Introduction to Numerical Analysis

Laura Grigori

EPFL and PSI

September 10, 2025

Plan

Representation of real numbers on computers

- The computer only allows the representation of a finite set of real numbers ($\mathbb{R}$) and of numbers with a finite number of digits

$$\text{e.g. } \frac{1}{3} = 0.\bar{3} \rightarrow 0.33333 \tag{1}$$

- A real number $x \in \mathbb{R}$ is truncated by the calculator and it is replaced by the floating point number $\text{fl}(x) \in \mathbb{F}$

Definition 1.1

The set of floating point numbers $\mathbb{F}$ is the subset of real numbers which can be represented on a computer, i.e., $\mathbb{F} \subset \mathbb{R}$, with $\dim(\mathbb{F}) < +\infty$. In general, $\mathbb{F} = \mathbb{F}_0 \cup \{0\}$, with $\mathbb{F}_0$ being the floating point numbers excluding zero.

Floating point numbers

A floating point number $x \in \mathbb{F}_0(\beta, t, L, U)$ is represented as

$$x = (-1)^s \cdot m \cdot \beta^{e-t} = (-1)^s \cdot (a_1 a_2 \dots a_t)_\beta \cdot \beta^{e-t},$$

where

- β is the base (the numerical system);
- $m = (a_1 a_2 \dots a_t)_\beta$ is the mantissa ($0 < m < \beta^t - 1$) with t being the number of significant digits (the digits a_i are such that $0 \leq a_i \leq \beta - 1$ and $a_1 \neq 0$);
- $e \in \mathbb{Z}$ is the exponent such that $e \in [L, U]$, with $L < 0$ and $U > 0$;
- $s = 0, 1$ is the sign.

Machine representation of real numbers

For the set $\mathbb{F}_0(\beta, t, L, U)$:

- the smallest and largest positive numbers are $x_{\min} = \beta^{L-1}$ and $x_{\max} = \beta^U (1 - \beta^{-t})$;
- the machine epsilon is the minimum real number greater than zero such that $\mathrm{fl}(1 + \epsilon_M) > 1$, is $\epsilon_M = \beta^{1-t}$;
- the relative roundoff error for $x \in \mathbb{R}$ when it is represented by $\mathrm{fl}(x) \in \mathbb{F}_0$ is:

$$\frac{|x - \mathrm{fl}(x)|}{|x|} \leq \frac{1}{2}\epsilon_M, \quad x \neq 0,$$

with $\frac{1}{2}\epsilon_M$ being an upper bound.

Example for $\mathbb{F}_0(2, 2, -1, 2)$

Consider $\beta = 2$ (numerical system in base 2), $t = 2$ (number of digits), $L = -1$, and $U = 2$, then:

$$\epsilon_M = \beta^{1-t} = \frac{1}{2}, \quad x_{\min} = \beta^{L-1} = \frac{1}{4}, \quad \text{and} \quad x_{\max} = \beta^U \left(1 - \beta^{-t}\right) = 3.$$

- ϵ_M is the smallest positive number such that:

$$\text{fl}(1 + \epsilon_M) > 1$$

- Values of the exponent e: $-1, 0, 1$, and 2
- Mantissa $m = (a_1 a_2)_2$, hence $a_1 = 1$, a_2 is 0 or 1
- For $s = 0$, the positive real numbers in $\mathbb{F}_0$ are $x = m\beta^{e-t} = m2^{e-2}$:

e	-1	0	1	2
$m = (10)_2 = 2$	$\frac{1}{4}$	$\frac{1}{2}$	1	2
$m = (11)_2 = 3$	$\frac{3}{8}$	$\frac{3}{4}$	$\frac{3}{2}$	3

- Larger $|fl(x)|$, lesser dense are the numbers in $\mathbb{R}$

Example for $\mathbb{F}_0(2, 2, -1, 2)$

Consider $\beta = 2$ (numerical system in base 2), $t = 2$ (number of digits), $L = -1$, and $U = 2$, then:

$$\epsilon_M = \beta^{1-t} = \frac{1}{2}, \quad x_{\min} = \beta^{L-1} = \frac{1}{4}, \quad \text{and} \quad x_{\max} = \beta^U \left(1 - \beta^{-t}\right) = 3.$$

- ϵ_M is the smallest positive number such that:

$$\text{fl}(1 + \epsilon_M) > 1$$

- Values of the exponent e: $-1, 0, 1$, and 2
- Mantissa $m = (a_1 a_2)_2$, hence $a_1 = 1$, a_2 is 0 or 1
- For $s = 0$, the positive real numbers in $\mathbb{F}_0$ are $x = m\beta^{e-t} = m2^{e-2}$:

e	-1	0	1	2
$m = (10)_2 = 2$	$\frac{1}{4}$	$\frac{1}{2}$	1	2
$m = (11)_2 = 3$	$\frac{3}{8}$	$\frac{3}{4}$	$\frac{3}{2}$	3

- Larger $|fl(x)|$, lesser dense are the numbers in $\mathbb{R}$

Example for $\mathbb{F}_0(2, 2, -1, 2)$

Consider $\beta = 2$ (numerical system in base 2), $t = 2$ (number of digits), $L = -1$, and $U = 2$, then:

$$\epsilon_M = \beta^{1-t} = \frac{1}{2}, \quad x_{\min} = \beta^{L-1} = \frac{1}{4}, \quad \text{and} \quad x_{\max} = \beta^U \left(1 - \beta^{-t}\right) = 3.$$

- ϵ_M is the smallest positive number such that:

$$\text{fl}(1 + \epsilon_M) > 1$$

- Values of the exponent e: $-1, 0, 1$, and 2
- Mantissa $m = (a_1 a_2)_2$, hence $a_1 = 1$, a_2 is 0 or 1
- For $s = 0$, the positive real numbers in $\mathbb{F}_0$ are $x = m\beta^{e-t} = m2^{e-2}$:

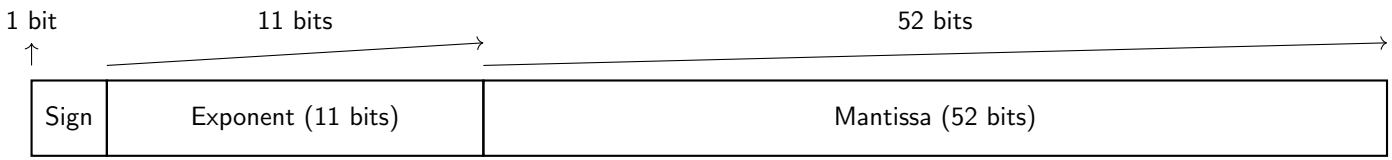
e	-1	0	1	2
$m = (10)_2 = 2$	$\frac{1}{4}$	$\frac{1}{2}$	1	2
$m = (11)_2 = 3$	$\frac{3}{8}$	$\frac{3}{4}$	$\frac{3}{2}$	3

- Larger $|fl(x)|$, lesser dense are the numbers in $\mathbb{R}$

Single/double precision

When the basis $\beta = 2$ is used, we reserve:

- in single precision ($N = 32$ bits), 1 digit for s, 23 digits for m, and 8 digits for e;
- in double precision ($N = 64$ bits), 1 digit for s, 52 digits for m, and 11 digits for e.



Matlab double precision: since first digit a_1 is always equal to 1, number of digits t used for the mantissa m is $52 + 1 = 53$, and:

$$\epsilon_M = 2^{1-53} \approx 2 \times 10^{-16}$$

$$x_{\min} \approx 10^{-308}, \quad x_{\max} \approx 10^{308}$$

Round-off errors

- Round-off errors accumulate during computations
- Potentially can lead to numerically unstable algorithms

Example: For any $x \in \mathbb{R} \setminus \{0\}$, we have:

$$\frac{(1+x)-1}{x} \equiv 1.$$

In floating point arithmetic:

$$\frac{\mathrm{fl}(1+\mathrm{fl}(x))-1}{\mathrm{fl}(x)} = y,$$

where y is a real number generally different from 1.

x	Relative Error (%)
10^{-10}	$8 \cdot 10^{-6}$
10^{-14}	$8 \cdot 10^{-2}$
10^{-15}	11
10^{-16}	100

Table: Relative error for different values of x

Round-off errors

- Round-off errors accumulate during computations
- Potentially can lead to numerically unstable algorithms

Example: For any $x \in \mathbb{R} \setminus \{0\}$, we have:

$$\frac{(1+x)-1}{x} \equiv 1.$$

In floating point arithmetic:

$$\frac{\mathrm{fl}(1+\mathrm{fl}(x))-1}{\mathrm{fl}(x)} = y,$$

where y is a real number generally different from 1.

x	Relative Error (%)
10^{-10}	$8 \cdot 10^{-6}$
10^{-14}	$8 \cdot 10^{-2}$
10^{-15}	11
10^{-16}	100

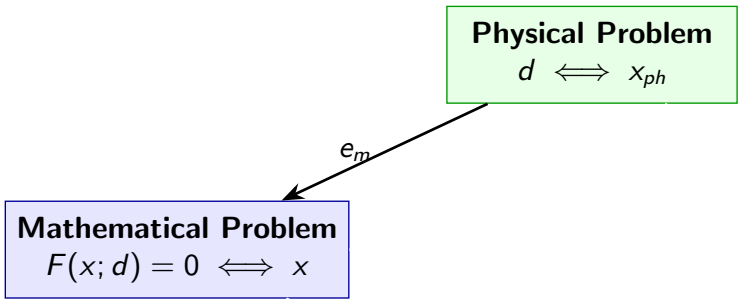
Table: Relative error for different values of x

The computational process

From the Physical Problem to the Mathematical Problem

- Consider a physical problem (PP) with a physical solution x_{ph}, which depends on some data d
- The mathematical problem (MP) represents the mathematical formulation of the PP with the mathematical solution x:

$$F(x, d) = 0, \quad x \in \mathcal{X}, d \in \mathcal{D}, \quad e_m := x_{ph} - x$$



- d = data,
- x_{ph} = solution of the physical problem,
- x = solution of the mathematical problem,
- e_m = modeling error.

Example

- Physical problem: a body falling under the action of external forces
- Physical solution: x_{ph} the velocity of the body at time $t_f > 0$
- Consider the model:

$$\text{Find } V(t): \begin{cases} m\dot{V}(t) = f_{\text{ext}}(t) & \text{for } t > 0, \\ V(0) = 0, \end{cases}$$

where $V(t)$ is body velocity, m its mass, $f_{\text{ext}}(t)$ external forces.

Mathematical solution: $x = V(t_f)$
Mathematical problem (MP):

$$F(x; d) = x - \int_0^{t_f} \frac{f_{\text{ext}}(t)}{m}\, dt = 0,$$

where the data are $d = (t_f, m, f_{\text{ext}}(t))$.

The Mathematical Problem

Definition 1.3

Let us consider an admissible perturbation on the data δd (i.e. such that $d + \delta d \in \mathcal{D}$) inducing the perturbation δx on the solution $x \in \mathcal{X}$ (i.e. for which $F(x + \delta x; d + \delta d) = 0$). The solution $x \in \mathcal{X}$ is continuously dependent on the data if:

$$\exists \eta_0(d) > 0, \exists K_0(d) > 0 : \textit{if } \|\delta d\| \leq \eta_0(d) \Rightarrow \|\delta x\| \leq K_0(d)\|\delta d\|.$$

Definition 1.4

The mathematical problem $F(x; d) = 0$ is well-posed (stable) if and only if there exists a unique solution $x \in \mathcal{X}$ continuously dependent on the data $d \in \mathcal{D}$.

The Mathematical Problem

Definition 1.5

The (relative) conditioning (number) of a problem $F(x; d) = 0$ for the data $d \in \mathcal{D}$ is:

$$K(d) := \sup \left(\frac{\|\delta x\|/\|x\|}{\|\delta d\|/\|d\|}, \quad \forall \delta d : d + \delta d \in \mathcal{D}, \delta d \neq 0 \right).$$

- $K(d)$ measures the sensitivity of well-posed MP, i.e. even small changes of d can lead to large variations of x
- If $K(d)$ is "small," the problem $F(x; d) = 0$ is well-conditioned. Conversely, if $K(d)$ is "large," the problem is ill-conditioned

The Mathematical Problem

Definition 1.5

The (relative) conditioning (number) of a problem $F(x; d) = 0$ for the data $d \in \mathcal{D}$ is:

$$K(d) := \sup \left(\frac{\|\delta x\|/\|x\|}{\|\delta d\|/\|d\|}, \quad \forall \delta d : d + \delta d \in \mathcal{D}, \delta d \neq 0 \right).$$

- $K(d)$ measures the sensitivity of well-posed MP, i.e. even small changes of d can lead to large variations of x
- If $K(d)$ is "small," the problem $F(x; d) = 0$ is well-conditioned. Conversely, if $K(d)$ is "large," the problem is ill-conditioned

Example of ill-conditioned problem

Consider the MP:

$$F(x; d) = dx - \alpha = 0, \text{ for some } \alpha \in \mathbb{R}, x \in \mathcal{X} \equiv \mathbb{R}, d \in \mathcal{D} \equiv \mathbb{R}$$

The perturbed problem:

$$F(x + \delta x; d + \delta d) = (d + \delta d)(x + \delta x) - \alpha = 0$$

$$\Longrightarrow \frac{\delta x}{x} = -\frac{d}{d + \delta d}\frac{\delta d}{d}.$$

Conditioning:

$$K(d) \simeq \sup_{\delta d:(d+\delta d)\in D,\, \|\delta d\|\neq 0} \left|\frac{d}{d + \delta d}\right|$$

large if $\delta d \simeq -d$.

Example of ill-conditioned problem

Consider the MP:

$$F(x; d) = dx - \alpha = 0, \text{ for some } \alpha \in \mathbb{R}, x \in \mathcal{X} \equiv \mathbb{R}, d \in \mathcal{D} \equiv \mathbb{R}$$

The perturbed problem:

$$F(x + \delta x; d + \delta d) = (d + \delta d)(x + \delta x) - \alpha = 0$$

$$\implies \frac{\delta x}{x} = -\frac{d}{d + \delta d}\frac{\delta d}{d}.$$

Conditioning:

$$K(d) \simeq \sup_{\delta d:(d+\delta d)\in D,\, \|\delta d\|\neq 0} \left|\frac{d}{d + \delta d}\right|$$

large if $\delta d \simeq -d$.

Example of ill-conditioned problem

Consider the MP:

$$F(x; d) = dx - \alpha = 0, \text{ for some } \alpha \in \mathbb{R}, x \in \mathcal{X} \equiv \mathbb{R}, d \in \mathcal{D} \equiv \mathbb{R}$$

The perturbed problem:

$$F(x + \delta x; d + \delta d) = (d + \delta d)(x + \delta x) - \alpha = 0$$

$$\Longrightarrow \quad \frac{\delta x}{x} = -\frac{d}{d + \delta d}\frac{\delta d}{d}.$$

Conditioning:

$$K(d) \simeq \sup_{\delta d:(d+\delta d)\in D,\, \|\delta d\|\neq 0} \left| \frac{d}{d + \delta d} \right|$$

large if $\delta d \simeq -d$.

The computational process

From the Mathematical Problem to the Numerical Problem

- The numerical problem (NP) is an approximation of the MP
- We refer to the NP as:

$$F_h(x_h, d_h) = 0, \quad x_h \in \mathcal{X}_h, d_h \in \mathcal{D}_h, \mathcal{X}_h, \mathcal{D}_h \text{ suitable spaces}$$

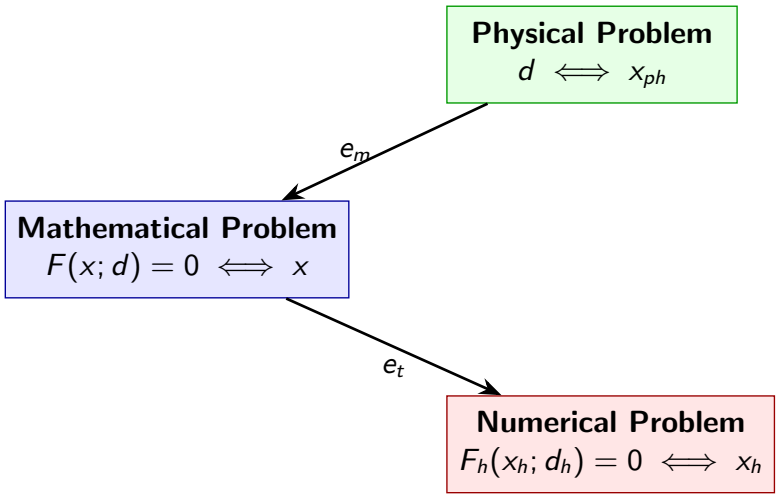
- d = data,
- x_{ph} = solution of the physical problem,
- x = solution of mathematical problem,
- x_h = solution of numerical problem,
- e_m = modeling error,
- e_t = truncation error.

The computational process

From the Mathematical Problem to the Numerical Problem

- The numerical problem (NP) is an approximation of the MP
- We refer to the NP as:

$$F_h(x_h, d_h) = 0, \quad x_h \in \mathcal{X}_h, d_h \in \mathcal{D}_h, \mathcal{X}_h, \mathcal{D}_h \text{ suitable spaces}$$



- d = data,
- x_{ph} = solution of the physical problem,
- x = solution of mathematical problem,
- x_h = solution of numerical problem,
- e_m = modeling error,
- e_t = truncation error.

The computational process

From the Mathematical Problem to the Numerical Problem

- The final solution $\hat{x}_h$ is affected by round-off error, $e_r := x_h - \hat{x}_h$
- The computational error $e_c := x - \hat{x}_h = e_t + e_r$

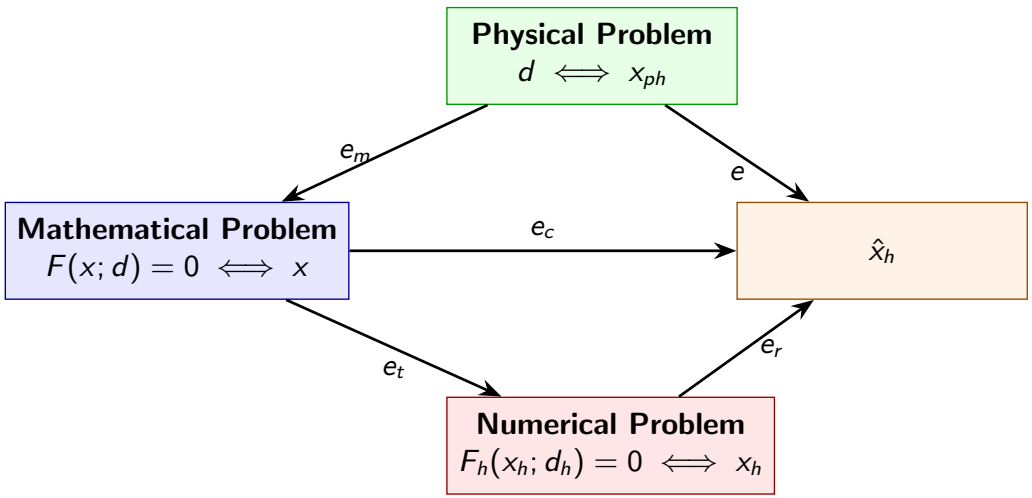
- d = data,
- x_{ph} = solution of phys. pb.,
- x = solution of the math. pb.,
- x_h = solution of numer. pb.,
- $\hat{x}_h$ = final solution,
- e_m = modeling error,
- $e_t := x - x_h$ truncation error,
- e_r = roundoff error,
- e_c = computational error,
- e = total error,

(often $|e_r| \ll |e_t|$ and so $x_h \approx \hat{x}_h$)

The computational process

From the Mathematical Problem to the Numerical Problem

- The final solution $\hat{x}_h$ is affected by round-off error, $e_r := x_h - \hat{x}_h$
- The computational error $e_c := x - \hat{x}_h = e_t + e_r$



- d = data,
- x_{ph} = solution of phys. pb.,
- x = solution of the math. pb.,
- x_h = solution of numer. pb.,
- $\hat{x}_h$ = final solution,
- e_m = modeling error,
- $e_t := x - x_h$ truncation error,
- e_r = roundoff error,
- e_c = computational error,
- e = total error,

(often $|e_r| \ll |e_t|$ and so $x_h \approx \hat{x}_h$)

Example

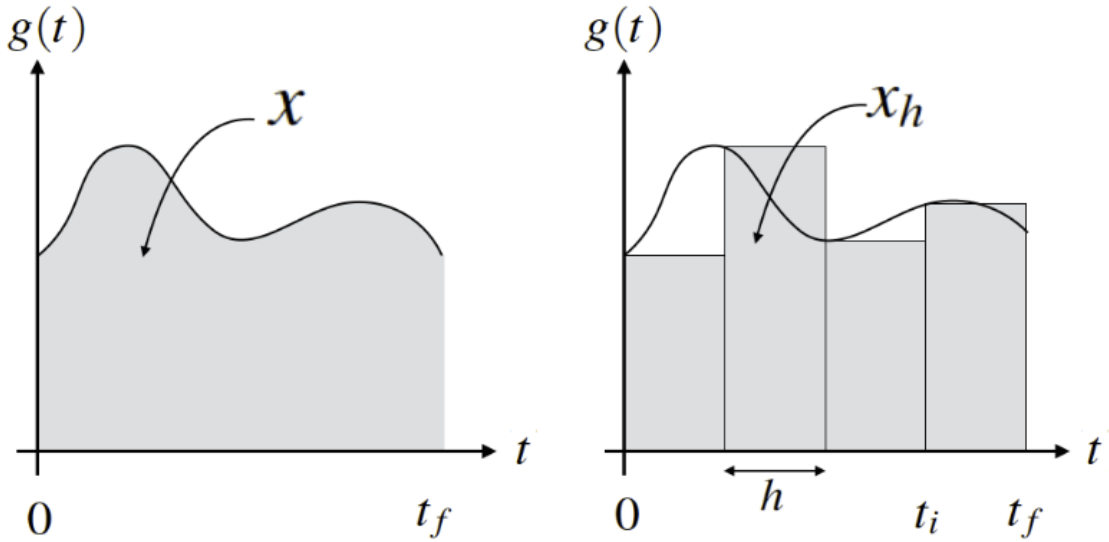
Given the MP:

$$F(x;d) = x - \int_0^{t_f} g(t)\,dt = 0 \quad \text{with the data} \quad d = \{t_f, g(t)\},$$

we can have the NP:

$$F_h(x_h;d_h) = x_h - h\sum_{i=0}^{n-1} g(t_i) = 0,$$

where $t_i = ih$ for $i = 0,\ldots,n$, with $h = \frac{t_f}{n}$.



The numerical problem

$h, n =$ discretization parameters ($h \to 0$ and $n \to \infty$).

Definition 1.6

The numerical problem $F_h(x_h; d_h) = 0$ is well-posed (stable) if and only if there exists a unique solution $x_h \in \mathcal{X}_h$ continuously dependent on the data $d_h \in \mathcal{D}_h$.

Definition

Let us consider an admissible perturbation on the data δd_h (i.e., such that $d_h + \delta d_h \in D_h$) inducing the perturbation δx_h on the solution $x_h \in \mathcal{X}_h$ (i.e., for which $F_h(x_h + \delta x_h; d_h + \delta d_h) = 0$). The solution $x_h \in \mathcal{X}_h$ is continuously dependent on the data if:

$$\exists \delta_{0,h}(d_h) > 0, \exists K_{0,h}(d_h) > 0 \text{ such that:}$$

$$\|\delta d_h\| \leq \delta_{0,h}(d_h) \implies \|\delta x_h\| \leq K_{0,h}(d_h)\|\delta d_h\|.$$

The Numerical Problem (Method)

Definition 1.7

The (relative) conditioning (number) of the numerical problem $F_h(x_h; d_h) = 0$ *for the data* $d_h \in \mathcal{D}_h$ *is:*

$$K_h(d_h) := \sup \left(\frac{\|\delta x_h\| / \|x_h\|}{\|\delta d_h\| / \|d_h\|} \ \forall \delta d_h : d_h + \delta d_h \in \mathcal{D}_h, \ \|\delta d_h\| \neq 0 \right).$$

Remark: If $K_h(d_h)$ is "small," the numerical problem $F_h(x_h; d_h) = 0$ is well-conditioned. Conversely, if $K_h(d_h)$ is "large," the problem is ill-conditioned.

The Numerical Problem: consistency

Definition 1.8

The numerical method (problem) $F_h(x_h; d_h) = 0$ *is consistent iff:*

$$F_h(x; d) - F(x; d) \to 0 \text{ as } h \to 0,$$

when the data $d \in \mathcal{D}$ *is admissible for* $F_h(\cdot; \cdot)$ *(i.e.,* $d \in \mathcal{D}_h$*).*

Definition 1.9

The numerical method (problem) $F_h(x_h; d_h) = 0$ *is strongly consistent iff:*

$$F_h(x; d) \equiv F(x; d) = 0 \text{ for all } h > 0,$$

when the data $d \in \mathcal{D}$ *is admissible for* $F_h(\cdot; \cdot)$ *(i.e.,* $d \in \mathcal{D}_h$*).*

Example of consistency

Given

$$\text{MP: } F(x;d) = x - d = 0, \text{ with } d = \sqrt{2}, \text{ for which } x = \sqrt{2}$$

Consider two different NP associated with the MP:

1. NP: $F_n(x_n;d) = x_{n+1} - \frac{3}{4}x_n - \frac{1}{2x_n} = 0$ for $n \geq 0$, with $x_0 = 1$;
   - n is the discretization parameter/iteration number
   - strongly consistent: $F_n(x;d) = \sqrt{2} - \frac{3}{4}\sqrt{2} - \frac{1}{2\sqrt{2}} = 0$ for all $n \geq 0$

2. NP: $F_n(x_n;d) = x_{n+1} - \frac{3}{4}x_n - \frac{1}{2x_n} + \frac{1}{(1+n)^5} = 0$ for $n \geq 0$, with $x_0 = 1$
   - NP not strongly consistent: $F_n(x;d) = \frac{1}{(1+n)^5} \neq 0$ for $n \geq 0$
   - NP consistent: $\lim_{n\to+\infty} F_n(x;d) = 0$.

Example of consistency

Given

$$\text{MP: } F(x;d) = x - d = 0, \text{ with } d = \sqrt{2}, \text{ for which } x = \sqrt{2}$$

Consider two different NP associated with the MP:

1. NP: $F_n(x_n;d) = x_{n+1} - \frac{3}{4}x_n - \frac{1}{2x_n} = 0$ for $n \geq 0$, with $x_0 = 1$;
   - n is the discretization parameter/iteration number
   - strongly consistent: $F_n(x;d) = \sqrt{2} - \frac{3}{4}\sqrt{2} - \frac{1}{2\sqrt{2}} = 0$ for all $n \geq 0$

2. NP: $F_n(x_n;d) = x_{n+1} - \frac{3}{4}x_n - \frac{1}{2x_n} + \frac{1}{(1+n)^5} = 0$ for $n \geq 0$, with $x_0 = 1$
   - NP not strongly consistent: $F_n(x;d) = \frac{1}{(1+n)^5} \neq 0$ for $n \geq 0$
   - NP consistent: $\lim_{n\to+\infty} F_n(x;d) = 0$.

Example of consistency

Given

$$\text{MP: } F(x;d) = x - d = 0, \text{ with } d = \sqrt{2}, \text{ for which } x = \sqrt{2}$$

Consider two different NP associated with the MP:

1. NP: $F_n(x_n;d) = x_{n+1} - \frac{3}{4}x_n - \frac{1}{2x_n} = 0$ for $n \geq 0$, with $x_0 = 1$;
   - n is the discretization parameter/iteration number
   - strongly consistent: $F_n(x;d) = \sqrt{2} - \frac{3}{4}\sqrt{2} - \frac{1}{2\sqrt{2}} = 0$ for all $n \geq 0$

2. NP: $F_n(x_n;d) = x_{n+1} - \frac{3}{4}x_n - \frac{1}{2x_n} + \frac{1}{(1+n)^5} = 0$ for $n \geq 0$, with $x_0 = 1$
   NP not strongly consistent: $F_n(x;d) = \frac{1}{(1+n)^5} \neq 0$ for $n \geq 0$
   NP consistent: $\lim_{n\to+\infty} F_n(x;d) = 0$.

Example of consistency

Given

$$\text{MP: } F(x;d) = x - d = 0, \text{ with } d = \sqrt{2}, \text{ for which } x = \sqrt{2}$$

Consider two different NP associated with the MP:

1. NP: $F_n(x_n;d) = x_{n+1} - \frac{3}{4}x_n - \frac{1}{2x_n} = 0$ for $n \geq 0$, with $x_0 = 1$;
   - n is the discretization parameter/iteration number
   - strongly consistent: $F_n(x;d) = \sqrt{2} - \frac{3}{4}\sqrt{2} - \frac{1}{2\sqrt{2}} = 0$ for all $n \geq 0$

2. NP: $F_n(x_n;d) = x_{n+1} - \frac{3}{4}x_n - \frac{1}{2x_n} + \frac{1}{(1+n)^5} = 0$ for $n \geq 0$, with $x_0 = 1$
   NP not strongly consistent: $F_n(x;d) = \frac{1}{(1+n)^5} \neq 0$ for $n \geq 0$
   NP consistent: $\lim_{n\to+\infty} F_n(x;d) = 0$.

Example of consistency

Given

$$\text{MP: } F(x;d) = x - d = 0, \text{ with } d = \sqrt{2}, \text{ for which } x = \sqrt{2}$$

Consider two different NP associated with the MP:

1. NP: $F_n(x_n;d) = x_{n+1} - \frac{3}{4}x_n - \frac{1}{2x_n} = 0$ for $n \geq 0$, with $x_0 = 1$;
   - n is the discretization parameter/iteration number
   - strongly consistent: $F_n(x;d) = \sqrt{2} - \frac{3}{4}\sqrt{2} - \frac{1}{2\sqrt{2}} = 0$ for all $n \geq 0$

2. NP: $F_n(x_n;d) = x_{n+1} - \frac{3}{4}x_n - \frac{1}{2x_n} + \frac{1}{(1+n)^5} = 0$ for $n \geq 0$, with $x_0 = 1$
   NP not strongly consistent: $F_n(x;d) = \frac{1}{(1+n)^5} \neq 0$ for $n \geq 0$
   NP consistent: $\lim_{n\to+\infty} F_n(x;d) = 0$.

Example of consistency

Given

$$\text{MP: } F(x;d) = x - d = 0, \text{ with } d = \sqrt{2}, \text{ for which } x = \sqrt{2}$$

Consider two different NP associated with the MP:

1. NP: $F_n(x_n;d) = x_{n+1} - \frac{3}{4}x_n - \frac{1}{2x_n} = 0$ for $n \geq 0$, with $x_0 = 1$;
   - n is the discretization parameter/iteration number
   - strongly consistent: $F_n(x;d) = \sqrt{2} - \frac{3}{4}\sqrt{2} - \frac{1}{2\sqrt{2}} = 0$ for all $n \geq 0$

2. NP: $F_n(x_n;d) = x_{n+1} - \frac{3}{4}x_n - \frac{1}{2x_n} + \frac{1}{(1+n)^5} = 0$ for $n \geq 0$, with $x_0 = 1$
   NP not strongly consistent: $F_n(x;d) = \frac{1}{(1+n)^5} \neq 0$ for $n \geq 0$
   NP consistent: $\lim_{n\to+\infty} F_n(x;d) = 0$.

The Numerical Problem: convergence

Definition 1.10

Let

- $x(d) \in \mathcal{X}$ *be the solution of the MP* $F(x; d) = 0$ *for* $d \in \mathcal{D}$,
- $x_h(d + \delta d_h) \in \mathcal{X}_h$ *be the solution of the NP* $F_h(x_h; d + \delta d_h) = 0$, *with* $d + \delta d_h \in \mathcal{D}_h$.

The numerical method $F_h(x_h; d + \delta d_h) = 0$ *is convergent if and only if:*

$$\forall \epsilon > 0, \exists h_0 = h_0(\epsilon) > 0, \exists \Delta = \Delta(h_0, \epsilon) : \forall h < h_0(\epsilon), \forall \delta d_h : \|\delta d_h\| \leq \Delta$$

$$\implies \| x(d) - x_h(d + \delta d_h)\| \leq \epsilon.$$

In practice, the numerical method is convergent if the error $e_c = e_c(x_h) := |x - x_h|$ tends to zero when improving the discretization, i.e.:

$$e_c \to 0 \text{ as } h \to 0, (\text{or } n \to \infty).$$

The Numerical Problem: convergence

Definition 1.10

Let

- $x(d) \in \mathcal{X}$ *be the solution of the MP* $F(x; d) = 0$ *for* $d \in \mathcal{D}$,
- $x_h(d + \delta d_h) \in \mathcal{X}_h$ *be the solution of the NP* $F_h(x_h; d + \delta d_h) = 0$, *with* $d + \delta d_h \in \mathcal{D}_h$.

The numerical method $F_h(x_h; d + \delta d_h) = 0$ *is convergent if and only if:*

$$\forall \epsilon > 0, \exists h_0 = h_0(\epsilon) > 0, \exists \Delta = \Delta(h_0, \epsilon) : \forall h < h_0(\epsilon), \forall \delta d_h : \|\delta d_h\| \leq \Delta$$

$$\implies \| x(d) - x_h(d + \delta d_h)\| \leq \epsilon.$$

In practice, the numerical method is convergent if the error $e_c = e_c(x_h) := |x - x_h|$ tends to zero when improving the discretization, i.e.:

$$e_c \to 0 \text{ as } h \to 0, (\text{or } n \to \infty).$$

The Numerical Problem: convergence

Definition 1.10

Let

- $x(d) \in \mathcal{X}$ *be the solution of the MP* $F(x; d) = 0$ *for* $d \in \mathcal{D}$,
- $x_h(d + \delta d_h) \in \mathcal{X}_h$ *be the solution of the NP* $F_h(x_h; d + \delta d_h) = 0$, *with* $d + \delta d_h \in \mathcal{D}_h$.

The numerical method $F_h(x_h; d + \delta d_h) = 0$ *is convergent if and only if:*

$$\forall \epsilon > 0, \exists h_0 = h_0(\epsilon) > 0, \exists \Delta = \Delta(h_0, \epsilon) : \forall h < h_0(\epsilon), \forall \delta d_h : \|\delta d_h\| \leq \Delta$$

$$\implies \| x(d) - x_h(d + \delta d_h)\| \leq \epsilon.$$

In practice, the numerical method is convergent if the error $e_c = e_c(x_h) := |x - x_h|$ tends to zero when improving the discretization, i.e.:

$$e_c \to 0 \text{ as } h \to 0, (\text{or } n \to \infty).$$

The Numerical Problem (Method)

Definition 1.11

If the error e_c can be bounded as a function of h as:

$$e_c = e_c(x_h) \leq Ch^p,$$

for some $p > 0$ and C independent of h and p, the numerical method is convergent of order p.
If there exists $\tilde{C} > 0$ such that $\tilde{C}h^p \leq e_c \leq Ch^p$, then we can write:

$$e_c \approx Ch^p.$$

Estimating the convergence order

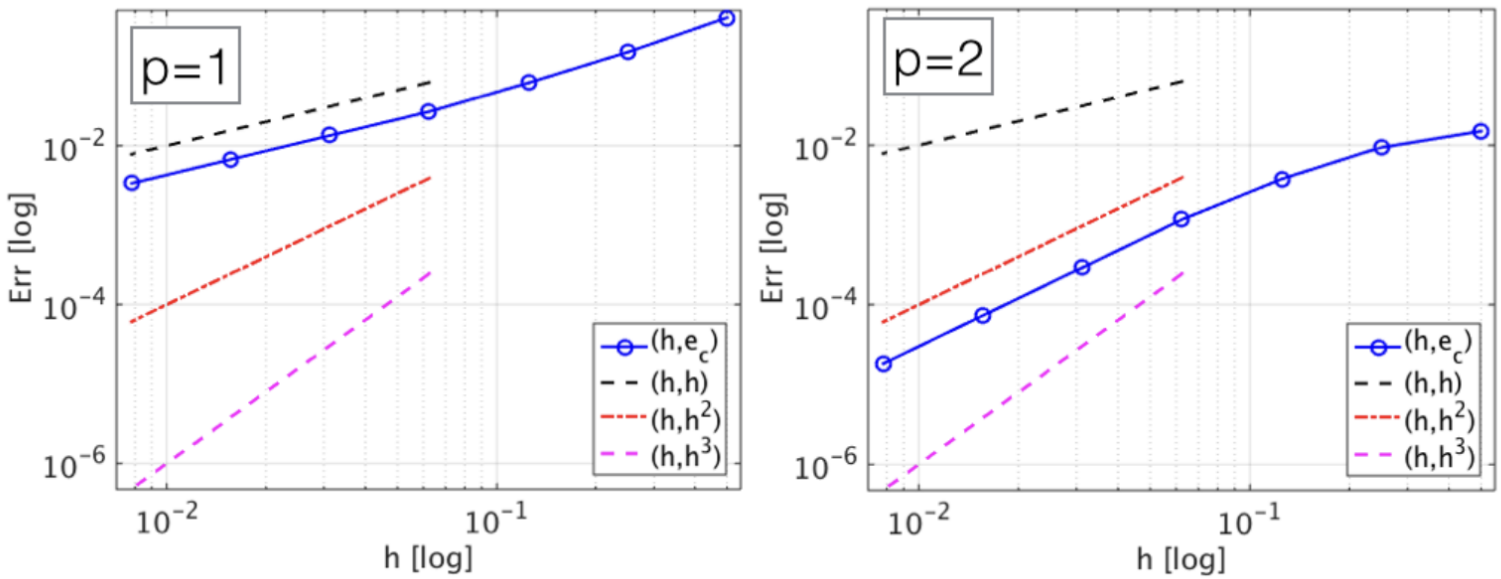
Assuming $e_c \approx Ch^p$, convergence order p can be estimated as:

1. *Algebraically:* given the exact solution of the mathematical problem x and two approximated solutions x_1 and x_2 corresponding to h_1 and h_2:

$$p = \frac{\log\left(e_c(x_1)/e_c(x_2)\right)}{\log\left(h_1/h_2\right)}$$

2. *Graphically*
   - Plot the errors e_c computed for different values of h vs. h in log–log scales
   - Verify if the curves (h, e_c) and (h, h^p) are parallel in log-log scales

$$\log e_c = \log(Ch^p) = \log C + p \log h \implies p = atan(\theta), \theta = \text{slope of } (h, e_c) \text{ curve}$$



Estimating the convergence order

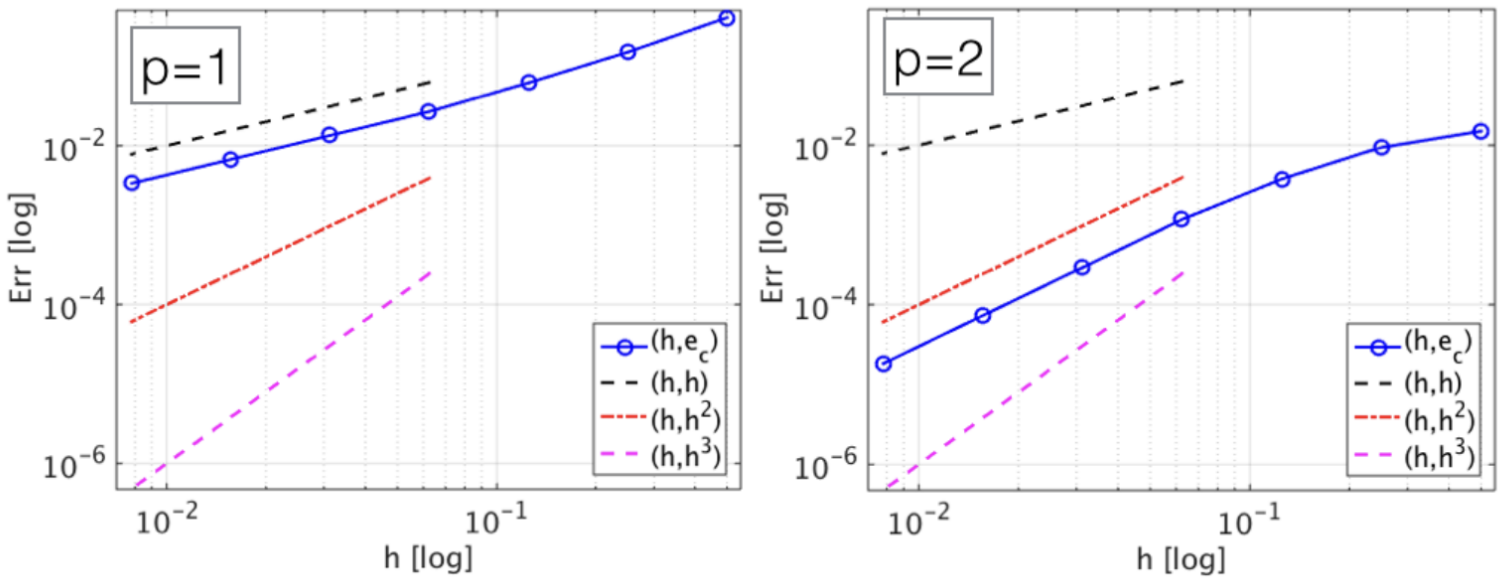
Assuming $e_c \approx Ch^p$, convergence order p can be estimated as:

1. *Algebraically:* given the exact solution of the mathematical problem x and two approximated solutions x_1 and x_2 corresponding to h_1 and h_2:

$$p = \frac{\log\left(e_c(x_1)/e_c(x_2)\right)}{\log\left(h_1/h_2\right)}$$

2. *Graphically*
   - Plot the errors e_c computed for different values of h vs. h in log–log scales
   - Verify if the curves (h, e_c) and (h, h^p) are parallel in log-log scales

$$\log e_c = \log(Ch^p) = \log C + p \log h \implies p = atan(\theta), \theta = \text{slope of } (h, e_c) \text{ curve}$$



The Numerical Problem (Method)

The numerical method (problem) must be well-posed, consistent, and convergent.

Theorem 1.1 (Lax–Richtmyer Equivalence Theorem). If the numerical method (problem) $F_h(\hat{x}_h; d_h) = 0$ is consistent, then it is convergent if and only if it is well-posed (stable).

Implications:

- If the numerical method (problem) is consistent and well-posed, then it is also convergent;
- If the numerical method (problem) is consistent and convergent, then it is also well-posed.

Choice of the Numerical Method

- **Properties of the mathematical problem.**
- **Efficiency:**
  - **Accuracy.** Convergence properties of the method, convergence order.
  - **Computational costs:** Number (order of magnitude) of floating point operations required for the execution of the algorithm; the flops are the number of these operations per second. The complexity of an algorithm may depend on the dimension of the problem m as $O(1)$, $O(m)$, $O(m^2)$, $O(m^3)$, or $O(m!)$.
- **Computer memory:** Time required to access the computer memory (depending on the implementation) and storage capabilities.

Computational costs

Complexity	**Flops**
$O(1)$	independent
$O(m)$	linear
$O(m^\gamma)$	polynomial
$O(\gamma^m)$	exponential
$O(m!)$	factorial

Table: Complexity and Flop Comparison

Example of a computation

- Compute $\det(A)$, $A \in \mathbb{R}^{m \times m}$
- Using Cramer's rule requires $O(m!)$ flops
- Estimated times using a calculator with a 1GHz $= 10^9$ flops/s CPU:

m	5	10	15	20
$m!$	120	$\sim 10^6$	$\sim 10^{12}$	$\sim 10^{18}$
CPU time	$\sim 10^{-7}$ s	$\sim 10^{-3}$ s	~ 30 min	~ 77 years