Technische Universität Braunschweig

Institute for Numerical Analysis

Ramp Up Mathematics — Numerical Analysis Ramp Up for Data Science

Matthias Bollhöfer, SS 2024

Contents I

- **Basics**
  - Finite Precision Representation
  - Round off Errors
  - Error Propagation
- **Error Analysis**
  - Introduction
  - Conditioning
  - Forward and Backward Analysis
- **Linear Systems**
  - Matrix Norms
  - Condition when Solving Linear Systems
  - The *LU* Decomposition
  - The Cholesky Decomposition
  - The CG Method
- **Nonlinear Systems**
  - Newton's Method

Contents II

- **Linear Least Squares**
  - Linear Least Squares and Normal Equations
  - The QR Decomposition
  - Stable Computation of the QR Decomposition
- **The Singular Value Decomposition**
  - Basic Properties
  - The Key Property
- **Fourier Transformation**
  - Fourier Series and Discrete Fourier Transformation
  - Fast Fourier Transformation

Short Refresh Matrix Norms

Recall:

- given $A \in \mathbb{R}^{m,n}$ and vector norm $\|\bullet\|_p$ we define the induced matrix norm via $\|A\|_p := \max\limits_{\|\vec{x}\|_p=1} \|A\vec{x}\|_p$
- most prominent examples
  - max norm $p = \infty$, $\|A\|_\infty = \max\limits_{i=1,\dots,m} \sum\limits_{j=1,\dots,n} |a_{ij}|$, "maximum row sum"
  - 1–norm $p = 1$, $\|A\|_1 = \max\limits_{j=1,\dots,n} \sum\limits_{i=1,\dots,m} |a_{ij}|$, "maximum column sum"
  - for matrices A, B and a vector $\vec{x}$ this implies that $\|A\vec{x}\| \leqslant \|A\| \, \|\vec{x}\|$, $\|AB\| \leqslant \|A\| \, \|B\|$

Example 2.1

Let $M = \begin{bmatrix} 1 & 5 \\ -3 & 0 \end{bmatrix}$*, then we obtain*

1. $\|A\|_\infty =$,
2. $\|A\|_1 =$

Condition Number

Consider solving a linear system

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

with a given nonsingular matrix $A \in \mathbb{R}^{n,n}$ and right hand side $\vec{b} \in \mathbb{R}^n$. We are seeking for the solution $\vec{x} \in \mathbb{R}^n$.

Definition 2.1 (Condition number)

Let $A \in \mathbb{R}^{n,n}$ be invertible. Then we call $\kappa_p(A) = \|A^{-1}\|_p \, \|A\|_p$ condition number of A

Example 2.2 (Condition numbers)

$$M = \begin{bmatrix} 1 & 5 \\ -3 & 0 \end{bmatrix} \Rightarrow \det M = \qquad \Rightarrow M^{-1} =$$

norm-wise condition w.r.t. $\| \bullet \|_\infty$

$$\kappa_\infty(M) = \|M^{-1}\|_\infty \cdot \|M\|_\infty =$$

Condition Number

Now consider for some small $\varepsilon > 0$ the perturbed linear system

$$(A + \varepsilon F)\vec{x}(\varepsilon) = \vec{b} + \varepsilon \vec{f}$$

with some suitable perturbation matrix $F \in \mathbb{R}^{n,n}$ and some perturbation vector $\vec{f} \in \mathbb{R}^n$, rescaled such that $\|F\| = \|A\|$, $\|\vec{f}\| = \|\vec{b}\|$.
Relative input errors:

$$\frac{\|(A + \varepsilon F) - A\|}{\|A\|} \leqslant \varepsilon, \quad \frac{\|(\vec{b} + \varepsilon \vec{f}) - \vec{b}\|}{\|\vec{b}\|} \leqslant \varepsilon.$$

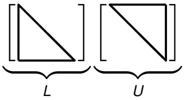
Then one can show that

$$\frac{\|\vec{x}(\varepsilon) - \vec{x}\|}{\|\vec{x}\|} \leqslant 2|\varepsilon| \cdot \kappa(A) + \mathcal{O}(\varepsilon^2)$$

The condition number $\kappa(A)$ measures how errors in the input data A and b amplify the output result $\vec{x}$.

The LU Decomposition

- we now briefly recall Gaussian elimination, the most common method to solve linear systems
- Without pivoting(row interchanges), Gaussian elimination is not backward stable!
- Gaussian elimination is also referred to as LU decomposition, since transforming a matrix A to upper triangular form U also yields a lower triangular matrix L with unit diagonal.
  L consists of the elimination parameters and we obtain $PA = LU$, where P refers to interchanges by pivoting.
- decomposition: L lower, U upper triangular

$PA =$


- once the factorization is computed solve the linear system $A\vec{x} = \vec{b}$ as follows:
  1. $\vec{b} \to \vec{c} = P\vec{b}$
  2. solve $L\vec{y} = \vec{c}$ by forward substitution,
  3. after that , solve $U\vec{x} = \vec{y}$ by back(ward) substitution

  $\Rightarrow$ $P\vec{b} = \vec{c} = L\vec{y} = L(U\vec{x}) = PA\vec{x}$, we have computed the solution $\vec{x}$

Stability of the LU Decomposition — Partial Pivoting

- Without interchanges, the diagonal entries a_{kk} can become zero or small in magnitude (which is numerically the almost the same as if they were zero)

- we will introduce *partial pivoting* to stabilize the algorithm, before eliminating entries in column k:
  1. find $r = \mathrm{argmax}_{s \geqslant k} |a_{sk}|$
  2. interchange rows r and k
  3. eliminate sub-diagonal entries in column k

Partial Pivoting

Example 2.3 (*LU* decomposition with partial pivoting)

$$A = \begin{bmatrix} 3 & 17 & 10 \\ 2 & 4 & -2 \\ 6 & 18 & -12 \end{bmatrix} \qquad \boxed{\begin{array}{ccc} 3 & 17 & 10 \\ 2 & 4 & -2 \\ 6 & 18 & -12 \end{array}} \quad \boxed{\begin{array}{c} 1. \\ 2. \\ 3. \end{array}} \qquad \begin{array}{c} R1 \\ \updownarrow \\ R3 \end{array}$$

$$\rightarrow \qquad \boxed{\begin{array}{c} 3. \\ 2. \\ 1. \end{array}} \qquad \begin{array}{c} \\ -\frac{1}{3} \cdot R1 \\ -\frac{1}{2} \cdot R1 \end{array}$$

$$\rightarrow \qquad \boxed{\begin{array}{c} 3. \\ 2. \\ 1. \end{array}} \qquad \begin{array}{c} R2 \\ \updownarrow \\ R3 \end{array}$$

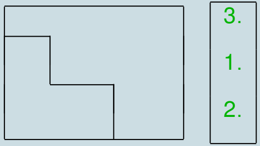
Partial Pivoting

Example 2.4 (*LU* decomposition with partial pivoting (continued))

$\rightarrow$

$-\frac{1}{4} \cdot R2$

$\rightarrow$



Note that we have interchanged *complete rows* of L and U!

Partial Pivoting

Example 2.5 (*LU* decomposition with partial pivoting (continued))

Because of the interchanges we have to reorder the rows of A accordingly

$$\underbrace{\begin{bmatrix} 3 & 17 & 10 \\ 2 & 4 & -2 \\ 6 & 18 & -12 \end{bmatrix}}_{A} \to \underbrace{\begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}}_{PA} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ \frac{1}{3} & -\frac{1}{4} & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} 6 & 18 & -12 \\ 0 & 8 & 16 \\ 0 & 0 & 6 \end{bmatrix}}_{U}$$

using the permutation matrix

$$P = \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}$$

Example 2.6 (MATLAB-Demo `lugui`)

$\gg$ lugui *% cf. MathWorks web site*

Partial Pivoting

Example 2.7 (Forward / Back(ward) substitution)

$$\begin{bmatrix} 3 & 17 & 10 \\ 2 & 4 & -2 \\ 6 & 18 & -12 \end{bmatrix} x = \begin{bmatrix} 30 \\ 4 \\ 12 \end{bmatrix}$$

solve linear system using LU decomposition and partial pivoting $PA = LU$

2.1 *Interchange components of* $\begin{bmatrix} 30 \\ 4 \\ 12 \end{bmatrix} \to \begin{bmatrix} \\ \\ \end{bmatrix}$ *w.r.t.* p

2.2 *denote by* $\vec{y}$ *the relation* $\vec{y} = U\vec{x}$

2.3 *solve* $\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ \frac{1}{3} & -\frac{1}{4} & 1 \end{bmatrix}}_{L} \vec{y} = \begin{bmatrix} \\ \\ \end{bmatrix}$, *we obtain* $\vec{y} = \begin{bmatrix} \\ \\ \end{bmatrix}$

Partial Pivoting

Example 2.8 (Forward / Back(ward) substitution (continued))

2.4 *solve* $\underbrace{\begin{bmatrix} 6 & 18 & -12 \\ 0 & 8 & 16 \\ 0 & 0 & 6 \end{bmatrix}}_{U} \vec{x} = \begin{bmatrix} \\ \\ \\ \end{bmatrix}$, *we obtain finally* $\vec{x} = \begin{bmatrix} \\ \\ \\ \end{bmatrix}$.

We have seen that *complete rows* of L and U have to be interchanged in order to correctly handle the permutations.

Theorem 2.1 (LU decomposition with partial pivoting)

Let $A \in \mathbb{R}^{n,n}$ be nonsingular. There exist a permutation matrix P, a lower triangular matrix L with unit diagonal where $|l_{ij}| \leqslant 1$ and an upper triangular matrix U such that

$$PA = LU,$$

Costs: LU decomposition $\mathcal{O}(n^3)$, interchanges $\mathcal{O}(n^2)$.

Cholesky Decomposition

- Let $A = A^T \in \mathbb{R}^{n,n}$ be a symmetric matrix, i.e., $a_{ij} = a_{ji}$, for all $i, j = 1, \ldots, n$. Then we know that
  1. *all eigenvalues* $\lambda_1, \ldots, \lambda_n$ are *real*
  2. there exists a *complete set of eigenvectors* $q_1, \ldots, q_n \in \mathbb{R}^n$ which can be chosen *orthonormal*, i.e., we have

$$A = Q \Lambda Q^{-1} = Q \Lambda Q^T, \text{ where } Q = [q_1, \ldots, q_n], \ \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n).$$

- Let in addition $A = A^T$ be positive definite (SPD), i.e., all *eigenvalues* are *positive* or, equivalently, $\vec{x}^\top A \vec{x} > 0$, for all $\vec{x} \in \mathbb{R}^n \setminus \{0\}$.
- now consider solving a linear system

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

  with an SPD matrix.
- In this case one can show that
  1. pivoting is not needed
  2. the *LU* decomposition is symmetric as well, i.e., $A = GG^T$ for a lower triangular matrix G
  3. the diagonal entries g_{kk} of G can be shown to be positive as well

Cholesky Decomposition

Let A be SPD and suppose that $A = LU$. Denote by D the diagonal matrix which has the same diagonal entries as U.

- It follows that $A = LU = LD(D^{-1}U)$. Because of symmetry we must already have $L^\top = D^{-1}U$

$$\Rightarrow A = LDL^\top = \underbrace{(LD^{1/2})}_{G}\ \underbrace{(D^{1/2}L^\top)}_{G^\top},\ \text{where } D^{1/2} = \mathrm{dgl}(\sqrt{u_{1,1}}, \dots, \sqrt{u_{n,n}}).$$

- this variant of the LU decomposition is referred to as *Cholesky decomposition*

Theorem 2.2 (Cholesky Decomposition)

Let $A \in \mathbb{R}^{n,n}$ be an SPD matrix. Then there exists a unique lower triangular matrix $G \in \mathbb{R}^{n,n}$ with positive diagonal entries such that

$$A = GG^\top.$$

Cholesky Decomposition

Example 2.9 (Cholesky Decomposition)

$$A = \begin{bmatrix} 4 & -2 & -4 \\ -2 & 5 & 4 \\ -4 & 4 & 9 \end{bmatrix} \Rightarrow A = \underbrace{\begin{bmatrix} & 0 & 0 \\ & & 0 \\ & & \end{bmatrix}}_{G} \underbrace{\begin{bmatrix} & & \\ 0 & & \\ 0 & 0 & \end{bmatrix}}_{G^\top}$$

Example 2.10 (MATLAB-Demo `chol`)

$\gg$ help chol

- Costs $\mathcal{O}(n^3)$, because of symmetry roughly half as expensive as LU decomposition
- Cholesky algorithm is numerically *stable*
- Solving$A\vec{x} = \vec{b}$ using forward/back(ward) substitution with G and G^T
  1. Compute Cholesky decomposition $A = GG^T$
  2. solve $G\vec{y} = \vec{b}$
  3. solve $G^T\vec{x} = \vec{y}$

The Conjugate Gradient Method

We still assume that $A \in \mathbb{R}^{n,n}$ is SPD.
The **conjugate gradient** (CG) method solves the unconstrained minimization problem

$$\vec{x}^* = \underset{\vec{x}}{\operatorname{argmin}}\, g(\vec{x}), \text{ where } g(\vec{x}) = \frac{1}{2}\vec{x}^T A\vec{x} - \vec{x}^T\vec{b}$$

iteratively using skillfully chosen descent directions.

Given an approximate solution $\vec{x}_{k-1}$, CG defines $\vec{x}_k := \vec{x}_{k-1} + \alpha_k \vec{p}_k$, where $\vec{p}_k$ is a given search direction and α_k is chosen to minimize

$$\alpha_k = \underset{\alpha}{\operatorname{argmin}}\, g(\vec{x}_{k-1} + \alpha\vec{p}_k)$$

This way $\vec{x}_k$ is obtained.
Minimization yields

$$0 = \frac{d}{d\alpha} g(\vec{x}_{k-1} + \alpha\vec{p}_k) = \nabla g(\vec{x}_{k-1} + \alpha\vec{p}_k) \cdot \vec{p}_k = (A(\vec{x}_{k-1} + \alpha\vec{p}_k) - \vec{b})^T p_k$$

$$\Rightarrow \alpha_k \equiv \alpha = \frac{(\vec{b} - A\vec{x}_{k-1})^T\vec{p}_k}{\vec{p}_k^T A\vec{p}_k} = \frac{\vec{r}_{k-1}^T\vec{p}_k}{\vec{p}_k^T A\vec{p}_k}, \text{ where } \vec{r}_{k-1} = \vec{b} - A\vec{x}_{k-1}$$

Computations with Large Matrices — CG method

One can show that from a global perspective, the optimal choices of $\vec{p}_1, \vec{p}_2, \vec{p}_3, \ldots$ have to satisfy

$$\vec{p}_i^T A \vec{p}_j = 0, \text{ for all } i \neq j$$

These are so-called *conjugate directions*.
Given an initial guess $\vec{x}_0 \in \mathbb{R}^p$, introduce residual vectors $\vec{r}_k = \vec{b} - A\vec{x}_k$, $k = 0, 1, 2, \ldots$
One can show that $\vec{p}_{k+1}$ can be easily computed from $\vec{p}_k$, $\vec{r}_k$:

- $\vec{x}_k = \vec{x}_{k-1} + \alpha_k \vec{p}_k$,
- $\vec{r}_k = \vec{b} - A(\vec{x}_{k-1} + \alpha_k \vec{p}_k) = \vec{r}_{k-1} - \alpha_k A \vec{p}_k$,
- $\vec{p}_{k+1} = \vec{r}_k + \beta_k \vec{p}_k$,

where $\vec{p}_1 = \vec{r}_0$, $\alpha_k = \frac{\vec{r}_{k-1}^T \vec{p}_k}{\vec{p}_k^T A \vec{p}_k} = \frac{\rho_{k-1}}{\vec{p}_k^T A \vec{p}_k}$, $\beta_k = \frac{\rho_k}{\rho_{k-1}}$, $\rho_k = \vec{r}_k^T \vec{r}_k$.

CG method

Theorem 2.3

Let $A \in \mathbb{R}^{p,p}$ be SPD., $\vec{x}_0 \in \mathbb{R}^p$ initial guess. Define the energy norm induced by A via $\|\vec{x}\|_A = \sqrt{\vec{x}^T A \vec{x}}$. Then after k steps of the CG method we have

$$\|\vec{x} - \vec{x}_k\|_A \leqslant 2 \left(\frac{\sqrt{\kappa_2(A)} - 1}{\sqrt{\kappa_2(A)} + 1} \right)^k \|\vec{x} - \vec{x}_0\|_A$$

Example 2.11 (MATLAB-Demo pcg)

$\gg$ help pcg

- From a practical point of view it is advantageous to find a cheap SPD matrix M, $M = LL^T$ such that $M \approx A$ and $\kappa_2(L^{-1}AL^{-T}) \ll \kappa_2(A)$ to accelerate convergence.
- This process is called *preconditioning* and in principle we solve $L^{-1}AL^{-T}\vec{y} = L^{-1}\vec{b}$, where $L^{-T}\vec{y} = \vec{x}$, instead.
- In practice the CG method needs to be changed only slightly with a step of type $M\vec{y} = \vec{c}$ at every step $\rightarrow$ PCG.

Preconditioned Conjugate Gradient (PCG) Method

Example 2.12 (MATLAB)

$$A = \begin{pmatrix} 1 & 1 & & \\ 1 & 2 & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & n \end{pmatrix}, \quad b = A \cdot \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix},$$

where we set $n = 10^5$.

- *run* `pcg` *with a tolerance of* 10^{-6}
- *next run* `pcg` *with a tolerance of* 10^{-6} *but use* $M = \mathrm{dgl}(1, \ldots, n)$ *for preconditioning.*