

DEN: Linear algebra – numerical view

Algorithm (GEM: Gauss elimination method for reducing a full rank matrix to upper-triangular form, with partial pivoting)

Given: matrix $C = (c_{i,j})_{i,j=1}^{n,m}$ of real numbers, where $m \geq n$.

0. Set $k = 1$.

1. If $c_{i,k} = 0$ for all $i = k, \dots, n$, then $\text{rank}(A) < n$. The algorithm fails and stops.

Otherwise, do the “pivoting”: Among the rows $i = k, \dots, n$, choose the row k' that has the largest possible value of $|c_{i,k}|$. If $k' \neq k$, exchange rows k and k' .

The (new) number $c_{k,k}$ is called a “pivot”. Continue with step 2.

2. For $i = k + 1, \dots, n$ do the following:

If $c_{i,k} \neq 0$, then let $l_{i,k} = \frac{c_{i,k}}{c_{k,k}}$, set $c_{i,k} = 0$ and for $j > k$ replace $c_{i,j}$ with $c_{i,j} - l_{i,k}c_{k,j}$.

3. If $k < n$, increase k by one and go back to step 1.

Otherwise the algorithm stops.

The **output** is the matrix $(c_{i,j})_{i,j=1}^{n,m}$.

Note: The output has the form

$$\begin{pmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,n-1} & c_{1,n} & \cdots & c_{1,m} \\ 0 & c_{2,2} & \cdots & c_{2,n-1} & c_{2,n} & \cdots & c_{2,m} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & c_{n-1,n-1} & c_{n-1,n} & \cdots & c_{n-1,m} \\ 0 & 0 & \cdots & 0 & c_{n,n} & \cdots & c_{n,m} \end{pmatrix}.$$

Fact.

Gaussian elimination applied to an $n \times (n + c)$ matrix requires at most $\frac{2}{3}n^3 + (c - \frac{1}{2})n^2 - (c + \frac{1}{6})n$ operations.

Corollary.

The computational complexity of GEM when reducing an $n \times n$ matrix or an $n \times (n + 1)$ matrix is

$$\frac{2}{3}n^3 + O(n^2).$$

Algorithm (GJM: Gauss-Jordan elimination method for reducing a full-rank matrix to an extended diagonal matrix, with partial pivoting)

Given: matrix $C = (c_{i,j})_{i,j=1}^{n,m}$ of real numbers, where $m \geq n$.

0. Set $k = 1$.

1. If $c_{i,k} = 0$ for all $i = k, \dots, n$, then $\text{rank}(A) < n$. The algorithm fails and stops.

Otherwise, do the “pivoting”: Among the rows $i = k, \dots, n$, choose the row k' that has the largest possible value of $|c_{i,k}|$. If $k' \neq k$, exchange rows k and k' .

The (new) number $c_{k,k}$ is called a “pivot”. Continue with step 2.

2. For $j > k$ replace $c_{k,j}$ with $\frac{c_{k,j}}{c_{k,k}}$. Set $c_{k,k} = 1$.

For all $i \neq k$ do the following:

If $c_{i,k} \neq 0$, then for $j > k$ replace $c_{i,j}$ with $c_{i,j} - c_{i,k}c_{k,j}$; then set $c_{i,k} = 0$.

3. If $k < n$, increase k by one and go back to step 1.

Otherwise the algorithm stops.

The **output** is the matrix $(c_{i,j})_{i,j=1}^{n,m}$.

Note: The output has the form

$$\begin{pmatrix} 1 & 0 & \cdots & 0 & 0 & c_{1,n+1} & \cdots & c_{1,m} \\ 0 & 1 & \cdots & 0 & 0 & c_{2,n+1} & \cdots & c_{2,m} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 & 0 & c_{n-1,n+1} & \cdots & c_{n-1,m} \\ 0 & 0 & \cdots & 0 & 1 & c_{n,n+1} & \cdots & c_{n,m} \end{pmatrix}.$$

Fact.

Gauss-Jordan elimination applied to an $n \times (n+c)$ matrix requires at most $n^3 + \frac{1}{2}(4c-3)n^2 + \frac{1}{2}(1-2c)n$ operations.

Corollary.

The computational complexity of GJM when reducing an $n \times n$ matrix or an $n \times (n+1)$ matrix is

$$n^3 + O(n^2).$$

Algorithm (Solving an upper triangular system by back substitution)

Given: a system $U\vec{x} = \vec{d}$, where the matrix U is square, regular and upper triangular.

We find the solution \vec{x}_0 using the formulas

$$\begin{aligned} x_n &= \frac{d_n}{u_{n,n}} \\ x_{n-1} &= \frac{d_{n-1} - u_{n-1,n}x_n}{u_{n-1,n-1}} \\ x_{n-2} &= \frac{d_{n-2} - u_{n-2,n}x_n - u_{n-2,n-1}x_{n-1}}{u_{n-2,n-2}} \\ &\vdots \\ x_1 &= \frac{d_1 - u_{1,n}x_n - u_{1,n-1}x_{n-1} - \cdots - u_{1,2}x_2}{u_{1,1}} \end{aligned}$$

In general, for $k = n, n-1, \dots, 1$ we calculate

$$x_k = \frac{1}{u_{k,k}} \left(d_k - \sum_{i=k+1}^n u_{k,i}x_i \right).$$

Algorithm (Solving a lower triangular system by forward substitution)

Given: A system $L\vec{x} = \vec{d}$, where the matrix L is square, regular and lower triangular.

We find the solution \vec{x}_0 using the formulas

$$\begin{aligned} x_1 &= \frac{d_1}{l_{1,1}} \\ x_2 &= \frac{d_2 - l_{2,1}x_1}{l_{2,2}} \\ x_3 &= \frac{d_3 - l_{3,1}x_1 - l_{3,2}x_2}{l_{3,3}} \\ &\vdots \\ x_n &= \frac{d_n - l_{n,1}x_1 - l_{n,2}x_2 - \cdots - l_{n,n-1}x_{n-1}}{l_{n,n}} \end{aligned}$$

In general, for $k = 1, 2, \dots, n$ we calculate

$$x_k = \frac{1}{l_{k,k}} \left(d_k - \sum_{i=1}^{k-1} u_{k,i}x_i \right).$$

Fact.

Any system $A\vec{x} = \vec{b}$ whose regular matrix A is upper (resp. lower) triangular can be solved by back (resp. forward) substitution with computational complexity n^2 .

Iterative improvement of solution:

1. Find a solution \vec{x} of the system $A\vec{x} = \vec{b}$,
 2. Determine the residuum $\vec{r} = \vec{b} - A\vec{x}$. Solve the system $A\vec{E}_x = \vec{r}$.
- If the error \vec{E}_x is not sufficiently small, do the correction $\vec{x} := \vec{x} + \vec{E}_x$.

Definition.

Consider a real $n \times n$ matrix A . We say that $n \times n$ matrices L, U are an **LU decomposition** or **LU factorization** of A if U is an upper-triangular matrix, L is a lower-triangular matrix whose diagonal entries are all 1, and $A = LU$.

Theorem.

Let A be a real $n \times n$ matrix. If its rank is k and its first k leading principal minors are non-zero, then it has an LU decomposition.

Fact.

If GEM without pivoting applied to A yields an upper-triangular matrix, then A has an LU decomposition.

Moreover, if we create matrix L by inserting coefficients $l_{i,k}$ into a unit matrix of appropriate dimension, and denote by U the matrix resulting from this run of GEM, then L, U are the LU decomposition of A .

Definition.

Consider a real $n \times n$ matrix A . We say that $n \times n$ matrices L, U, P are an **LUP decomposition** or **LUP factorization** of A if U is an upper-triangular matrix, L is a lower-triangular matrix whose diagonal entries are all 1, P is a permutation matrix, and $PA = LU$.

Fact.

For every square matrix A there exists an LUP decomposition.

Algorithm (LUP decomposition of a matrix, with partial pivoting)

Given: an $n \times n$ matrix $A = (a_{i,j})_{i,j=1}^n$ of real numbers.

0. Let $U = A$ and $L = P = E_n$ (unit matrix). Set $k = 1, l = 1$.

1. If $u_{k,l} = 0$ for all $i \geq k$, then go to step **3**. Otherwise:

Among the rows $i = k, \dots, n$, choose the row k' that has the largest possible value of $|u_{i,l}|$. If $k' > k$, then exchange rows k and k' in matrices U and P ; in the matrix L exchange the first $k - 1$ entries of rows k and k' .

Continue with step **2**.

2. For $i = k + 1, \dots, n$ do the following: Let $l_{i,k} = \frac{u_{i,l}}{u_{k,l}}$, set $u_{i,l} = 0$ and for $j > l$ replace $u_{i,j}$ with $u_{i,j} - l_{i,k}u_{k,j}$.

Increase k by one and continue with step **3**.

3. If $l < n$, increase l by one and go back to step **1**.

Otherwise the algorithm stops.

The **output** is the matrices P, L, U .

Algorithm (solving systems of linear equations using LUP factorization)

Given: a system $A\vec{x} = \vec{b}$, where A is a regular square matrix.

1. Find the LUP factorization $LU = AP$.
2. Using the forward substitution, solve the system $L\vec{y} = P\vec{b}$ for \vec{y} .
3. Using the back substitution, solve the system $U\vec{x} = \vec{y}$ for \vec{x} .

$$A \mapsto L, U, P \quad (L|P\vec{b}) \xrightarrow{\text{FS}} \vec{y} \quad (U|\vec{y}) \xrightarrow{\text{BS}} \vec{x}$$

Definition.

Let V be a vector space. A mapping $\|\cdot\|: V \mapsto \mathbb{R}$ is called a **norm** if it has the following properties:

- $\|\vec{x}\| \geq 0$ for all $\vec{x} \in \mathbb{R}^n$;
- $\|\vec{x}\| = 0$ if and only if $\vec{x} = \vec{0}$;
- $\|c\vec{x}\| = |c| \cdot \|\vec{x}\|$ for all $\vec{x} \in \mathbb{R}^n$ and $c \in \mathbb{R}$;
- $\|\vec{x} + \vec{y}\| \leq \|\vec{x}\| + \|\vec{y}\|$ for all $\vec{x}, \vec{y} \in \mathbb{R}^n$ (triangle inequality).

Traditional norms on \mathbb{R}^n :

$$\|\vec{x}\| = \sqrt{\sum_{k=1}^n |x_k|^2} \quad (\text{Euclidean norm}),$$

$$\|\vec{x}\|_\infty = \max_{k=1, \dots, n} |x_k| \quad (\text{max norm}),$$

$$\|\vec{x}\|_1 = \sum_{k=1}^n |x_k| \quad (\text{sum norm}).$$

Definition.

Let V be a vector space of matrices. A mapping $\|\cdot\|: V \mapsto \mathbb{R}$ is called a **matrix norm** if it is a norm and also satisfies

- $\|AB\| \leq \|A\| \cdot \|B\|$ for all $A, B \in M_{n \times n}$.

Traditional matrix norms:

$$\|A\|_\infty = \|A\|_R = \max_{i=1, \dots, n} \sum_{j=1}^n |a_{i,j}| \quad (\text{row-sum norm}),$$

$$\|A\|_1 = \|A\|_C = \max_{j=1, \dots, n} \sum_{i=1}^n |a_{i,j}| \quad (\text{column-sum norm}),$$

$$\|A\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^n |a_{i,j}|^2} \quad (\text{Frobenius norm}).$$

Definition.

Consider a norm $\|\vec{x}\|$ for vectors from \mathbb{R}^n and a norm $\|A\|_M$ for matrices from $M_{n \times n}$. We say that these norms are **compatible** if $\|A\vec{x}\| \leq \|A\|_M \cdot \|\vec{x}\|$ for all $A \in M_{n \times n}$ and $\vec{x} \in \mathbb{R}^n$.

Theorem.

(i) Let $\|\cdot\|$ be a vector norm. The number defined for $A \in M_{n \times n}$ by the formula

$$\|A\|_M = \sup \left\{ \frac{\|A\vec{x}\|}{\|\vec{x}\|}; \vec{x} \in \mathbb{R}^n \setminus \{\vec{0}\} \right\} = \sup \{ \|A\vec{x}\|; \vec{x} \in \mathbb{R}^n \wedge \|\vec{x}\| \leq 1 \}$$

determines a matrix norm compatible with $\|\cdot\|$. We call it the matrix norm **induced** by $\|\cdot\|$.

(ii) Let $\|\cdot\|_M$ be a matrix norm. The number defined for $\vec{x} \in \mathbb{R}^n$ by the formula

$$\|\vec{x}\| = \left\| \begin{pmatrix} x_1 & 0 & \dots & 0 \\ x_2 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ x_n & 0 & \dots & 0 \end{pmatrix} \right\|_M$$

is a norm on \mathbb{R}^n compatible with $\|\cdot\|_M$. It is called the norm induced by $\|\cdot\|_M$.

Definition.

We define the **spectral radius** of A as $\rho(A) = \max_j (|\lambda_j|)$, where the maximum runs through all eigenvalues λ_j of A (including possible complex ones).

Fact.

Consider an $n \times n$ matrix A .

The inequality $\rho(A) \leq \|A\|$ is true for all induced matrix norms.

Conversely, for every $\varepsilon > 0$ there is an induced matrix norm $\|\cdot\|$ such that $\|A\| - \varepsilon < \rho(A)$.

Definition.

For an $n \times n$ matrix A we define its **condition number** as $\text{cond}(A) = \|A\| \cdot \|A^{-1}\|$.

Theorem.

Let \vec{x}_0 be a solution of the system $A\vec{x} = \vec{b}$ and \vec{x} its approximation. Set $\vec{E}_x = \vec{x}_0 - \vec{x}$, let $\vec{r} = \vec{b} - A\vec{x}$ be the residual of \vec{x} . Then

$$\frac{\|\vec{E}_x\|}{\|\vec{x}\|} \leq \text{cond}(A) \frac{\|\vec{r}\|}{\|\vec{b}\|}.$$

That is,

$$\varepsilon_x \leq \text{cond}(A)\varepsilon_r.$$

Fact.

If a vector \vec{b} is reliable to k significant digits, then the solution \vec{x} of an equation $A\vec{x} = \vec{b}$ is reliable to $k - \log_{10}(\text{cond}(A))$ significant digits.

Theorem.

Assume that vectors \vec{x}_0, \vec{x} and \vec{b}_0, \vec{b} are related by the formulas $A\vec{x}_0 = \vec{b}_0$ and $A\vec{x} = \vec{b}$. Denote $\vec{E}_x = \vec{x}_0 - \vec{x}$ and $\vec{E}_b = \vec{b}_0 - \vec{b}$. Then

$$\varepsilon_x \leq \text{cond}(A)\varepsilon_b.$$

Theorem.

Assume that matrices A_0, A and vectors \vec{x}_0, \vec{x} and \vec{b}_0, \vec{b} are related by the formulas $A_0\vec{x}_0 = \vec{b}_0$ and $A\vec{x} = \vec{b}$. Denote $E_A = A_0 - A$, $\vec{E}_x = \vec{x}_0 - \vec{x}$ and $\vec{E}_b = \vec{b}_0 - \vec{b}$. Then

$$\varepsilon_x \leq \text{cond}(A) \left(\varepsilon_b + \varepsilon_A \cdot \frac{\|\vec{x}\|}{\|\vec{x}_0\|} \right).$$

Theorem.

If a matrix B satisfies $\|B\|_M < 1$ for some compatible matrix norm, then the corresponding iterative method $\vec{x}_{k+1} = B\vec{x}_k + \vec{c}$ converges to \vec{x}_f for arbitrary choice of \vec{x}_0 and we have

$$\|\vec{x}_f - \vec{x}_{k+1}\| \leq \|B\|_M \|\vec{x}_f - \vec{x}_k\|, \quad \|\vec{x}_f - \vec{x}_{k+1}\| \leq \frac{\|B\|_M}{1 - \|B\|_M} \|\vec{x}_{k+1} - \vec{x}_k\|.$$

Theorem.

An iterative method $\vec{x}_{k+1} = B\vec{x}_k + \vec{c}$ converges if and only if $\rho(B) < 1$.

Algorithm (JIM: Jacobi iteration method)

Given: a system $A\vec{x} = \vec{b}$ of linear equations and tolerance ε .

0. Choose arbitrary initial vector \vec{x}_0 . Set $k = 0$.

1. Compute

$$(\vec{x}_{k+1})_i = -\frac{1}{a_{i,i}} \left(\sum_{j=1}^{i-1} a_{i,j}(\vec{x}_k)_j + \sum_{j=i+1}^n a_{i,j}(\vec{x}_k)_j \right) + \frac{b_i}{a_{i,i}}.$$

If $\|\vec{x}_{k+1} - \vec{x}_k\|_\infty \geq \varepsilon$, increase k by one and go back to step 1.

Algorithm (GSM: Gauss-Seidel iteration)

Given: a system $A\vec{x} = \vec{b}$ of linear equations and tolerance ε .

0. Choose arbitrary initial vector \vec{x}_0 . Set $k = 0$.

1. Compute

$$(\vec{x}_{k+1})_i = -\frac{1}{a_{i,i}} \left(\sum_{j=1}^{i-1} a_{i,j} (\vec{x}_{k+1})_j + \sum_{j=j+1}^n a_{i,j} (\vec{x}_k)_j \right) + \frac{b_i}{a_{i,i}}.$$

If $\|\vec{x}_{k+1} - \vec{x}_k\|_\infty \geq \varepsilon$, increase k by one and go back to step **1**.

Definition.

Consider an $n \times n$ matrix A .

We say that A is **strictly diagonally dominant** if

$$|a_{i,i}| > \sum_{j \neq i} |a_{i,j}|$$

for all $i = 1, \dots, n$.

We say that A is **positive definite** if $\vec{x}^T A \vec{x} > 0$ for all non-zero vectors $\vec{x} \in \mathbb{R}^n$.

Theorem.

If A is strictly diagonally dominant, then both JIM and GSM converge for arbitrary choice of initial vector.

If A is symmetric and positive definite, then GSM converges for arbitrary choice of initial vector.

Algorithm (Relaxation (SOR, Successive OverRelaxation method))

Given: a system $A\vec{x} = \vec{b}$ of linear equations, tolerance ε , and parameter of relaxation ω .

0. Choose arbitrary initial vector \vec{x}_0 . Set $k = 0$.

1. Compute

$$(\vec{x}_{k+1})_i = (1 - \omega)(\vec{x}_k)_i - \frac{\omega}{a_{i,i}} \left(\sum_{j=1}^{i-1} a_{i,j} (\vec{x}_{k+1})_j + \sum_{j=j+1}^n a_{i,j} (\vec{x}_k)_j \right) + \frac{\omega b_i}{a_{i,i}}.$$

If $\|\vec{x}_{k+1} - \vec{x}_k\|_\infty \geq \varepsilon$, increase k by one and go back to step **1**.

Theorem. (Ostrovsky)

Let A be a symmetric $n \times n$ matrix with positive diagonal entries. Then $\rho(B_\omega) < 1$ if and only if A is positive definite and $0 < \omega < 2$.

Definition.

Consider an $n \times n$ matrix A . A number λ is called an **eigenvalue** of A if there is a non-zero vector \vec{x} such that $A\vec{x} = \lambda\vec{x}$. Vectors \vec{x} with this property are then called **eigenvectors** associated with λ .

Fact.

If a real $n \times n$ matrix is symmetric, then its eigenvalues are real.
Moreover, there exists a basis of \mathbb{R}^n composed of its eigenvectors.

Algorithm (power method for finding the largest eigenvalue and an associated eigenvector)

Given: an $n \times n$ matrix A and tolerance $\varepsilon > 0$.

0. Choose arbitrary initial vector \vec{x}_0 and set $k = 0$.

1. Compute

$$\vec{x}_{k+1} = \frac{A\vec{x}_k}{\|A\vec{x}_k\|_\infty}.$$

If $\|\vec{x}_{k+1} - \vec{x}_k\|_\infty \geq \varepsilon$, increase k by one and go back to step 1.

Definition.

For an $n \times n$ matrix A and a vector $\vec{x} \in \mathbb{R}^n$ we define their **Rayleigh quotient** by the formula

$$\frac{\vec{x}^T A \vec{x}}{\vec{x}^T \vec{x}}.$$

Fact.

If \vec{x} is an eigenvector of a matrix A , then $\frac{\vec{x}^T A \vec{x}}{\vec{x}^T \vec{x}}$ is equal to the associated eigenvalue.

Algorithm (power method for finding the largest eigenvalue and an associated eigenvector)

Given: an $n \times n$ matrix A and tolerance $\varepsilon > 0$.

0. Choose arbitrary initial vector \vec{x}_0 .

If complex eigenvalues are needed, choose a vector with non-zero imaginary part in each component.

Set $k = 0$, let $l_0 = \frac{\vec{x}_0^* A \vec{x}_0}{\vec{x}_0^* \vec{x}_0}$.

1. Compute

$$\vec{y}_{k+1} = \frac{1}{l_k} A \vec{x}_k, \quad \vec{x}_{k+1} = \frac{\vec{y}_{k+1}}{\|\vec{y}_{k+1}\|_\infty}, \quad l_{k+1} = \frac{\vec{x}_{k+1}^* A \vec{x}_{k+1}}{\vec{x}_{k+1}^* \vec{x}_{k+1}}.$$

If $\|\vec{x}_{k+1} - \vec{x}_k\|_\infty \geq \varepsilon$, increase k by one and go back to step 1.

Alternative: Use Euclidean norm, that is,

$$\vec{x}_{k+1} = \frac{|\lambda_k| \cdot A \vec{x}_k}{\lambda_k \cdot \|A \vec{x}_k\|_2}, \quad l_{k+1} = \vec{x}_{k+1}^* A \vec{x}_{k+1}.$$

Theorem.

Let A be an $n \times n$ matrix with eigenvalues λ_j , where $|\lambda_1| > |\lambda_j|$ for $j \geq 2$. Let $M = \max_{j \geq 2} |\lambda_j|$.

For \vec{x}_0 let $\{l_k\}$, $\{\vec{x}_k\}$ be sequences generated by the power method. Then $\{l_k\}$ converges to λ_1 and $\{\vec{x}_k\}$ converges to some eigenvector \vec{v} associated with λ_1 .

Moreover, $|\lambda_1 - l_k| = O\left(\left[\frac{M}{|\lambda_1|}\right]^{2k}\right)$ and $\|\vec{v}_1 - \vec{x}_k\| = O\left(\left[\frac{M}{|\lambda_1|}\right]^k\right)$.

Algorithm (inverse power method for finding eigenvalue and an associated eigenvector)

Given: an $n \times n$ matrix A , arbitrary real number μ that is not an eigenvalue of A and tolerance $\varepsilon > 0$.

1. Set $B = (A - \mu E_n)^{-1}$.

2. Applying power iteration to the matrix B , find the largest eigenvalue λ_B of the matrix B and an associated eigenvector \vec{v} .

Output: The number $\mu + \frac{1}{\lambda_B}$ is the eigenvalue of A closest to μ with eigenvector \vec{v} .

Algorithm (deflation method for finding eigenvalue and an associated eigenvector)

Given: a symmetric $n \times n$ matrix A and its eigenvalues λ_1 through λ_N with orthogonal eigenvectors \vec{v}_1 through \vec{v}_N satisfying $\|v_i\| = 1$. Given tolerance $\varepsilon > 0$.

1. Set up the matrix $B = A - \sum_{j=1}^N \lambda_j \vec{v}_j \cdot \vec{v}_j^T$.
2. Apply power iteration to the matrix B and obtain λ_{N+1} , \vec{x}_{N+1} .