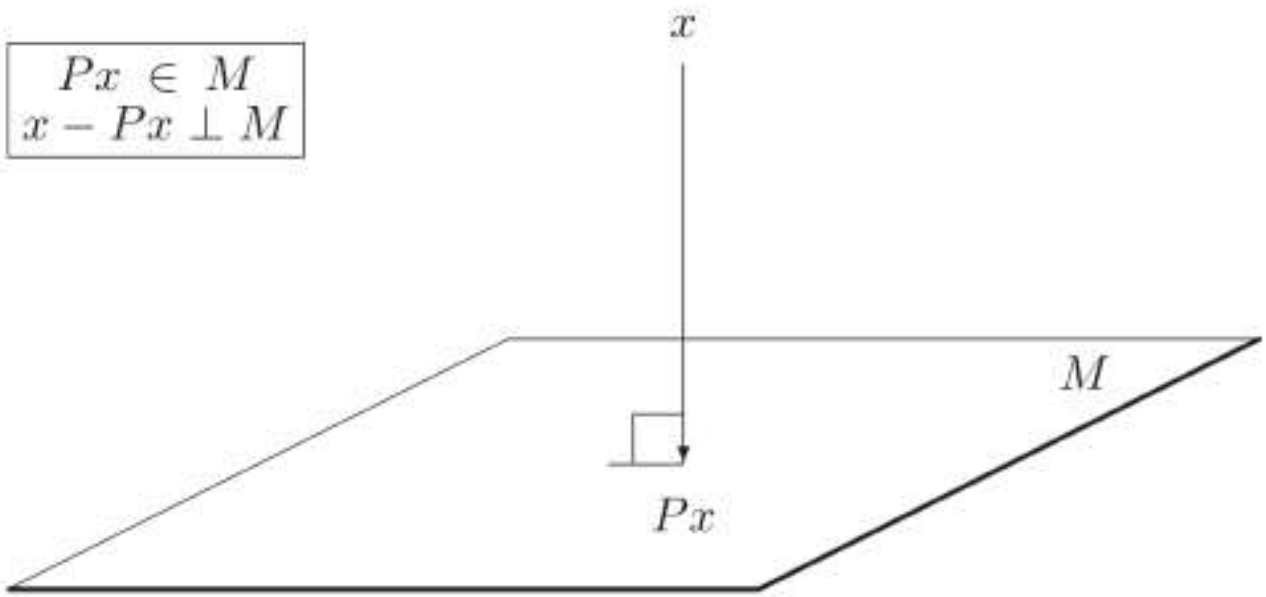


Figure 1.2 *Orthogonal projection of x onto a subspace M.*

Orthogonal and Oblique Projectors

It is interesting to consider the mapping P^H defined as the adjoint of P

$$(P^H x, y) = (x, Py), \quad \forall\, x, \forall\, y. \tag{1.68}$$

First note that P^H is also a projector because for all x and y,

$$((P^H)^2 x, y) = (P^H x, Py) = (x, P^2 y) = (x, Py) = (P^H x, y).$$

A consequence of the relation (1.68) is

$$\text{Null}(P^H) = \text{Ran}(P)^\perp \tag{1.69}$$

$$\text{Null}(P) = \text{Ran}(P^H)^\perp. \tag{1.70}$$

The above relations lead to the following proposition.

Orthogonal and Oblique Projectors

Proposition 1.37 *A projector is orthogonal if and only if it is Hermitian.*

Proof. By definition, an orthogonal projector is one for which $\mathrm{Null}(P) = \mathrm{Ran}(P)^{\perp}$. Therefore, by (1.69), if P is Hermitian, then it is orthogonal. Conversely, if P is orthogonal, then (1.69) implies $\mathrm{Null}(P) = \mathrm{Null}(P^H)$ while (1.70) implies $\mathrm{Ran}(P) = \mathrm{Ran}(P^H)$. Since P^H is a projector and since projectors are uniquely determined by their range and null spaces, this implies that $P = P^H$. □

Given *any* unitary $n \times m$ matrix V whose columns form an orthonormal basis of $M = \mathrm{Ran}(P)$, we can represent P by the matrix $P = VV^H$. This is a particular case of the matrix representation of projectors (1.65). In addition to being idempotent, the linear mapping associated with this matrix satisfies the characterization given above, i.e.,

$$VV^H x \in M \quad \text{and} \quad (I - VV^H)x \in M^{\perp}.$$

It is important to note that this representation of the orthogonal projector P is not unique. In fact, any orthonormal basis V will give a different representation of P in the above form. As a consequence for any two orthogonal bases V_1, V_2 of M, we must have $V_1 V_1^H = V_2 V_2^H$, an equality which can also be verified independently.

Properties of Orthogonal Projectors

When P is an orthogonal projector, then the two vectors Px and $(I - P)x$ in the decomposition $x = Px + (I - P)x$ are orthogonal. The following relation results:

$$\|x\|_2^2 = \|Px\|_2^2 + \|(I - P)x\|_2^2.$$

A consequence of this is that for any x,

$$\|Px\|_2 \leq \|x\|_2.$$

Thus, the maximum of $\|Px\|_2/\|x\|_2$, for all x in $\mathbb{C}^n$ does not exceed one. In addition the value one is reached for any element in $\mathrm{Ran}(P)$. Therefore,

$$\|P\|_2 = 1$$

for any orthogonal projector P.

An orthogonal projector has only two eigenvalues: zero or one. Any vector of the range of P is an eigenvector associated with the eigenvalue one. Any vector of the null-space is obviously an eigenvector associated with the eigenvalue zero.

Next, an important optimality property of orthogonal projectors is established.

Properties of Orthogonal Projectors

Theorem 1.38 *Let P be the orthogonal projector onto a subspace M. Then for any given vector x in $\mathbb{C}^n$, the following is true:*

$$\min_{y \in M} \|x - y\|_2 = \|x - Px\|_2. \tag{1.71}$$

Proof. Let y be any vector of M and consider the square of its distance from x. Since $x - Px$ is orthogonal to M to which $Px - y$ belongs, then

$$\|x - y\|_2^2 = \|x - Px + (Px - y)\|_2^2 = \|x - Px\|_2^2 + \|(Px - y)\|_2^2.$$

Therefore, $\|x - y\|_2 \geq \|x - Px\|_2$ for all y in M. This establishes the result by noticing that the minimum is reached for $y = Px$. □

By expressing the conditions that define $y^* \equiv Px$ for an orthogonal projector P onto a subspace M, it is possible to reformulate the above result in the form of necessary and sufficient conditions which enable us to determine the best approximation to a given vector x in the least-squares sense.

Corollary 1.39 *Let a subspace M, and a vector x in $\mathbb{C}^n$ be given. Then*

$$\min_{y \in M} \|x - y\|_2 = \|x - y^*\|_2, \tag{1.72}$$

if and only if the following two conditions are satisfied,

$$\begin{cases} y^* & \in & M \\ x - y^* & \perp & M. \end{cases}$$

Existence (and uniqueness) of a solution

From the numerical viewpoint is far more complex!

Linear systems are among the most important and common problems encountered in scientific computing. From the theoretical point of view, it is well understood when a solution exists, when it does not, and when there are infinitely many solutions. In addition, explicit expressions of the solution using determinants exist. However, the numerical viewpoint is far more complex. Approximations may be available but it may be difficult to estimate how accurate they are. This clearly will depend on the data at hand, i.e., primarily on the coefficient matrix. This section gives a very brief overview of the existence theory as well as the sensitivity of the solutions.

Existence (and uniqueness) of a solution

From the numerical viewpoint is far more complex!

Consider the *linear system*

$$Ax = b. \tag{1.73}$$

Here, x is termed the *unknown* and b the *right-hand side.* When solving the linear system (1.73), we distinguish three situations.

Case 1 The matrix A is nonsingular. There is a unique solution given by $x = A^{-1}b$.

Case 2 The matrix A is singular and $b \in \mathrm{Ran}(A)$. Since $b \in \mathrm{Ran}(A)$, there is an x_0 such that $Ax_0 = b$. Then $x_0 + v$ is also a solution for any v in $\mathrm{Null}(A)$. Since $\mathrm{Null}(A)$ is at least one-dimensional, there are infinitely many solutions.

Case 3 The matrix A is singular and $b \notin \mathrm{Ran}(A)$. There are no solutions.

Gilbert Strang. Linear algebra and its applications, 3rd ed., Brooks/Cole, Thomson Learning,(1988). Kenneth Hoffman and Ray Kunze. Linear algebra, 2nd ed, Englewood Cliffs, NJ, Prentice-Hall (1971).