# MPI - Lecture 10

## Conditioning and stability of an algorithm

### Recap

---

Recap: Errors

**Definition 1.** *Let a number $\alpha$ be an approximate value of a number $a$.*

- *The **absolute error** is the value $|\alpha - a|$.*
- *For $a \neq 0$, the **relative error** is $\dfrac{|\alpha - a|}{|a|}$.*

### Systems of linear equations

---

System of linear equations

We want to solve a system of $n$ linear equations. We write the system in matrix representation

$$Ax = b,$$

where $A \in \mathbb{R}^{n,n}$ is regular and $b \in \mathbb{R}^{n,1}$.

This is often a partial subproblem of a larger problem.

---

Example: system of linear equations (1/2)

Consider two systems of linear equations with $2$ unknowns:

$$\begin{pmatrix} 1 & 1/2 \\ 1/2 & 1/3 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 3/2 \\ 1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 1 & 1/5 \\ 1/5 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 3/2 \\ 1 \end{pmatrix}.$$

The solutions are

$$(x, y)^T = (0, 3)^T \quad \text{and} \quad (x, y)^T = (85/52, -35/52)^T \approx (1.6346, -0.67308)^T.$$

Let us try to simulate an error on the input, or during a calculation, by changing the right side to $\begin{pmatrix} 3/2 \\ 5/6 \end{pmatrix}$.

$$\begin{pmatrix} 1 & 1/2 \\ 1/2 & 1/3 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 3/2 \\ 5/6 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 1 & 1/5 \\ 1/5 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 3/2 \\ 5/6 \end{pmatrix}.$$

The solutions change to

$$(x,y)^T = (1,1)^T \quad \text{and} \quad (x,y)^T = (125/78, -20/39)^T \approx (1.6026, -0.51282)^T.$$

---

Example: system of linear equations (2/2)

The change in the right side was

$$\begin{pmatrix} 3/2 \\ 1 \end{pmatrix} - \begin{pmatrix} 3/2 \\ 5/6 \end{pmatrix} = \begin{pmatrix} 0 \\ 1/6 \end{pmatrix},$$

a vector of Euclidean length $1/6$ (the relative error is $0.09$).

The change in the solution of **the first equation** was

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

(the relative error is $0.75$) and the one in the solution of **the second equation**

$$\begin{pmatrix} 85/52 \\ -35/52 \end{pmatrix} - \begin{pmatrix} 125/78 \\ -20/39 \end{pmatrix} = \begin{pmatrix} 5/156 \\ -25/156 \end{pmatrix}$$

(the relative error is $0.09$).

Why is it that the first system is more sensitive to this change? Why are the two relative errors so different?

---

Norm - reminder

A **norm** on a vector space $V$ is a mapping $\|\cdot\| : V \mapsto \mathbb{R}_0^+$ which satisfies

1. $\|x\| = 0 \quad \Rightarrow \quad x = 0$,
2. $\|\alpha x\| = |\alpha| \cdot \|x\|$,
3. $\|x + y\| \leq \|x\| + \|y\|$ (triangle inequality),

for all $x, y \in V$ and all scalars $\alpha$.

On $\mathbb{R}^n$ (or $\mathbb{C}^n$) the most used norm is probably the **Euclidean norm**:

$$\|x\| = \left(\sum_{i=1}^{n} x_i^2\right)^{\frac{1}{2}},$$

where $x = (x_1, x_2 \ldots, x_n) \in \mathbb{R}^n$.

Other commonly used norms include

- $\|x\|_\infty = \max\{|x_i| : i \in \{1, \ldots, n\}\}$ maximum norm,
- $\|x\|_1 = \sum_{i=1}^{n} |x_i|$ taxicab or $L_1$ norm.

---

Matrix norm

Given a vector norm $\|\cdot\|$, we define the **induced matrix norm** of the matrix $A \in \mathbb{R}^{n,n}$ (or for $A \in \mathbb{C}^{n,n}$) as follows

$$\|A\| = \sup\{\|Ax\| : x \in \mathbb{R}^{n,1} \text{ and } \|x\| = 1\}.$$

Such norm satisfies

- $\|I\| = 1$,
- $\|Ax\| \leq \|A\| \cdot \|x\|$ (norm consistency),
- $\|AB\| \leq \|A\| \cdot \|B\|$.

## Forward and backzard error

---

Forward and backward error

Let $V$ be a numerical algorithm whose theoretical (accurate) output is denoted by $V^*(d)$ where $d$ is the input.

The result in the finite arithmetic is denoted $V(d)$. Furthermore, denote the so-called **forward error** by $\Delta v := V^*(d) - V(d)$.

The least (in a norm) number $\Delta d$ such that $V(d + \Delta d) = V^*(d)$ is the **backward error**.

$$\begin{tikzcd}
d+\Delta d \arrow[r, "V"] & V^*(d) \\
d \arrow[u, "\Delta d"] \arrow[r, "V"'] \arrow[ur, dashed, "V^*"'] & V(d) \arrow[u, "\Delta v"']
\end{tikzcd}$$

If for every input $d$ the backward error is relatively small, we say that the algorithm is **backward stable**.

("Small" depends on the context.)

## Conditioning

Conditioning

The **conditioning** of a problem expresses the dependence of the output on the inputs - given a little perturbation $\delta d$ of the input, we look how the output changes.

The **relative condition number** of a problem is

$$C_r = \lim_{\varepsilon \to 0^+} \sup_{\substack{d+\delta d \in D \\ \|\delta d\| \leq \varepsilon}} \frac{\frac{\|V(d+\delta d) - V(d)\|}{\|V(d)\|}}{\frac{\|\delta d\|}{\|d\|}},$$

where $D$ is the domain of $V$.

If $C_r \approx 1$, then we say that the problem is **well-conditioned**.

If it is large, we say the problem is **ill-conditioned**.

## Conditioning of the problem

Conditioning of the problem: System of linear equations

Let us see the conditioning of $Ax = b$. We suppose that the right side $b$ is the input of the problem, and $x$ is the output.

Given a small perturbation $\delta x$ we have:

$$A(x + \delta x) = Ax + A\delta x = b + \delta b,$$

where $A\delta x = \delta b$.

We have $\|b\| = \|Ax\| \leq \|A\| \cdot \|x\|$, which implies $\dfrac{1}{\|x\|} \leq \dfrac{\|A\|}{\|b\|}$.

Furthermore, $\|\delta x\| = \|A^{-1}\delta b\| \leq \|A^{-1}\| \cdot \|\delta b\|$.

Finally,

$$\frac{\|\delta x\|}{\|x\|} \leq \|A\| \cdot \|A^{-1}\| \frac{\|\delta b\|}{\|b\|},$$

---

Conditioning of the problem: System of linear equations

$$\frac{\|\delta x\|}{\|x\|} \leq \left(\|A\| \cdot \|A^{-1}\|\right) \frac{\|\delta b\|}{\|b\|}$$

The number $\kappa(A) = \|A\| \cdot \|A^{-1}\|$ is the **condition number** of the matrix $A$.

The above inequality reads: the relative error of the results is less than the relative error of the input times the condition number.

The greater $\kappa(A)$ is, the more ill-conditioned the problem is.

(Note that $b$ may contain an error coming from its origin, for instance a measurement.)

Of course, the condition number depends on the chosen norm.

---

Example of two sets of linear equations revisited

Let us revisit the example we saw earlier:

$$A_1 = \begin{pmatrix} 1 & 1/2 \\ 1/2 & 1/3 \end{pmatrix} \quad \text{and} \quad A_2 = \begin{pmatrix} 1 & 1/5 \\ 1/5 & -1 \end{pmatrix},$$

The inverses are

$$A_1^{-1} = \begin{pmatrix} 4 & -6 \\ -6 & 12 \end{pmatrix} \quad \text{and} \quad A_2^{-1} \approx \begin{pmatrix} 0.961538 & 0.192308 \\ 0.192308 & -0.961538 \end{pmatrix},$$

To calculate the condition number $\kappa(A) = \|A\| \cdot \|A^{-1}\|$ we use the norm $\|A\|_\infty$:

$$\kappa(A_1) = \frac{3}{2} \cdot 18 = 27 \quad \text{and} \quad \kappa(A_2) = \frac{18}{13} \approx 1.3846056.$$

The problem with the matrix $A_1$ is significantly more **ill-conditioned** than with the matrix $A_2$. This is in accordance with our previous observations.

## Direct and iterative methods

### Directive methods

---

Direct methods

A **direct method** calculates a solution of a problem in finitely many steps such that in absolute theoretical precision in gives the exact solution.

### Iterative methods

---

Idea of iterative methods

**Iterative methods** look for approximate solutions to mathematical problems by constructing a sequence of approximate solutions:

$$x_0,\ x_1,\ x_2,\ \ldots$$

Every following (approximate) solution is derived from the previous:

$$x_k = T(x_{k-1}),$$

for $k > 0$ and some mapping $T$.

The mapping $T$ is chosen so that the sequence $(x_i)$ has a limit which is the (exact) solution of the problem.

If $T$ is the same for all $k$, the method is called **stationary**.

## Description of the iterative method

Basic iterative methods for $Ax = b$

We will construct a sequence of vectors $x_0, x_1, x_2, \ldots$ which will approximate the solution of $Ax = b$.

The vector $x_0$ is chosen randomly.
We choose a regular matrix $Q$ and the following terms will be calculated as

$$Qx_k = (Q - A)x_{k-1} + b$$

for all $k > 0$.

**The idea**: choose the matrix $Q$ so that the sequence $(x_k)$ converges to some $x^*$. Then,

$$Qx^* = (Q - A)x^* + b$$

and thus

$$Ax^* = b.$$

Convergence - choice of $Q$

We use the equality $x_k = Q^{-1}\big((Q - A)x_{k-1} + b\big)$ in

$$\begin{aligned} x_k - x &= Q^{-1}\big((Q - A)x_{k-1} + b\big) - x \\ &= (I - Q^{-1}A)x_{k-1} - x + Q^{-1}b \\ &= (I - Q^{-1}A)x_{k-1} - (I - Q^{-1}A)x \\ &= (I - Q^{-1}A)(x_{k-1} - x), \end{aligned}$$

where $x$ is the exact solution satisfying $Ax = b$.

Denote $W = I - Q^{-1}A$ and the **error vector** $e_k = x_k - x$.

We have $e_k = We_{k-1}$.

The vector $e_k$ will be "smaller" than $e_{k-1}$ if $W$ is "small".
("Small" can be determined using norms.)

Since $e_k = W^k e_0$, to lower the error at each step we need to have $\lim\limits_{k\to+\infty} W^k = 0$.

## Convergence

Convergence vs. spectral radius

The **Spectral radius** of a matrix $M$ is the number $\rho(M)$ defined as the greatest eigenvalues (in absolute value), i.e.,

$$\rho(M) = \max\{|\lambda| \colon \lambda \text{ is an eigenvalue of } M\},$$

**Theorem 2.** *If $M \in \mathbb{C}^{n,n}$, then*

$$\lim_{k\to+\infty} M^k = 0 \quad \Leftrightarrow \quad \rho(M) < 1,$$

Thus, in our case, the method converges **if and only if**

$$\rho(W) < 1,$$

i.e., all the eigenvalues of the matrix $W = I - Q^{-1}A$ are in absolute value less than $1$.

Speed of convergence of $e_k$

How fast is the error vector $e_k$ converging to $0$?

We have

$$e_k = W^k e_0.$$

We estimate in norm

$$\|e_k\| = \left\|W^k e_0\right\| \leq \left\|W^k\right\| \cdot \|e_0\| \leq \|W\|^k \cdot \|e_0\| .$$

The condition of convergence $\rho(W) < 1$ does not imply anything on the speed from the previous estimate.

However, the estimate on the right side is strictly decreasing if $\|W\| < 1$.

When to stop? (1/2)

The iterative method is stop at the step $k$ if $x_k$ reaches some desired precision.

(The desired precision is given by the nature of the problem.)

In the case $\|W\| < 1$, we know that the sequence $(\|e_k\|)_k$ is strictly decreasing and we may stop iterating when

$$\|e_k - e_{k-1}\| < \varepsilon,$$

where $\varepsilon$ is a constant supplied by the user.

This is impractical since we do not have the exact solution.

In the step $k$ we can calculate the so-called **residue** $Ax_k - b$ and the **convergence criterion** can be set to

$$\|Ax_k - b\| < \varepsilon.$$

---

When to stop? (2/2)

Instead of calculating the residues, one may use a more efficient criterion

$$\|x_{k+1} - x_k\| < \varepsilon.$$

We have

$$\begin{aligned}\|e_k\| = \|x_k - x\| &= \|x_k - x_{k+1} + x_{k+1} - x\| \\ &\le \|x_k - x_{k+1}\| + \|\underbrace{x_{k+1} - x}_{=e_{k+1}}\| \\ &< \varepsilon + \|W\| \cdot \|e_k\|,\end{aligned}$$

where, supposing $\|W\| < 1$, the last inequality gives

$$\|e_k\| < \frac{\varepsilon}{1 - \|W\|}.$$

Thus, this criterion can be effectively used if $\|W\| < 1$, but not too close to $1$.

---

Finite precision calculations

All ideas so far were made in the **theoretical absolute precision**.

In **finite precision** the method may not converge even if $\|W\| < 1$ due to rounding errors.

However, an advantage of iterative methods in a finite precision arithmetic is that at each step the rounding errors from the previous step are "forgotten". We start the new iteration with a better approximate solution.

In finite arithmetic the method can diverge even if the problem is not ill-conditioned.

Thus, in practice, we need another parameter of the method - a maximum number of iterations. If we reach this number of iterations without satisfying a convergence criterion, then the method outputs failure.

## Concrete algorithms

Choices of $Q$

---

Denote by $a_{i,j}$ the entries of the matrix $A$ and denote

$$L = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ a_{2,1} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ a_{n,1} & \cdots & a_{n,n-1} & 0 \end{pmatrix} \text{ and } D = \begin{pmatrix} a_{1,1} & 0 & \cdots & 0 \\ 0 & a_{2,2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a_{n,n} \end{pmatrix}.$$

Denote $U$ so that $A = L + D + U$.

We will mention the following choices of $Q$:

- Richardson method $Q = I$,
- Jacobi method $Q = D$,
- successive overrelaxation / SOR method $Q = \frac{1}{\omega} D + L$.

Richardson method

---

Set $Q = I$.

The recurrence relation is given by

$$x_k = (I - A)x_{k-1} + b$$

The convergence is for a narrow class of matrices: $A$ must be close to $I$ so that

$$\|I - A\| < 1.$$

Jacobi method

Set $Q = D$.

The recurrence relation is given by

$$Dx_k = (D - A)x_{k-1} + b = -(L + U)x_{k-1} + b.$$

**Proposition 3.** *If the matrix $A$ is diagonally dominant, then the Jacobi method converges for any choice of $x_0$.*

A matrix is diagonally dominant if, for each row, the sum of the absolute values of all the entries except the one on the diagonal is less than the absolute value of the entry on the diagonal.

SOR method

Set $Q = \frac{1}{\omega}D + L$, where $\omega \in \mathbb{R} \setminus \{0\}$.

The recurrence relation is given by

$$\left(\frac{1}{\omega}D + L\right)x_k = \left(\frac{1}{\omega}D + L - A\right)x_{k-1} + b = \left(\left(-1 + \frac{1}{\omega}\right)D - U\right)x_{k-1} + b.$$

**Proposition 4.** *For $0 < \omega < 2$ the SOR method converges if $A$ is symmetric, positive definite and has positive diagonal entries.*

The parameter $\omega$ is used to speed up the convergence.
The choice $\omega = 1$ is the *Gauss-Seidel* method.

Algorithm

**Inputs:** matrices $A, Q$, vector $b$, precision $\varepsilon$, maximum number of iterations $K$.

1. choose $\widehat{x}_0$ at random
2. for $k$ from $1$ to $K$ do
   2.1 $\widehat{x}_{k+1} = Q^{-1}(Q - A)\widehat{x}_k + Q^{-1}b$
   2.2 if $\|A\widehat{x}_k - b\| < \varepsilon$, **return** $\widehat{x}_k$ (or in general if any convergence criterion is satisfied)
3. **return** "no solution found after $K$ steps".

---

Demonstration - Jacobi method (1/2)

Let $A = \begin{pmatrix} 2 & 1 \\ 1 & 4 \end{pmatrix}$. $\left\| I - D^{-1}A \right\| = \frac{1}{2}$.

We use the Jacobi method to calculate a solution for $b = (3, 5)^T$.
The exact solution is $(1, 1)^T$.

The convergence criterion used is $\|A\widehat{x}_k - b\| < 10^{-2}$.

| $k$ | $\widehat{x}_k$ | $\|A\widehat{x}_k - b\|$ |
|---|---|---|
| 0 | $(0.5, 1.5)$ | 1.58113883008 |
| 1 | $(0.75, 1.125)$ | 0.450693909433 |
| 2 | $(0.9375, 1.0625)$ | 0.197642353761 |
| 3 | $(0.96875, 1.015625)$ | 0.0563367386791 |
| 4 | $(0.9921875, 1.0078125)$ | 0.0247052942201 |
| 5 | $(0.99609375, 1.001953125)$ | 0.00704209233489 |

---

Demonstration - Jacobi method (2/2)

...the same problem but with a different $\widehat{x}_0$, which is further from the exact solution.

| $k$ | $\widehat{x}_k$ | $\|A\widehat{x}_k - b\|$ |
|---|---|---|
| 0 | $(-10, 10)$ | 28.1780056072 |
| 1 | $(-3.5, 3.75)$ | 9.01734439844 |
| 2 | $(-0.375, 2.125)$ | 3.5222507009 |
| 3 | $(0.4375, 1.34375)$ | 1.1271680498 |
| 4 | $(0.828125, 1.140625)$ | 0.440281337613 |
| 5 | $(0.9296875, 1.04296875)$ | 0.140896006226 |
| 6 | $(0.978515625, 1.017578125)$ | 0.0550351672016 |
| 7 | $(0.9912109375, 1.00537109375)$ | 0.0176120007782 |
| 8 | $(0.997314453125, 1.002197265625)$ | 0.0068793959002 |