Linear Equations

- Problem Formulation
- Conditioning of Linear Equation Systems
- Gauss Elimination
- Linear Equations with Band-Structured Matrix
- Linear Equations with Positive Definite Matrices

Contents

Problem Formulation

Given a matrix $A \in \mathbb{R}^{m \times n}$ and a vector $b \in \mathbb{R}^m$. We are searching for solutions of the linear equation system

$$Ax = b \, .$$

Names:

- For $m < n$: "under-determined linear equation system"
- For $m = n$: "quadratic linear equation system"
- For $m > n$: "over-determined linear equation system"

Known Results from Linear Algebra

The equation system $Ax = b$ has a solution if and only if

$\text{rank}(A) = \text{rank}(A, b)$.

This condition can, e.g., by checked with Gram-Schmidt algorithms.

For the quadratic case $n = m$, the following statements are equivalent:

- $Ax = b$ has the unique solution x.
- $\text{rank}(A) = n$.
- $\det(A) \neq 0$.
- All eigenvalues of A are different from 0.

Contents

Overview: Perturbation Theory

Whenever we solve equations of the form $Ax = b$ numerically, we have to take into account two possible sources of errors:

- Errors that are due to working with inexact numerical values for A and b.
- Errors that are due to numerical rounding problems when implementing the algorithm based on finite precision arithmetics.

Matrix Norms

Recall that any vector norm $\|\cdot\| : \mathbb{R}^n \to \mathbb{R}$ can be generalized for matrices by defining

$$\|A\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|} \ ,$$

for any $A \in \mathbb{R}^{m \times n}$.

Simple examples:

- $\|A\|_1 = \max_j \ \sum_{i=0}^{m} \ |A_{i,j}|$.
- $\|A\|_\infty = \max_i \ \sum_{j=0}^{n} \ |A_{i,j}|$.

The Spectral Norm

The eigenvalues $\lambda_1, \dots, \lambda_n \in \mathbb{C}$ of a square matrix $A \in \mathbb{R}^{n \times n}$ are the roots of the characteristic polynomial $p(\lambda) = \det(A - \lambda I)$.

The matrix 2-norm is given by

$$\|A\|_2 = \max_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2} = \max \left\{ \sqrt{\lambda} \mid \det(A^T A - \lambda I) = 0 \right\} \ ,$$

If A is symmetric this expression for the 2-norm can be simplified

$$\|A\|_2 = \max \left\{ |\lambda| \mid \det(A - \lambda I) = 0 \right\} \ .$$

A Useful Matrix-Norm Inequality

If $\|\cdot\|$ is a matrix norm (assume $\|I\| = 1$) and $A \in \mathbb{R}^{n\times n}$ a given matrix whose norm satisfies $\|A\| < 1$, then the matrix $I + A$ is invertible and we have

$$\|(I+A)^{-1}\| \leq \frac{1}{1-\|A\|}$$

Proof: From the inequality

$$\|(I+A)x\| \geq \underbrace{(I - \|A\|)}_{>0} \|x\|$$

we conclude that we have $(I + A)x \neq 0$ for all vectors $x \neq 0$; that is, $I + A$ is invertible. Moreover, we have

$$1 = \|(I+A)(I+A)^{-1}\| \geq \|(I+A)^{-1}\|(1 - \|A\|) \ ,$$

which implies the statement.

Condition Numbers

Theorem:

Let x denote the solution of the invertible and quadratic linear equation system $Ax = b$ and let δx be such that

$$(A + \delta A)(x + \delta x) = b + \delta b \; .$$

assuming $\|\delta A\| < \frac{1}{\|A^{-1}\|}$ and $\|b\| \neq 0$ as well as $\|A\| \neq 0$. Then we have

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{\mathrm{cond}(A)}{1 - \mathrm{cond}(A)\frac{\|\delta A\|}{\|A\|}} \left(\frac{\|\delta b\|}{\|b\|} + \frac{\|\delta A\|}{\|A\|} \right) \quad \text{with} \quad \mathrm{cond}(A) = \|A\| \cdot \|A^{-1}\| \; .$$

Interpretation:

The condition number $\mathrm{cond}(A)$ can be interpreted as an error amplification factor,

$$\frac{\|\delta x\|}{\|x\|} \approx \mathrm{cond}(A) \left(\frac{\|\delta b\|}{\|b\|} + \frac{\|\delta A\|}{\|A\|} \right) \; .$$

Proof

From $Ax = b$ and the equation

$$(A + \delta A)(x + \delta x) = b + \delta b \; .$$

we find $\delta x = (A + \delta A)^{-1}(\delta b - \delta A x)$ (recall that $\|\delta A\| \leq \frac{1}{\|A^{-1}\|}$ implies that $A + \delta A$ is invertible). This yields the estimate

$$\begin{aligned}
\|\delta x\| &\leq \|(A + \delta A)^{-1}\| \left(\|\delta b\| + \|\delta A\|\|x\|\right) \\
&\leq \|(I + A^{-1}\delta A)^{-1}\|\|A^{-1}\| \left(\|\delta b\| + \|\delta A\|\|x\|\right) \\
&\leq \frac{\|A^{-1}\|}{1 - \|A^{-1}\|\|\delta A\|} \left(\|\delta b\| + \|\delta A\|\|x\|\right)
\end{aligned}$$

(Last step has used the matrix-norm ineq. from previous slides)

Proof (continued)

Because we have $Ax = b$, it follows that $\|A\|\|x\| \geq \|b\|$ and

$$\begin{aligned}
\|\delta x\| &\leq \frac{\|A^{-1}\|}{1 - \|A^{-1}\|\|\delta A\|} \left(\|\delta b\| + \|\delta A\|\|x\|\right) \\
&\leq \frac{\|A^{-1}\|\|A\|\|x\|}{1 - \|A^{-1}\|\|\delta A\|} \left(\frac{\|\delta b\|}{\|b\|} + \frac{\|\delta A\|}{\|A\|}\right) .
\end{aligned}$$

In the last step we substitute $\mathsf{cond}(A) = \|A\| \cdot \|A^{-1}\|$:

$$\|\delta x\| \leq \frac{\mathsf{cond}(A)\|x\|}{1 - \mathsf{cond}(A)\frac{\|\delta A\|}{\|A\|}} \left(\frac{\|\delta b\|}{\|b\|} + \frac{\|\delta A\|}{\|A\|}\right) .$$

The statement of the theorem follows after dividing by $\|x\|$.

Contents

Triangular Matrices

If the matrix A is triangular, i.e.,

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ 0 & a_{2,2} & \dots & a_{2,n} \\ \vdots & \ddots & \ddots & \\ 0 & \dots & 0 & a_{n,n} \end{pmatrix},$$

the system Ax can be solved by a "backwards substitution":

$$x_n = \frac{b_n}{a_{n,n}} \qquad \text{and} \qquad x_j = \frac{1}{a_{j,j}} \left(b_j - \sum_{k=j+1}^{n} a_{j,k} x_k \right)$$

for $j = n-1, \dots, 1$. (Complexity: $\mathbf{O}(n^2)$.)

Gauss Elimination

For general matrices A the main idea is to transform the matrix step by step into an upper triangular matrix. For this aim, three types of operations can be applied:

- We may swap (permute) rows of A if necessary.
- We may swap (permute) columns as long as we re-enumerate the unknowns x_j, too.
- We may multpily one row by a scalar factor and add it to another row.

Gauss Elimination

In the first step of the Gauss elimination procedure is as follows:

- Permute the rows of A such that $a_{1,1} \neq 0$.
- For all row indices $j = 2, \ldots, n$: substract $q_j = \frac{a_j}{a_{11}}$ times the first row from the j-th row.
- The result of these operations is a matrix of the form

$$C = \begin{pmatrix} c_{1,1} & c_{1,2} & \ldots & c_{1,n} \\ 0 & c_{2,2} & \ldots & c_{2,n} \\ \vdots & \vdots & & \vdots \\ 0 & c_{2,n} & \ldots & c_{n,n} \end{pmatrix}.$$

Gauss Elimination

The first step of the Gauss elimination procedure can also be summarized in the form:

$$C = G_1 P_1 A \quad \text{and} \quad d = G_1 P_1 b$$

such that the equation system $Ax = b$ is equivalent to the equation system $Cx = d$. Here, P_1 is a permutation matrix (such that $P_1^2 = I$) and G_1 a Frobenius matrix of the form

$$G_1 = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ q_2 & 1 & 0 & \dots & 0 \\ q_3 & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ q_n & 0 & \dots & 0 & 1 \end{pmatrix} .$$

Gauss Elimination

In the second step of the Gauss elimination we apply the same transformation strategy to the remaining $(n-1) \times (n-1)$ dense block such that

$$E = G_2 P_2 G_1 P_1 A \quad \text{and} \quad f = G_2 P_2 G_1 P_1 b$$

such that the equation system $Ax = b$ is equivalent to the equation system $Ex = f$. Here, E has the form

$$E = \begin{pmatrix} c_{1,1} & c_{1,2} & c_{1,3} & \dots & c_{1,n} \\ 0 & e_{2,2} & e_{2,3} & \dots & e_{2,n} \\ 0 & 0 & e_{3,3} & \dots & e_{3,n} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & e_{n,3} & \dots & e_{n,n} \end{pmatrix} .$$

Gauss Elimination

If we keep on applying the same strategy we end up with a triangular matrix

$$R = G_{n-1}P_{n-1}\ldots G_1P_1A\ .$$

and a vector $s = G_{n-1}P_{n-1}\ldots G_1P_1b$ such that the equation system $Ax = b$ is equivalent to the equation system $Rx = s$. The triangular system can then be solved by backwards substitution. The complexity of computing R is of order $\mathbf{O}(n^3)$.

LR Decomposition

We assume for simplicity that we do not have to permute the rows of A.

The matrix

$$L = G_1^{-1} G_2^{-1} \dots G_{n-1}^{-1} = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ l_{2,1} & 1 & 0 & \dots & 0 \\ l_{3,1} & l_{3,2} & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ l_{n,1} & l_{n,2} & \dots & l_{n,n-1} & 1 \end{pmatrix}$$

is lower triangular and satisfies $LR = A$ by construction.

Uniqueness of LR-Decomposition

The LR-Decomposition is unique: if there were two LR Decompositions

$$A = L_1 R_1 = L_2 R_2$$

we would have

$$L_2^{-1} L_1 = R_2 R_1^{-1} = I \ ,$$

since the matrix $L_2^{-1} L_1$ is lower triangular (with ones on the diagonal) and R is an upper triangular matrix. Thus, we have $R_1 = R_2$ and $L_1 = L_2$.

Application of LR Decomposition in Practice

In practice, we are often in the situation that we want to solve mutiple linear equation systems of the form

$$Ax_i = b_i \;, \qquad i = 1, 2, \ldots, N$$

for changing vectors $b_i \in \mathbb{R}^m$. A naive implementation of Gauss elimination would lead the complexity $\mathbf{O}(N \cdot n^3)$ for computing $x_1, \ldots, x_N$.

A major speed-up can be obtained by computing if we store the LR-decomposition $A = LR$ of the matrix A and then compute

$$x_1 = R^{-1}L^{-1}b_1 \;, \quad x_2 = R^{-1}L^{-2}b_2 \;. \;\ldots$$

Now, the complexity is $\mathbf{O}(n^3 + N \cdot n^2)$.

Computation of Determinants

If a LR-decomposition of the matrix $A \in \mathbb{R}^{n\times n}$ is known, the determinant can be computed from

$$\det(A) = \det(LR) = \det(L)\det(R) = \det(R) = \prod_{i=1}^{n} R_{ii} \ .$$

Contents

Band-Structured Matrices

A band-structured matrix is a matrix whose components satisfy

$$a_{j,k} = 0 \quad \text{for} \quad k < j - m_{\mathsf{l}} \quad \text{or} \quad k > j + m_{\mathsf{r}} \; .$$

Here m_{l} and m_{r} are called the widths of the band-matrix A.

Examples:

- For $m_{\mathsf{l}} = 0$ and $m_{\mathsf{r}} = n - 1$: upper triangular matrix.
- For $m_{\mathsf{l}} = 0$ and $m_{\mathsf{r}} = n - 1$: lower triangular matrix.
- For $m_{\mathsf{l}} = 1$ and $m_{\mathsf{r}} = 1$: tridiagonal matrix.

Gauss Elimination for Band-Structured Matrices

If we do not have to permute rows, equation systems of the form

$$Ax = b$$

can be solved with Gauss-elimination, too, while exploiting the band-structure. The complexity is in this case

$$\mathbf{O}(nm_{\mathsf{l}}m_{\mathsf{r}}) + \mathbf{O}(n(m_{\mathsf{l}} + m_{\mathsf{r}}))\ .$$

Example: Tridiagonal Matrices

For the case that A is a tridiagonal matrix of the form

$$A = \begin{pmatrix} a_1 & b_1 & & \\ c_2 & \ddots & \ddots & \\ & \ddots & \ddots & b_{n-1} \\ & & c_n & a_n \end{pmatrix}$$

the matrices L and R have the form

$$L = \begin{pmatrix} 1 & & & \\ \gamma_2 & \ddots & & \\ & \ddots & 1 & \\ & & \gamma_n & 1 \end{pmatrix} \quad \text{and} \quad R = \begin{pmatrix} \alpha_1 & b_1 & & \\ & \ddots & \ddots & \\ & & \alpha_{n-1} & b_{n-1} \\ & & & \alpha_n \end{pmatrix} .$$

Example: Tridiagonal Matrices

The coefficients α_i and γ_i can be computed by the recursion

$$\begin{aligned} & & \alpha_1 &= a_1 \,, \\ \gamma_i &= \tfrac{c_i}{\alpha_{i-1}} \,, \qquad \text{and} \qquad & \alpha_i &= a_i - \gamma_i b_i \end{aligned}$$

for $i = 2, \ldots, n$.

Application: computation of splines (see Lecture 3).

Contents

Symmetric Positive Definite Matrices

A matrix $A \in \mathbb{R}^{n \times n}$ is called symmetric positive definite if we have $A = A^T$ and

$$\forall v \in \mathbb{R}^n \setminus \{0\}, \qquad v^T A v > 0 \ .$$

Equation systems of the form $Ax = b$ with A being symmetric positive semi-definite are solved by representing A in the form

$$A = LDL^T \ ,$$

where $L \in \mathbb{R}^{n \times n}$ is a lower diagonal matrix with ones on the diagonal and $D \in \mathbb{R}^{n \times n}$ a diagonal matrix.

Cholesky Algorithm (LDL^T-Variant)

Let us work out the matrix product

$$LDL^T =$$

$$\begin{pmatrix} 1 & & & \\ L_{2,1} & 1 & & \\ L_{3,1} & L_{3,2} & 1 & \\ \vdots & \vdots & & \ddots \end{pmatrix} \begin{pmatrix} D_1 & & & \\ & D_2 & & \\ & & D_3 & \\ & & & \ddots \end{pmatrix} \begin{pmatrix} 1 & & & \\ L_{2,1} & 1 & & \\ L_{3,1} & L_{3,2} & 1 & \\ \vdots & \vdots & & \ddots \end{pmatrix}^T$$

$$= \begin{pmatrix} D_1 & & & \text{symmetric} \\ L_{2,1}D_1 & L_{2,1}^2 D_1 + D_2 & & \\ L_{3,1}D_2 & L_{3,1}L_{2,1}D_1 + L_{3,2}D_2 & L_{3,1}^2 D_1 + L_{3,2}^2 D_2 + D_3 & \\ \vdots & \vdots & & \ddots \end{pmatrix}$$

Cholesky Algorithm (LDL^T-Variant)

We want to determine L and D such that

$$A = \begin{pmatrix} D_1 & & & \text{symmetric} \\ L_{2,1}D_1 & L_{2,1}^2 D_1 + D_2 & & \\ L_{3,1}D_2 & L_{3,1}L_{2,1}D_1 + L_{3,2}D_2 & L_{3,1}^2 D_1 + L_{3,2}^2 D_2 + D_3 & \\ \vdots & \vdots & & \ddots \end{pmatrix}$$

- We first set $D_1 = A_{1,1} > 0$ (since A is positive definite).
- We can find $L_{2,1} = A_{2,1}/D_1$.
- The diagonal entry $D_2 = A_{2,2} - L_{2,1}^2 D_1 > 0$ must be positive.
- General recursion (for $i > j$):

$$D_{j,j} = A_{j,j} - \sum_{k=1}^{j-1} L_{j,k}^2 D_k \quad \text{and} \quad L_{i,j} = \frac{1}{D_j}\left(A_{i,j} - \sum_{k=1}^{j-1} L_{i,k}L_{j,k}D_k\right).$$

Cholesky Algorithm (LDL^T-Variant)

We want to determine L and D such that

$$A = \begin{pmatrix} D_1 & & & \text{symmetric} \\ L_{2,1}D_1 & L_{2,1}^2 D_1 + D_2 & & \\ L_{3,1}D_2 & L_{3,1}L_{2,1}D_1 + L_{3,2}D_2 & L_{3,1}^2 D_1 + L_{3,2}^2 D_2 + D_3 & \\ \vdots & \vdots & & \ddots \end{pmatrix}$$

- We first set $D_1 = A_{1,1} > 0$ (since A is positive definite).
- We can find $L_{2,1} = A_{2,1}/D_1$.
- The diagonal entry $D_2 = A_{2,2} - L_{2,1}^2 D_1 > 0$ must be positive.
- General recursion (for $i > j$):

$$D_{j,j} = A_{j,j} - \sum_{k=1}^{j-1} L_{j,k}^2 D_k \quad \text{and} \quad L_{i,j} = \frac{1}{D_j}\left(A_{i,j} - \sum_{k=1}^{j-1} L_{i,k}L_{j,k}D_k\right) .$$