COURSE 1

Outline of the course:

- Introductive notions. Finite and divided differences.

- Approximation of functions: interpolation of Lagrange, Hermite and Birkhoff type. Least squares approximation.

- Numerical integration. Newton-Cotes quadrature formulas. Repeated quadrature formulas. General quadrature formulas. Romberg's algorithm. Adaptive quadratures formulas. Gauss type quadrature formulas.

- Numerical methods for solving linear systems - direct methods (Gauss, Gauss-Jordan, LU-methods). Perturbations of a linear system.

- Numerical methods for solving linear systems - iterative methods (Jacobi, Gauss-Seidel, SOR).

- Methods for solving nonlinear equations in R: one-step methods (Newton (tangent) method) and multi-step methods (secant, bisection and false position methods). Lagrange, Hermite and Birkhoff inverse interpolation.

- Methods for solving nonlinear systems: successive approximation and Newton methods.

- Numerical methods for solving differential equations: Taylor interpolation, Euler and Runge-Kutta methods.

- Revision of the main types of problems.

Evaluation methods

- Written exam: 70%

- Lab activities (evaluation and continuous observations during the semester): 30%

- Each lab will be evaluated.

- Each lab should be delivered in the same week or the next one.

- Respect delivery dates for each lab assignment. Each delay will be penalized: 1 point/one week of delay.

- A lab that will not be delivered will have grade 1.

- You may deliver 2 lab assignments during one lab.

References:

1. I. Chiorean, T. Cătinaş, R. Trîmbitaş, *Analiză Numerică*, Ed. Presa Univ. Clujeană, 2010.

2. W. Gander, M. Gander, F. Kwok, *Scientific Computing, An Introduction using Maple and MATLAB*, Springer, 2014.

3. R. L. Burden, J. D. Faires, *Numerical Analysis*, PWS Publishing Company, 2010.

4. R. Trîmbitaş, *Numerical Analysis in Matlab*, Ed. Presa Univ. Clujeană, 2011

Chapter 1. Preliminary notions

We will study numerical methods and algorithms and analyze the error. In the most cases, finding an exact solution is not possible, so we approximate it, we find it numerically.

1.1. Preliminaries

Definition 1 *Let* $x^* \in \mathbb{R}$ *be an unknown value of interest. An element* $\tilde{x} \in \mathbb{R}$ *which approximates* x^* *is called* **the approximation** *or* **the approximant** *of* x^*.

The expression $\Delta x = x^* - \tilde{x}$ *or* $\Delta x = \tilde{x} - x^*$ *is called* **the error**.

The value $|\Delta x| = |x^* - \tilde{x}|$ *is called* **the absolute error** of approximation.

The value $\delta x = \frac{|\Delta x|}{|x^*|} = \frac{|x^* - \tilde{x}|}{|x^*|}$, $x^* \neq 0$ *is called* **the relative error** of approximation. *The relative error is a proportion, so we can also* *express it as a percentage by multiplying the relative error by* 100%*. (The value* $100\% \cdot \delta x$ *is called* **the percent error** of approximation.)

The relative error is used to put error into perspective and it gives an indication of how good a measurement is relative to the size of the thing being measured.

For example, an error of $1cm$ would be a lot if the total length is $15cm$, but insignificant if the length is $5km$.

Example: Consider two approximative measurements of the weight $5.00g$: $5.05g$ and 4.95g. The absolute error is $0.05g$. The relative error is $0.05g/5.00g = 0.01$ or 1%.

The notions are correspondingly extended to normed space.

Definition 2 *If V is a K-linear space then a real functional $p : V \to [0, \infty)$, with the properties:*

$$\begin{aligned} &1)\ p(v_1 + v_2) \le p(v_1) + p(v_2), \quad \forall v_1, v_2 \in V,\\ &2)\ p(\alpha v) = |\alpha|\, p(v), \quad \forall \alpha \in K, v \in V,\\ &3)\ p(v) = 0 \Longrightarrow v = 0 \end{aligned}$$

is called **a norm** *on V.*

Definition 3 *Let K be a field and V be a given set. We say that V is a K-***linear space** (a linear space over K) *if there exist an internal operation:*

$$" + " : V \times V \to V; \quad (v_1, v_2) \to v_1 + v_2,$$

and an external operation:

$$" \cdot " : K \times V \to V; \quad (\alpha, v) \to \alpha v$$

that satisfy the following conditions:

1) $(V, +)$ is a commutative group

2)

$$
\begin{aligned}
&a)\ (\alpha+\beta)v=\alpha v+\beta v, \quad \forall\alpha,\beta\in K, \quad \forall v\in V,\\
&b)\ \alpha(v_1+v_2)=\alpha v_1+\alpha v_2, \quad \forall\alpha\in K, \quad \forall v_1,v_2\in V,\\
&c)\ (\alpha\beta)v=\alpha(\beta v), \quad \forall\alpha,\beta\in K, \quad \forall v\in V,\\
&d)\ 1\cdot v=v, \quad \forall v\in V.
\end{aligned}
$$

The elements of V are called *vectors* and those of K are called *scalars*.

Definition 4 *Let V and V' be two K-linear spaces. A function $f:V\to V'$ is called* linear transformation *or* linear operator *if:*

$$
\begin{aligned}
&1)\ f(v_1+v_2)=f(v_1)+f(v_2), \quad \forall v_1,v_2\in V \quad \textit{(aditivity)}\\
&2)\ f(\alpha v)=\alpha f(v), \quad \forall\alpha\in K, \quad \forall v\in V \qquad\qquad \textit{(homogenity).}
\end{aligned}
$$

Or, shortly,

$$f(\alpha v_1+\beta v_2)=\alpha f(v_1)+\beta f(v_2), \quad \forall\alpha,\beta\in K, \quad \forall v_1,v_2\in V.$$

Definition 5 *Let V be a linear space on $\mathbb{R}$ or $\mathbb{C}$. A linear operator $P : V \to V$ is called* **projector** *if*

$$P \circ P = P, \quad \textit{(shortly, } P^2 = P\textit{).}$$

Remark 6 *1) The identity operator $I : V \to V,\ I(v) = v$ and the null operator $0 : V \to V,\ 0(v) = 0$ are projectors.*

2) P is projector $\Rightarrow P^C := I - P$, **the complement of** *P, is projector.*

1.2. Finite and divided differences

Finite differences

Let $M = \{a_i \mid a_i = a + ih,$ with $i = 0, ..., m;\ a, h \in \mathbb{R}^*,\ m \in \mathbb{N}^*\}$ and $\mathcal{F} = \{f \mid f : M \to \mathbb{R}\}$.

Definition 7 *For* $f \in \mathcal{F}$,

$$(\triangle_h f)(a_i) = f(a_{i+1}) - f(a_i), \qquad i < m$$

is called **the finite difference of the first order** *of the function* f, *with step* h, *at point* a_i.

Theorem 8 *The operator* $\triangle_h$ *is a linear operator with respect to* f.

Proof. If $f, g : M \to \mathbb{R};\ A, B \in \mathbb{R}$ and $i < m$, we have

$$\begin{aligned}(\triangle_h(Af + Bg))(a_i) &= (Af + Bg)(a_{i+1}) - (Af + Bg)(a_i) \qquad (1)\\ &= A[f(a_{i+1}) - f(a_i)] + B[g(a_{\ i+1}) - g(a_i)]\\ &= A(\triangle_h f)(a_i) + B(\triangle_h g)(a_i).\end{aligned}$$

■

Definition 9 *Let* $0 \leq i < m,\ k \in \mathbb{N}$ *and* $1 \leq k \leq m - i$

$$\begin{aligned}(\triangle_h^k f)(a_i) &= (\triangle_h(\triangle_h^{k-1} f))(a_i) \\ &= (\triangle_h^{k-1} f)(a_{i+1}) - (\triangle_h^{k-1} f)(a_i), \quad \textit{with } \triangle_h^0 = I \textit{ și } \triangle_h^1 = \triangle_h\end{aligned} \tag{2}$$

is called **the k-th order finite difference** *of the function* f, *with step* h, *at point* a_i.

Theorem 10 *If* $0 \leq i < m;\ k, p \in \mathbb{N}$ *and* $1 \leq p + k \leq m - i$, *then*

$$(\triangle_h^p(\triangle_h^k f))(a_i) = \triangle_h^k(\triangle_h^p f)(a_i) = (\triangle_h^{p+k} f)(a_i). \tag{3}$$

Finite differences table: (f_i denotes $f(a_i)$)

a	f	$\triangle_h f$	$\triangle_h^2 f$	...	$\triangle_h^{m-1} f$	$\triangle_h^m f$
a_0	f_0	$\triangle_h f_0$	$\triangle_h^2 f_0$	...	$\triangle_h^{m-1} f_0$	$\triangle_h^m f_0$
a_1	f_1	$\triangle_h f_1$	$\triangle_h^2 f_1$	...	$\triangle_h^{m-1} f_1$	
	...					
a_{m-3}	f_{m-3}	$\triangle_h f_{m-3}$	$\triangle_h^2 f_{m-3}$			
a_{m-2}	f_{m-2}	$\triangle_h f_{m-2}$	$\triangle_h^2 f_{m-2}$			
a_{m-1}	f_{m-1}	$\triangle_h f_{m-1}$				
a_m	f_m					

where

$$\triangle_h^k f_i = \triangle_h^{k-1} f_{i+1} - \triangle_h^{k-1} f_i,\ \ k = 1, ..., m;\ \ i = 0, 1, ..., m-k.$$

Examples.

1. Considering $h = 0.25$, $a = 1$, $a_i = a + ih$, $i = \overline{0,4}$, and $f_0 = 0$, $f_1 = 2$, $f_2 = 6$, $f_3 = 14$, $f_4 = 17$ form the finite differences table.

Sol.: We get:

a	f	$\triangle_h f$	$\triangle_h^2 f$	$\triangle_h^3 f$	$\triangle_h^4 f$
1	0	2	2	2	-11
1.25	2	4	4	-9	
1.50	6	8	-5		
1.75	14	3			
2	17				

2. For $f(x) = e^x$ find $(\triangle_h^k f)(a_i)$, with $a_i = a + ih$, $i \in \mathbb{N}$.

Divided differences

Let $X = \{x_i \mid x_i \in \mathbb{R},\ i = 0, 1, ..., m,\ m \in \mathbb{N}^*\}$ and $f : X \to \mathbb{R}$.

Definition 11 *For $r \in \mathbb{N}$, $r < m$,*

$$(\mathcal{D}f)(x_r) := [x_r, x_{r+1}; f] = \frac{f(x_{r+1}) - f(x_r)}{x_{r+1} - x_r}$$

is called **the first order divided difference** *of the function f, regarding the points x_r and x_{r+1}.*

Theorem 12 *The operator $\mathcal{D}$ is linear with respect to f.*

Proof.

$$\begin{aligned}(\mathcal{D}(\alpha f + \beta g))(x_r) &= \frac{(\alpha f + \beta g)(x_{r+1}) - (\alpha f + \beta g)(x_{r)}}{x_{r+1} - x_r} \qquad (4)\\ &= \alpha(\mathcal{D}f)(x_r) + \beta(\mathcal{D}g)(x_r), \qquad \text{for } \alpha, \beta \in \mathbb{R}.\end{aligned}$$

■

Definition 13 *Let* $r, k \in \mathbb{N}, 0 \leq r < m$ *and* $1 \leq k \leq m - r,\ m \in \mathbb{N}^*$. *The quantity*

$$(\mathcal{D}^k f)(x_r) = \frac{(\mathcal{D}^{k-1} f)(x_{r+1}) - (\mathcal{D}^{k-1} f)(x_r)}{x_{r+k} - x_r}, \quad \textit{with } \mathcal{D}^0 = 1,\ \mathcal{D}^1 = \mathcal{D}, \tag{5}$$

is called **the k-th order divided difference** *of the function f, at x_r.*

$(\mathcal{D}^k f)(x_r)$ is also denoted by $\left[x_r, ..., x_{r+k}; f\right]$. Relation (5) can be written as

$$\left[x_r, ..., x_{r+k}; f\right] = \frac{\left[x_{r+1}, ..., x_{r+k}; f\right] - \left[x_r, ..., x_{r+k-1}; f\right]}{x_{r+k} - x_r}. \tag{6}$$

Remark 14 *The operator $\mathcal{D}^k$ is linear with respect to f.*

For $r = 0$ and $k = m$ we have

$$(\mathcal{D}^m f)(x_0) = \sum_{i=0}^{m} \frac{f(x_i)}{(x_i - x_0)...|...(x_i - x_m)}. \tag{7}$$

Theorem 15 *If* $f, g : X \to \mathbb{R}$ *then*

$$[x_0, ..., x_m; fg] = \sum_{k=0}^{m} [x_0, ..., x_k; f][x_k, ..., x_m; g].$$

Proof. The proof follows by complete induction with respect to m. ■

Table of divided differences:

x	f	$\mathcal{D}f$	$\mathcal{D}^2 f$	...	$\mathcal{D}^{\mathbf{m-1}} f$	$\mathcal{D}^m f$
x_0	f_0	$\mathcal{D}f_0$	$\mathcal{D}^2 f_0$	...	$\mathcal{D}^{m-1} f_0$	$\mathcal{D}^m f_0$
x_1	f_1	$\mathcal{D}f_1$	$\mathcal{D}^2 f_1$		$\mathcal{D}^{m-1} f_1$	
x_2	f_2	$\mathcal{D}f_2$	$\mathcal{D}^2 f_2$			
...	...	...				
x_{m-2}	f_{m-2}	$\mathcal{D}f_{m-2}$	$\mathcal{D}^2 f_{m-2}$			
x_{m-1}	f_{m-1}	$\mathcal{D}f_{m-1}$				
x_m	f_m					

with $f_i = f(x_i), \quad i = 0, 1, ..., m.$

Example 16 *For* $x_0 = 0$, $x_1 = 1$, $x_2 = 2$, $x_3 = 4$ *and* $f_0 = 3$, $f_1 = 4$, $f_2 = 7$, $f_3 = 19$ *form the divided differences table.*

x	f	Df	D^2f	D^3f
0	3	1	1	0
1	4	3	1	
2	7	6		
4	19			

Example 17 *Form the divided differences table for* $x_0 = 2$, $x_1 = 4$, $x_2 = 6$, $x_3 = 8$ *and* $f_0 = 4$, $f_1 = 8$, $f_2 = 20$, $f_3 = 48$.

Example 18 *Form the divided differences table for* $x_0 = 1$, $x_1 = 2$, $x_2 = 3$, $x_3 = 5$, $x_4 = 7$ *and* $f_0 = 3$, $f_1 = 5$, $f_2 = 9$, $f_3 = 11$, $f_4 = 15$.

Chapter 2. Polynomial interpolation

Interpolation is the science of ”*reading between the lines of a mathematical table*” (E. Whittaker, G. Robinson)

Assume we know only some values $f(x_i)$, $i = 0, ..., m$ of a function f.

x	x_0,	x_1,	...	z	...	x_m
$y = f(x)$	y_0,	y_1,	...	?	...	y_m

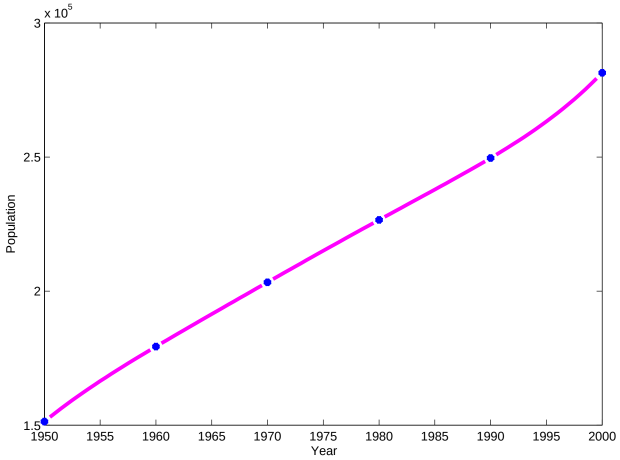
Is there a way to compute or approximate the function value $f(z)$ for some given z without evaluating f?

Applications:

1. approximating data at points where measurements are not available
2. sketching a function f which is expensive to evaluate if it is evaluated at some given points.

Example 19 *A census of the population of the United States is taken every 10 years. The following table lists the population, in thousands of people, from 1950 to 2000.*

1950	1960	1970	1980	1990	2000
151326	179323	203302	226542	249633	281422



Question: these data could be used to provide a reasonable estimate of the population in 1975? Answer: population in 1975 is 215042.

Predictions of this type can be obtained by using a function that fits the given data. This process is called **interpolation.** (If the desired value z is within the range of the interpolation points x_i, then we have *interpolation;* if z is outside the range, the process is called *extrapolation.)*

Example 20 *a) Some values obtained by physical measurements: The temperature of the air outside a house during a day:*

t	8*am*	9*am*	11*am*	1*pm*	5*pm*
T *in* $^\circ C$	12.1	13.6	15.9	18.5	16.1

What temperature was at 10am?

b) Estimate $\sin 1$ *if we know* $\sin\frac{\pi}{6} = \frac{1}{2}, \sin\frac{\pi}{4} = \frac{\sqrt{2}}{2}$ *and* $\sin\frac{\pi}{3} = \frac{\sqrt{3}}{2}$.

c) Estimate $\log_{10}(z)$ *for values of the argument* z *that are intermediate between the tabulated values.*

One of the most useful classes of functions mapping the set of real numbers into itself is the polynomials.

Polynomials are used as the basic means of approximation in nearly all areas of numerical analysis: the solutions of equations, the approximation of functions, of integrals and derivatives, solutions of integral and differential equations, etc.

Polynomials owe this popularity to their simple structure, which makes it easy to construct effective approximations and then make use of them.

Advantages:

- Easy to handle calculus with polynomials: the derivative and the primitive of a polynomial are easy to determine and are also polynomials. The evaluation of a polynomial can be made efficient (the Horner scheme).

- Properly chosen, they can approximate arbitrary well any continuous function:

(Weierstrass Approximation Theorem) Given any $f \in C[a, b]$, $\forall \varepsilon > 0$ (arbitr. small) $\exists P(x)$ polynomial that is as "close" to f as desired:

$$|f(x) - P(x)| < \varepsilon, \quad \forall x \in [a, b].$$

Of course, the smaller ε, the greater the degree of P may become.

2.1. Taylor interpolation

Theorem 21 *(Taylor theorem) Let $f \in C^n[a,b]$, such that there exists $f^{(n+1)}$ on $[a,b]$ and consider $x_0 \in [a,b]$. The Taylor polynomial is*

$$T_n(x) = \sum_{k=0}^{n} \tfrac{(x-x_0)^k}{k!} f^{(k)}(x_0) \tag{8}$$

and we have the approximation formula

$$f(x) = T_n(x) + R_n(x),$$

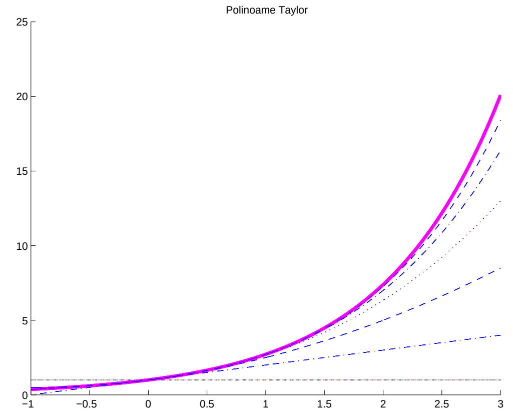
R_n denoting the remainder (the error).

For $\forall x \in [a,b]$ there exists a number ξ between x_0 and x such that

$$R_n(x) = \tfrac{(x-x_0)^{n+1}}{(n+1)!} f^{(n+1)}(\xi).$$

Remark 22 *Taylor polynomials agree as closely as possible with a given function around the specific point x_0, but not on the entire interval.*

Example 23 *We calculate the first six Taylor polynomials about* $x_0 = 0$ *for* $f(x) = e^x$*.*



Notice that even for the higher-degree polynomials, the error becomes progressively worse as we move away from $x_0 = 0$.

Example 24 *Consider* $f(x) = \frac{1}{x}$ *and* $x_0 = 1$*. Approximate the value of* $f(3)$ *by the first and the second degree Taylor polynomials.*

n	0	1	2
$T_n(3)$	1	-1	3

Taylor polynomial approximation is used when approximations are needed only at numbers close to x_0. It is more efficient to use methods that include information at various points.

2.2. Lagrange interpolation

Let $[a, b] \subset \mathbb{R},\ x_i \in [a, b],\ i = 0, 1, ..., m$ such that $x_i \neq x_j$ for $i \neq j$ and consider $f : [a, b] \to \mathbb{R}$.

The Lagrange interpolation problem (LIP) consists in determining the polynomial P of the smallest degree for which

$$P(x_i) = f(x_i),\ i = 0, 1, ..., m \tag{9}$$

i.e., the polynomial of the smallest degree which passes through the distinct points $(x_i, f(x_i))$, $i = 0, 1, ..., m$.

Since in (9) there are $m + 1$ conditions to be satisfied, we need $m + 1$ degrees of freedom. Consider the m-th degree polynomial

$$P(x) = a_0 + a_1 x + ... + a_{m-1} x^{m-1} + a_m x^m. \tag{10}$$

The $m+1$ coefficients $\{a_i\}$ have to be determined in such way that (9) are satisfied. This leads to the linear system of equations:

$$\begin{cases} a_0 + a_1 x_0 + ... + a_{m-1} x_0^{m-1} + a_m x_0^m = f(x_0) \\ a_0 + a_1 x_1 + ... + a_{m-1} x_1^{m-1} + a_m x_1^m = f(x_1) \\ \\ a_0 + a_1 x_m + ... + a_{m-1} x_m^{m-1} + a_m x_m^m = f(x_m). \end{cases}$$

Written in the matrix form, the system is

$$\underbrace{\begin{pmatrix} 1 & x_0 & ... & x_0^{m-1} & x_0^m \\ 1 & x_1 & ... & x_1^{m-1} & x_1^m \\ \vdots & & & & \\ 1 & x_m & ... & x_m^{m-1} & x_m^m \end{pmatrix}}_{V} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{pmatrix} = \begin{pmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_m). \end{pmatrix}.$$

The matrix V with the special structure containing the powers of the nodes is called a Vandermonde matrix.

Remark 25 *For $m+1$ distinct nodes the Vandermonde matrix is nonsingular and there exists a unique interpolating polynomial P of degree less or equal to m with $P(x_i) = f(x_i),\ i = 0, 1, ..., m.$*

Remark 26 *Because the Vandermonde matrix is ill conditioned this method is not recomended for computing the Lagrange polynomial.*

Definition 27 *A solution of (LIP) is called* **Lagrange interpolation polynomial**, *denoted by $L_m f$.*

Remark 28 *We have $(L_m f)(x_i) = f(x_i),\ i = 0, 1, ..., m.$*

$L_m f \in \mathbb{P}_m$ ($\mathbb{P}_m$ is the space of polynomials of at most m-th degree).

The Lagrange interpolation polynomial is given by

$$(L_m f)(x) = \sum_{i=0}^{m} \ell_i(x) f(x_i), \tag{11}$$

where by $\ell_i(x)$ denote **the Lagrange fundamental interpolation polynomials.**

We have

$$u(x) = \prod_{j=0}^{m} (x - x_j),$$

$$u_i(x) = \frac{u(x)}{x - x_i} = (x - x_0)...(x - x_{i-1})(x - x_{i+1})...(x - x_m) = \prod_{\substack{j=0 \\ j \neq i}}^{m} (x - x_j)$$

and

$$\ell_i(x) = \frac{u_i(x)}{u_i(x_i)} = \frac{(x - x_0)...(x - x_{i-1})(x - x_{i+1})...(x - x_m)}{(x_i - x_0)...(x_i - x_{i-1})(x_i - x_{i+1})...(x_i - x_m)} = \prod_{\substack{j=0 \\ j \neq i}}^{m} \frac{x - x_j}{x_i - x_j}, \tag{12}$$

for $i = 0, 1, ..., m$.

How do we know that the interpolation polynomial expanded in powers of x as in (10) and the polynomial constructed as in (11) represent the same polynomial?

Assume we have computed two interpolating polynomials $Q(x)$ and $P(x)$ each of degree m such that

$$Q(x_j) = f(x_j) = P(x_j), \quad j = 0, ..., m.$$

Then we can form the difference

$$d(x) = Q(x) - P(x),$$

that is a polynomial of degree less or equal to m.

Because of the interpolation property of P and Q, we have

$$d(x_j) = Q(x_j) - P(x_j) = 0, \quad j = 0, ..., m.$$

A non-zero polynomial of degree less than or equal to m cannot have more than m zeros. But d has $m + 1$ distinct zeros, hence it must be identically zero, so $Q(x) = P(x)$.

Proposition 29 *We also have*

$$\ell_i(x) = \frac{u(x)}{(x - x_i)u'(x_i)}, \ i = 0, 1, ..., m. \tag{13}$$

Proof. We have $u_i(x) = \frac{u(x)}{x - x_i}$, so $u(x) = u_i(x)(x - x_i)$. We get $u'(x) = u_i(x) + (x - x_i)u_i'(x)$, whence it follows $u'(x_i) = u_i(x_i)$. So, as

$$\ell_i(x) = \frac{u_i(x)}{u_i(x_i)}$$

we get

$$\ell_i(x) = \frac{u_i(x)}{u'(x_i)} = \frac{u(x)}{(x - x_i)u'(x_i)}, \;\; i = 0, 1, ..., m. \tag{14}$$

■

COURSE 2

Lagrange interpolation (continuation)

Let $[a,b] \subset \mathbb{R}$, $x_i \in [a,b]$, $i = 0, 1, ..., m$ such that $x_i \neq x_j$ for $i \neq j$ and consider $f : [a,b] \to \mathbb{R}$. The Lagrange interpolation polynomial is given by

$$(L_m f)(x) = \sum_{i=0}^{m} \ell_i(x) f(x_i), \tag{1}$$

where by $\ell_i(x)$ denote **the Lagrange fundamental interpolation polynomials.**

We have

$$u(x) = \prod_{j=0}^{m} (x - x_j),$$

$$u_i(x) = \frac{u(x)}{x - x_i} = (x - x_0)...(x - x_{i-1})(x - x_{i+1})...(x - x_m) = \prod_{\substack{j=0 \\ j \neq i}}^{m} (x - x_j)$$

and

$$\ell_i(x) = \frac{u_i(x)}{u_i(x_i)} = \frac{(x-x_0)...(x-x_{i-1})(x-x_{i+1})...(x-x_m)}{(x_i-x_0)...(x_i-x_{i-1})(x_i-x_{i+1})...(x_i-x_m)} = \prod_{\substack{j=0\\ j\neq i}}^{m} \frac{x-x_j}{x_i-x_j}, \tag{2}$$

for $i = 0, 1, ..., m$.

Theorem 1 *The operator L_m is linear.*

Proof.

$$\begin{aligned} L_m(\alpha f + \beta g)(x) &= \sum_{i=0}^{m} \ell_i(x)(\alpha f + \beta g)(x_i) = \sum_{i=0}^{m} [\ell_i(x)\alpha f(x_i) + \ell_i(x)\beta g(x_i)] \\ &= \alpha(L_m f)(x) + \beta(L_m g)(x), \end{aligned}$$

so

$$L_m(\alpha f + \beta g) = \alpha L_m f + \beta L_m g, \qquad \forall f, g : [a, b] \to \mathbb{R} \text{ and } \alpha, \beta \in \mathbb{R}.$$

■

Example 2 *a) Consider the nodes x_0, x_1 and a function f to be interpolated.*

b) Find the Lagrange polynomial that interpolates the data in the following table and find the approximative value of $f(-0.5)$.

x	-1	0	3
$f(x)$	8	-2	4

Sol.

a) We have $m = 1$,

$$\begin{aligned} u(x) &= (x - x_0)(x - x_1) \\ u_0(x) &= x - x_1 \\ u_1(x) &= x - x_0 \end{aligned}$$

$$\begin{aligned} (L_1 f)(x) &= l_0(x) f(x_0) + l_1(x) f(x_1) \\ &= \frac{x - x_1}{x_0 - x_1} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1), \end{aligned}$$

which is the line passing through the given points $(x_0, f(x_0))$ and $(x_1, f(x_1))$.

b) We have $m = 2$. The Lagrange polynomial is

$$(L_2 f)(x) = l_0(x) f(x_0) + l_1(x) f(x_1) + l_2(x) f(x_2).$$

$u(x) = (x+1)(x-0)(x-3)$ and it follows

$$l_0(x) = \frac{(x-0)(x-3)}{(-1-0)(-1-3)} = \frac{1}{4}x(x-3)$$
$$l_1(x) = \frac{(x+1)(x-3)}{(0+1)(0-3)} = -\frac{1}{3}(x+1)(x-3)$$
$$l_2(x) = \frac{(x+1)(x-0)}{(3+1)(3-0)} = \frac{1}{12}x(x+1),$$

The polynomial is

$$(L_2f)(x) = 2x(x-3) + \frac{2}{3}(x+1)(x-3) + \frac{1}{3}x(x+1).$$

and $(L_2f)(-0.5) = 2.25$.

Remark 3 *Disadvantages of the form (1) of Lagrange polynomial: requires many computations and if we add or substract a point we have to start with a complete new set of computations.*

Some calculations allow us to reduce the number of operations:

$$(L_m f)(x) = \frac{(L_m f)(x)}{1} = \frac{\sum\limits_{i=0}^{m} l_i(x) f(x_i)}{\sum\limits_{i=0}^{m} l_i(x)}.$$

Dividing the numerator and the denominator by

$$u(x) = \prod_{i=1}^{m} (x - x_i)$$

and denoting

$$A_i = \frac{1}{\prod\limits_{j=0, j\neq i}^{m} (x_i - x_j)} = \frac{1}{u_i(x_i)}$$

one obtains

$$(L_m f)(x) = \frac{\sum\limits_{i=0}^{m} \frac{A_i f(x_i)}{x - x_i}}{\sum\limits_{i=0}^{m} \frac{A_i}{x - x_i}}, \tag{3}$$

called **the barycentric form** *of Lagrange interpolation polynomial*.

Remark 4 *Formula (3) needs half of the number of arithmetic operations needed for (1) and it is easier to add or substract a point.*

The Lagrange polynomial generates **the Lagrange interpolation formula**

$$f = L_m f + R_m f,$$

where $R_m f$ denotes **the remainder** (**the error**).

Theorem 5 *Let* $\alpha = \min\{x, x_0, ..., x_m\}$ *and* $\beta = \max\{x, x_0, ..., x_m\}$. *If* $f \in C^m[\alpha, \beta]$ *and* $f^{(m)}$ *is derivable on* (α, β) *then* $\forall x \in (\alpha, \beta)$, *there exists* $\xi \in (\alpha, \beta)$ *such that*

$$(R_m f)(x) = \frac{u(x)}{(m+1)!} f^{(m+1)}(\xi). \tag{4}$$

Proof. Consider

$$F(z) = \begin{vmatrix} u(z) & (R_m f)(z) \\ u(x) & (R_m f)(x) \end{vmatrix}.$$

From hypothesis it follows that $F \in C^m[\alpha, \beta]$ and there exists $F^{(m+1)}$ on (α, β).

We have

$$F(x) = 0, \;\; F(x_i) = 0, \qquad i = 0, 1, ..., m,$$

as

$$u(x_i) = \prod_{j=0}^{m} (x_i - x_j) = 0$$

and

$$(R_m f)(x_i) = f(x_i) - (L_m f)(x_i) = f(x_i) - f(x_i) = 0,$$

so F has $m + 2$ distinct zeros in (α, β). Applying successively the Rolle theorem it follows that: F has $m + 2$ zeros in $(\alpha, \beta) \Rightarrow F'$ has at least $m + 1$ zeros in $(\alpha, \beta) \Rightarrow ... \Rightarrow F^{(m+1)}$ has at least one zero in (α, β)

So $F^{(m+1)}$ has at least one zero $\xi \in (\alpha, \beta)$, $F^{(m+1)}(\xi) = 0$.

We have

$$F^{(m+1)}(z) = \begin{vmatrix} u^{(m+1)}(z) & (R_m f)^{(m+1)}(z) \\ u(x) & (R_m f)(x) \end{vmatrix},$$

with

$$u(z) = \prod_{i=0}^{m} (z - z_i) \Rightarrow u^{(m+1)}(z) = (m+1)!,$$

and

$$\begin{aligned} (R_m f)^{(m+1)}(z) &= (f - (L_m f))^{(m+1)}(z) \\ &= f^{(m+1)}(z) - (L_m f)^{(m+1)}(z) = f^{(m+1)}(z) \end{aligned}$$

(as, $L_m f \in \mathbb{P}_m$).

We have $F^{(m+1)}(\xi) = 0$, for $\xi \in (\alpha, \beta)$, so

$$F^{(m+1)}(\xi) = \begin{vmatrix} (m+1)! & f^{(m+1)}(\xi) \\ u(x) & (R_m f)(x) \end{vmatrix} = 0,$$

i.e., $(m+1)!(R_m f)(x) = u(x) f^{(m+1)}(\xi)$, whence $(R_m f)(x) = \frac{u(x)}{(m+1)!} f^{(m+1)}(\xi)$. ■

Corollary 6 *If $f \in C^{m+1}[a,b]$ then*

$$|(R_m f)(x)| \leq \frac{|u(x)|}{(m+1)!} \left\| f^{(m+1)} \right\|_\infty, \qquad x \in [a,b]$$

where $\|\cdot\|_\infty$ denotes the uniform norm, and $\|f\|_\infty = \max_{x \in [a,b]} |f(x)|$.

Example 7 *If we know that $\lg 2 = 0.301$, $\lg 3 = 0.477$, $\lg 5 = 0.699$, find $\lg 76$. Study the approximation error.*

Example 8 *Which is the limit of the error for computing $\sqrt{115}$ using Lagrange interpolation formula for the nodes $x_0 = 100$, $x_1 = 121$ and $x_2 = 144$? Find the approximative value of $\sqrt{115}$.*

COURSE 3

The Aitken's algorithm

Let $[a,b] \subset \mathbb{R},\ x_i \in [a,b],\ i = 0, 1, ..., m$ such that $x_i \neq x_j$ for $i \neq j$ and consider $f : [a,b] \to \mathbb{R}$.

Usually, for a practical approximation problem, for a given function $f : [a,b] \to \mathbb{R}$ we have to find the approximation of $f(\alpha)$, $\alpha \in [a,b]$ with an error not greater than a given $\varepsilon > 0$.

If we have enough information about f and its derivatives, we use the inequality $|(R_m f)(x)| \leq \varepsilon$ to find m such that $(L_m f)(\alpha)$ approximates $f(\alpha)$ with the given precision.

We may use the condition $\frac{|u(x)|}{(m+1)!} \left\| f^{(m+1)} \right\|_\infty \leq \varepsilon$, but it should be known $\left\| f^{(m+1)} \right\|_\infty$ or a majorant of it.

A practical method for computing the Lagrange polynomial is **the Aitken's algorithm.** This consists in generating the table:

$$\begin{array}{l|l|l|lllll}
x_0 & f_{00} & & & & & \\
x_1 & f_{10} & f_{11} & & & & \\
x_2 & f_{20} & f_{21} & f_{22} & & & \\
x_3 & f_{30} & f_{31} & f_{32} & f_{33} & & \\
\vdots & \vdots & \vdots & \vdots & \vdots & & \\
x_m & f_{m0} & f_{m1} & f_{m2} & f_{m3} & \cdots & f_{mm}
\end{array}$$

where

$$f_{i0} = f(x_i), \quad i = 0, 1, ..., m,$$

and

$$f_{i,j+1} = \frac{1}{x_i - x_j} \begin{vmatrix} f_{jj} & x_j - x \\ f_{ij} & x_i - x \end{vmatrix}, \qquad i = 0, 1, ..., m; j = 0, ..., i - 1.$$

For example,

$$\begin{aligned} f_{11} &= \frac{1}{x_1 - x_0} \begin{vmatrix} f_{00} & x_0 - x \\ f_{10} & x_1 - x \end{vmatrix} \\ &= \frac{1}{x_1 - x_0} [f_{00}(x_1 - x) - f_{10}(x_0 - x)] \\ &= \frac{x - x_1}{x_0 - x_1} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1) = (L_1 f)(x), \end{aligned}$$

so f_{11} is the value in x of Lagrange polynomial for the nodes x_0, x_1. We have

$$f_{ii} = (L_i f)(x),$$

$L_i f$ being Lagrange polynomial for the nodes $x_0, x_1, ..., x_i$.

So $f_{11}, f_{22}, ..., f_{ii}, ..., f_{mm}$ is a sequence of approximations of $f(x)$.

If the interpolation procedure is convergent then the sequence is also convergent, i.e., $\lim\limits_{m \to \infty} f_{mm} = f(x)$. By Cauchy convergence criterion it follows

$$\lim_{i \to \infty} |f_{ii} - f_{i-1,i-1}| = 0.$$

This could be used as a stopping criterion, i.e.,

$$\left|f_{ii} - f_{i-1,i-1}\right| \leq \varepsilon, \quad \text{for a given precision } \varepsilon > 0.$$

Recommendation is to sort the nodes $x_0, x_1, ..., x_m$ with respect to the distance to x, such that

$$|x_i - x| \leq \left|x_j - x\right| \text{ if } i < j, \quad i, \quad j = 1, ..., m.$$

Example 1 *Approximate* $\sqrt{115}$ *with precision* $\varepsilon = 10^{-3}$, *using Aitken's algorithm.*

Newton interpolation polynomial

A useful representation for Lagrange interpolation polynomial is

$$(L_m f)(x) := (N_m f)(x) = f(x_0) + \sum_{i=1}^{m} (x - x_0)...(x - x_{i-1})(D^i f)(x_0) \tag{1}$$

$$= f(x_0) + \sum_{i=1}^{m} (x - x_0)...(x - x_{i-1})[x_0, ..., x_i; f],$$

which is called **Newton interpolation polynomial;** where $(D^if)(x_0)$(or denoted $[x_0, ..., x_i; f]$) is the i-th order divided difference of the function f at x_0, given by the table

	f	$\mathcal{D}f$	$\mathcal{D}^2f$	...	$\mathcal{D}^{\mathbf{m-1}}f$	$\mathcal{D}^mf$
x_0	f_0	$\mathcal{D}f_0$	$\mathcal{D}^2f_0$	...	$\mathcal{D}^{m-1}f_0$	$\mathcal{D}^mf_0$
x_1	f_1	$\mathcal{D}f_1$	$\mathcal{D}^2f_1$		$\mathcal{D}^{m-1}f_1$	
x_2	f_2	$\mathcal{D}f_2$	$\mathcal{D}^2f_2$			
...	...	...				
x_{m-2}	f_{m-2}	$\mathcal{D}f_{m-2}$	$\mathcal{D}^2f_{m-2}$			
x_{m-1}	f_{m-1}	$\mathcal{D}f_{m-1}$				
x_m	f_m					

Newton interpolation formula is

$$f = N_mf + R_mf,$$

where R_mf denotes the remainder.

Assume that we add the point $(x, f(x))$ at the top of the table of divided differences:

	f	Df	...	$D^{m+1}f$
x	$f(x)$	$(Df)(x) = [x, x_0; f]$		$[x, x_0, ..., x_m; f]$
x_0	$f(x_0)$	$(Df)(x_0) = [x_0, x_1; f]$	...	
x_1	$f(x_1)$	$(Df)(x_1) = [x_1, x_2; f]$		
...	...	...		
x_{m-1}	$f(x_{m-1})$	$(Df)(x_{m-1}) = [x_{m-1}, x_m; f]$		
x_m	$f(x_m)$			

For obtaining the interpolation polynomial we consider

$$[x, x_0; f] = \frac{f(x_0) - f(x)}{x_0 - x} \Longrightarrow f(x) = f(x_0) + (x - x_0)[x, x_0; f] \tag{2}$$

$$[x, x_0, x_1; f] = \frac{[x_0, x_1; f] - [x, x_0; f]}{x_1 - x} \tag{3}$$

$$\Longrightarrow [x, x_0; f] = [x_0, x_1; f] + (x - x_1)[x, x_0, x_1; f].$$

Inserting (3) in (2) we get

$$f(x) = f(x_0) + (x - x_0)[x_0, x_1; f] + (x - x_0)(x - x_1)[x, x_0, x_1; f].$$

If we continue eliminating the divided differences involving x in the same way, we get

$$f(x) = (N_m f)(x) + (R_m f)(x)$$

with

$$(N_m f)(x) = f(x_0) + \sum_{i=1}^{m} (x - x_0)...(x - x_{i-1})[x_0, ..., x_i; f]$$

and the remainder (the error) given by

$$(R_m f)(x) = (x - x_0)...(x - x_m)[x, x_0, ..., x_m; f]. \tag{4}$$

Remark 2 *The remainder for Lagrange interpolation formula is also given by*

$$(R_m f)(x) = \frac{(x - x_0)...(x - x_m)}{(m+1)!} f^{(m+1)}(\xi),$$

with ξ *between* $x, x_0, ..., x_m$, *so, by (4), it follows that* **the divided differences are approximations of the derivatives**

$$[x, x_0, ..., x_m; f] = \frac{f^{(m+1)}(\xi)}{(m+1)!}.$$

Remark 3 *We notice that*

$$(N_i f)(x) = (N_{i-1} f)(x) + (x - x_0)...(x - x_{i-1})[x_0, ..., x_i; f]$$

so the Newton polynomials of degree $2, 3, ...,$ *can be iteratively generated, similarly to Aitken's algorithm.*

Example 4 *Find* L_2f *for* $f(x) = \sin \pi x$, *and* $x_0 = 0, x_1 = \frac{1}{6}, x_2 = \frac{1}{2}$, *in both forms.*

Sol. *a) We have* $u(x) = x(x-\frac{1}{6})(x-\frac{1}{2})$; $u_0(x) = (x-\frac{1}{6})(x-\frac{1}{2})$; $u_1(x) = x(x-\frac{1}{2})$; $u_2(x) = x(x-\frac{1}{6})$

$$\begin{aligned}(L_2f)(x) &= \sum_{i=0}^{2} l_i(x)f(x_i) = \sum_{i=0}^{2} \frac{u_i(x)}{u_i(x_i)}f(x_i)\\ &= \frac{(x-\frac{1}{6})(x-\frac{1}{2})}{(-\frac{1}{6})(-\frac{1}{2})}0 + \frac{x(x-\frac{1}{2})}{\frac{1}{6}(-\frac{1}{3})}\frac{1}{2} + \frac{x(x-\frac{1}{6})}{\frac{1}{2}\cdot\frac{1}{3}}1\\ &= -3x^2 + \frac{7}{2}x.\end{aligned}$$

b)

$$\begin{aligned}(N_2f)(x) &= f(0) + \sum_{i=1}^{2}(x-x_0)...(x-x_{i-1})(D^if)(x_0)\\ &= f(0) + (x-x_0)(Df)(x_0) + (x-x_0)(x-x_1)(D^2f)(x_0)\\ &= x(Df)(x_0) + x(x-\frac{1}{6})(D^2f)(x_0)\end{aligned}$$

The table of divided differences:

x	f	Df	$D^2 f$
0	0	3	-3
$\frac{1}{6}$	$\frac{1}{2}$	$\frac{3}{2}$	
$\frac{1}{2}$	1		

so

$$(N_2 f)(x) = 3x - 3x(x - \frac{1}{6}) = -3x^2 + \frac{7}{2}x.$$

2.3. Hermite interpolation

Example 5 *In the following table there are some data regarding a moving car. We may estimate the position (and the speed) of the car when the time is $t = 10$ using Hermite interpolation.*

Time	0	3	5	8	13
Distance	0	225	383	623	993
Speed	75	77	80	74	72

Let $x_k \in [a,b]$, $k = 0, 1, ..., m$ be such that $x_i \neq x_j$, for $i \neq j$ and let $r_k \in \mathbb{N}$, $k = 0, 1, ..., m$. Consider $f : [a,b] \to \mathbb{R}$ such that there exist $f^{(j)}(x_k)$, $k = 0, 1, ..., m$; $j = 0, 1, ..., r_k$ and $n = m + r_0 + ... + r_m$.

The Hermite interpolation problem (HIP) consists in determining the polynomial P of the smallest degree for which

$$P^{(j)}(x_k) = f^{(j)}(x_k), \quad k = 0, ..., m; \; j = 0, ..., r_k.$$

Definition 6 *A solution of (HIP) is called* **Hermite interpolation polynomial**, *denoted by* $H_n f$*.*

Hermite interpolation polynomial, $H_n f$, satisfies the interpolation conditions:

$$(H_n f)^{(j)}(x_k) = f^{(j)}(x_k), \quad k = 0, ..., m; \ j = 0, ..., r_k.$$

Hermite interpolation polynomial is given by

$$(H_n f)(x) = \sum_{k=0}^{m} \sum_{j=0}^{r_k} h_{kj}(x) f^{(j)}(x_k) \in \mathbb{P}_n, \tag{5}$$

where $h_{kj}(x)$ denote **the Hermite fundamental interpolation polynomials.** They fulfill the relations:

$h_{kj}^{(p)}(x_\nu) = 0, \ \nu \neq k, \quad p = 0, 1, ..., r_\nu$

$h_{kj}^{(p)}(x_k) = \delta_{jp}, \ p = 0, 1, ..., r_k, \quad$ for $j = 0, 1, ..., r_k$ and $\nu, k = 0, 1, ..., m,$

with $\delta_{jp} = \begin{cases} 1, & j = p \\ 0, & j \neq p. \end{cases}$

We denote by

$$u(x) = \prod_{k=0}^{m} (x - x_k)^{r_k+1} \qquad \text{and } u_k(x) = \frac{u(x)}{(x - x_k)^{r_k+1}}.$$

We have

$$h_{kj}(x) = \frac{(x - x_k)^j}{j!} u_k(x) \sum_{\upsilon=0}^{r_k-j} \frac{(x - x_k)^\nu}{\nu!} \left[\frac{1}{u_k(x)}\right]_{x=x_k}^{(\nu)}. \tag{6}$$

Example 7 *Find the Hermite interpolation polynomial for a function f for which we know $f(0) = 1, f'(0) = 2$ and $f(1) = -3$ (equivalent with $x_0 = 0$ multiple node of order 2 or double node, $x_1 = 1$ simple node).*

Sol. We have $x_0 = 0, x_1 = 1, m = 1, r_0 = 1, r_1 = 0, n = m + r_0 + r_1 = 2$

$$\begin{aligned}(H_2 f)(x) &= \sum_{k=0}^{1} \sum_{j=0}^{r_k} h_{kj}(x) f^{(j)}(x_k) \\ &= h_{00}(x) f(0) + h_{01}(x) f'(0) + h_{10}(x) f(1).\end{aligned}$$

We have h_{00}, h_{01}, h_{10}. These fulfills relations:

$$h_{kj}^{(p)}(x_\nu) = 0,\ \nu \neq k, \quad p = 0, 1, ..., r_\nu$$
$$h_{kj}^{(p)}(x_k) = \delta_{jp},\ p = 0, 1, ..., r_k, \quad \text{for } j = 0, 1, ..., r_k \text{ and } \nu, k = 0, 1, ..., m.$$

We have $h_{00}(x) = a_1 x^2 + b_1 x + c_1 \in \mathbb{P}_2$, with $a_1, b_1, c_1 \in \mathbb{R}$, and the system

$$\begin{cases} h_{00}(x_0) = 1 \\ h'_{00}(x_0) = 0 \\ h_{00}(x_1) = 0 \end{cases} \Leftrightarrow \begin{cases} h_{00}(0) = 1 \\ h'_{00}(0) = 0 \\ h_{00}(1) = 0 \end{cases}$$

that becomes

$$\begin{cases} c_1 = 1 \\ b_1 = 0 \\ a_1 + b_1 + c_1 = 0. \end{cases}$$

Solution is: $a_1 = -1, b_1 = 0, c_1 = 1$ so $h_{00}(x) = -x^2 + 1$.

We have $h_{01}(x) = a_2 x^2 + b_2 x + c_2 \in \mathbb{P}_2$, with $a_2, b_2, c_2 \in \mathbb{R}$. The system is

$$\begin{cases} h_{01}(x_0) = 0 \\ h'_{01}(x_0) = 1 \\ h_{01}(x_1) = 0 \end{cases} \Leftrightarrow \begin{cases} h_{01}(0) = 0 \\ h'_{01}(0) = 1 \\ h_{01}(1) = 0 \end{cases}$$

and we get $h_{01}(x) = -x^2 + x$.

We have $h_{10}(x) = a_3x^2 + b_3x + c_3 \in \mathbb{P}_2$, with $a_3, b_3, c_3 \in \mathbb{R}$. The system is

$$\begin{cases} h_{10}(x_0) = 0 \\ h'_{10}(x_0) = 0 \\ h_{10}(x_1) = 1 \end{cases} \Leftrightarrow \begin{cases} h_{10}(0) = 0 \\ h'_{10}(0) = 0 \\ h_{10}(1) = 1 \end{cases}$$

and we get $h_{10}(x) = x^2$.

The Hermite polynomial is

$$(H_2f)(x) = -x^2 + 1 - 2x^2 + 2x - 3x^2 = -6x^2 + 2x + 1.$$

COURSE 4

The Hermite interpolation formula is

$$f = H_n f + R_n f,$$

where $R_n f$ denotes the remainder term (the error).

Theorem 1 *If* $f \in C^n[\alpha, \beta]$ *and* $f^{(n)}$ *is derivable on* (α, β), *with* $\alpha = \min\{x, x_0, ..., x_m\}$ *and* $\beta = \max\{x, x_0, ..., x_m\}$, *then there exists* $\xi \in (\alpha, \beta)$ *such that*

$$(R_n f)(x) = \frac{u(x)}{(n+1)!} f^{(n+1)}(\xi). \tag{1}$$

Proof. Consider

$$F(z) = \begin{vmatrix} u(z) & (R_n f)(z) \\ u(x) & (R_n f)(x) \end{vmatrix}.$$

$F \in C^n[\alpha, \beta]$ and there exists $F^{(n+1)}$ on (α, β).

We have

$$F(x) = 0, \quad F^{(j)}(x_k) = 0, \qquad k = 0, ..., m; \; j = 0, ..., r_k;$$

because

$$u(x) = \prod_{k=0}^{m} (x - x_k)^{r_k+1} \Rightarrow u^{(j)}(x_k) = 0, \; j = 0, ..., r_k$$

and

$$(R_m f)^{(j)}(x_k) = f^{(j)}(x_k) - (H_n f)^{(j)}(x_k) = f^{(j)}(x_k) - f^{(j)}(x_k) = 0.$$

So, F and its derivatives have $n+2$ distinct zeros in (α, β). Applying successively Rolle's theorem it follows that F' has at least $n+1$ zeros in $(\alpha, \beta) \Rightarrow ... \Rightarrow F^{(n+1)}$ has at least one zero $\xi \in (\alpha, \beta)$, $F^{(n+1)}(\xi) = 0$.

We have

$$F^{(n+1)}(z) = \begin{vmatrix} u^{(n+1)}(z) & (R_n f)^{(n+1)}(z) \\ u(x) & (R_n f)(x) \end{vmatrix},$$

with $u(z) = \prod_{k=0}^{m}(z - z_k)^{r_k+1} \in \mathbb{P}_{n+1} \Rightarrow u^{(n+1)}(z) = (n+1)!$, and $(R_n f)^{(n+1)}(z) = f^{(n+1)}(z) - (H_n f)^{(n+1)}(z) = f^{(n+1)}(z)$ (as, $H_n f \in \mathbb{P}_n$). We get

$$F^{(n+1)}(\xi) = \begin{vmatrix} (n+1)! & f^{(n+1)}(\xi) \\ u(x) & (R_n f)(x) \end{vmatrix} = 0,$$

whence it follows (1). ■

Corollary 2 *If $f \in C^{n+1}[a,b]$ then*

$$|(R_n f)(x)| \leq \frac{|u(x)|}{(n+1)!} \left\| f^{(n+1)} \right\|_\infty, \qquad x \in [a,b]$$

where $\|\cdot\|_\infty$ denotes the uniform norm ($\|f\|_\infty = \max\limits_{x\in[a,b]} |f(x)|$).

Remark 3 *In case of $m = 0$, i.e., $n = r_0$, (HIP) becomes* **Taylor interpolation problem**. *Taylor interpolation polynomial is*

$$(T_n f)(x) = \sum_{j=0}^{n} \frac{(x - x_0)^j}{j!} f^{(j)}(x_0).$$

Example 4 *Find the Hermite interpolation formula for the function* $f(x) = xe^x$ *for which we know* $f(-1) = -0.3679$*,* $f(0) = 0$*,* $f'(0) = 1$*,* $f(1) = 2.7183$*, (equivalent with* $x_0 = -1$ *simple,* $x_1 = 0$ *multiple of order 2 and* $x_2 = 1$ *simple). Which is the limit of the error for approximating* $f(\frac{1}{2})$?

Hermite interpolation with double nodes

Example 5 *In the following table there are some data regarding a moving car. We may estimate the position (and the speed) of the car when the time is $t = 10$ using Hermite interpolation.*

Time	0	3	5	8	13
Distance	0	225	383	623	993
Speed	75	77	80	74	72

Consider $f : [a, b] \to \mathbb{R}$, $x_0, x_1, ..., x_m \in [a, b]$

and the values $f(x_0), f(x_1), ..., f(x_m), f'(x_0), f'(x_1), ..., f'(x_m)$.

The Hermite interpolation polynomial with double nodes, H_{2m+1}, satisfies the interpolation properties:

$$H_{2m+1}(x_i) = f(x_i), \;\; i = \overline{0, m},$$
$$H'_{2m+1}(x_i) = f'(x_i), \;\; i = \overline{0, m}.$$

It is a polynomial of $n = 2m + 1$ degree.

For computation: use Lagrange polynomial written in Newton form, with divided differences table having each node x_i written twice.

Consider $z_0 = x_0$, $z_1 = x_0$, $z_2 = x_1$, $z_3 = x_1$, $\ldots$, $z_{2m} = x_m$, $z_{2m+1} = x_m$.

Form divided differences table: each node appear twice, in the first column write the values of f for each node twice; in the second column, at the odd positions put the values of the derivatives of f; the other elements are computed using the rule from divided differences.

We obtain the following table:

z_0	$f(z_0)$	$(\mathcal{D}^1 f)(z_0) = f'(x_0)$	$(\mathcal{D}^2 f)(z_0)$		$(\mathcal{D}^{2m} f)(z_0)$	$(\mathcal{D}^{2m+1} f)(z_0)$
z_1	$f(z_1)$	$(\mathcal{D}^1 f)(z_1)$	$\vdots$		$(\mathcal{D}^{2m} f)(z_1)$	
z_2	$f(z_2)$	$(\mathcal{D}^1 f)(z_2) = f'(x_1)$				
z_3	$f(z_3)$	$\vdots$				
$\vdots$	$\vdots$	$(\mathcal{D}^1 f)(z_{2m-1})$	$(\mathcal{D}^2 f)(z_{2m-1})$	$\ddots$		
z_{2m}	$f(z_{2m})$	$(\mathcal{D}^1 f)(z_{2m}) = f'(x_m)$		$\ldots$		
z_{2m+1}	$f(z_{2m+1})$			$\ldots$		

Newton interpolation polynomial for the nodes $x_0, ..., x_n$ is

$$(N_n f)(x) = f(x_0) + \sum_{i=1}^{n} (x - x_0)...(x - x_{i-1})(\mathcal{D}^i f)(x_0),$$

and similarly, Hermite interpolation polynomial is

$$(H_{2m+1} f)(x) = f(z_0) + \sum_{i=1}^{2m+1} (x - z_0)...(x - z_{i-1})(\mathcal{D}^i f)(z_0),$$

where $(\mathcal{D}^i f)(z_0)$, $i = 1, ..., 2m+1$ are the elements from the first line and columns $2, ..., 2m+1$.

Example 6 *Consider the double nodes* $x_0 = -1$ *and* $x_1 = 1$, *and* $f(-1) = -3, f'(-1) = 10, f(1) = 1, f'(1) = 2$. *Find the Hermite interpolation polynomial, that approximates the function* f, *in both forms, using the classical formula and using divided differences.*

Sol. We present here the method with divided differences. We have $m = 1, r_0 = r_1 = 1 \Rightarrow n = 3$

$z_0 = -1$	$f(-1) = -3$	$f'(-1) = 10$	$\frac{\frac{f(1)-f(-1)}{2} - f'(-1)}{z_2 - z_0} = -4$	$\frac{0-(-4)}{z_3 - z_0} = 2$
$z_1 = -1$	$f(-1) = -3$	$\frac{f(1)-f(-1)}{z_2 - z_1} = 2$	$\frac{f'(1) - \frac{f(1)-f(-1)}{2}}{z_3 - z_1} = 0$	
$z_2 = 1$	$f(1) = 1$	$f'(1) = 2$		
$z_3 = 1$	$f(1) = 1$			

The Hermite interpolation polynomial is

$$\begin{aligned}(H_3 f)(x) &= f(z_0) + \sum_{i=1}^{3} (x - z_0)...(x - z_{i-1})(\mathcal{D}^i f)(z_0) \\ &= f(z_0) + (x - z_0)(\mathcal{D}^1 f)(z_0) + (x - z_0)(x - z_1)(\mathcal{D}^2 f)(z_0) \\ &+ (x - z_0)(x - z_1)(x - z_2)(\mathcal{D}^3 f)(z_0)\end{aligned}$$

i.e.,

$$\begin{aligned}(H_3 f)(x) =& f(-1) + (x+1)f'(-1) + (x+1)^2 \tfrac{f(1)-f(-1)-2f'(-1)}{4} \\ &+ (x+1)^2 (x-1) \tfrac{2f'(1)-f(1)+f(-1)}{4} \\ =& -3 + 10(x+1) - 4(x+1)^2 + 2(x+1)^2(x-1) \\ =& 2x^3 - 2x^2 + 1.\end{aligned}$$

Example 7 *Considering the the following data*

x	0	2	3
$f(x)$	0	10	12
$f\prime(x)$	5	3	7

find the corresponding Hermite interpolation polynomial.

2.4. Birkhoff interpolation

Let $x_k \in [a,b],\ k = 0,1,...,m,\ x_i \neq x_j$ for $i \neq j, r_k \in \mathbb{N}$ and $I_k \subset \{0,1,...,r_k\},\ k = 0,1,...,m,\ f : [a,b] \to \mathbb{R}$ s.t. $\exists f^{(j)}(x_k),\ k = 0,...,m,\ j \in I_k$, and denote $n = |I_0| + ... + |I_m| - 1$, where $|I_k|$ is the cardinal of the set I_k.

The Birkhoff interpolation problem (BIP) consists in determining the polynomial P of the smallest degree such that

$$P^{(j)}(x_k) = f^{(j)}(x_k), \quad k = 0,...,m;\ j \in I_k.$$

Remark 8 *If $I_k = \{0,1,...,r_k\},\ k = 0,...,m$, then (BIP) reduces to a (HIP). Birkhoff interpolation is also called* lacunary Hermite interpolation.

In order to check if (BIP) has an unique solution, we consider the polynomial $P(x) = a_n x^n + ... + a_0$ and the $(n+1) \times (n+1)$ linear system

$$P^{(j)}(x_k) = f^{(j)}(x_k), \quad k = 0,...,m;\ j \in I_k, \tag{2}$$

having as unknowns the coefficients of the polynomial. If the determinant of the system (2) is nonzero than (BIP) has an unique solution.

Definition 9 *A solution of (BIP), if exists, is called* **Birkhoff interpolation polynomial**, *denoted by* $B_n f$.

Birkhoff interpolation polynomial is given by

$$(B_n f)(x) = \sum_{k=0}^{m} \sum_{j \in I_k} b_{kj}(x) f^{(j)}(x_k), \tag{3}$$

where $b_{kj}(x)$ denote the Birkhoff fundamental interpolation polynomials. They fulfill relations:

$$\begin{aligned} & b_{kj}^{(p)}(x_\nu) = 0,\ \nu \neq k, \quad p \in I_\nu \\ & b_{kj}^{(p)}(x_k) = \delta_{jp},\ p \in I_k, \quad \text{for } j \in I_k \text{ and } \nu, k = 0, 1, ..., m, \end{aligned} \tag{4}$$

with $\delta_{jp} = \begin{cases} 1, & j = p \\ 0, & j \neq p. \end{cases}$

Remark 10 *Because of the gaps of the interpolation conditions, it is hard to find an explicit expression for* b_{kj}, $k = 0, ..., m;\ j \in I_k$. *They are found using relations (4).*

Birkhoff interpolation formula is

$$f = B_n f + R_n f,$$

where $R_n f$ denotes the remainder term.

Example 11 *Let* $f \in C^2[0,1]$, *the nodes* $x_0 = 0$, $x_1 = 1$ *and we suppose that we know* $f(0) = 1$ *and* $f'(1) = \frac{1}{2}$. *Find the corresponding interpolation formula.*

Sol. *We have* $m = 1,\ I_0 = \{0\},\ I_1 = \{1\}$, *so* $n = 1 + 1 - 1 = 1$.

We check if there exists a solution of the problem.

Consider $P(x) = a_1 x + a_0 \in \mathbb{P}_1$ *and the system*

$$\begin{cases} P(0) = f(0) \\ P'(1) = f'(1) \end{cases} \Longleftrightarrow \begin{cases} a_0 = f(0) \\ a_1 = f'(1) \end{cases} .$$

The determinat of the system is

$$\begin{vmatrix} 0 & 1 \\ 1 & 0 \end{vmatrix} = -1 \neq 0,$$

so the problem has an unique solution.

The Birkhoff polynomial is

$$(B_1 f)(x) = b_{00}(x) f(0) + b_{11}(x) f'(1) \in \mathbb{P}_1.$$

We have $b_{00}(x) = ax + b \in \mathbb{P}_1$ *and*

$$\begin{cases} b_{00}(x_0) = 1 \\ b'_{00}(x_1) = 0 \end{cases} \Longleftrightarrow \begin{cases} b_{00}(0) = 1 \\ b'_{00}(1) = 0 \end{cases} \Leftrightarrow \begin{cases} b = 1 \\ a = 0 \end{cases} ,$$

whence

$$b_{00}(x) = 1.$$

For $b_{11}(x) = cx + d \in \mathbb{P}_1$ *we have*

$$\left\{ \begin{array}{l} b_{11}(x_0) = 0 \\ b'_{11}(x_1) = 1 \end{array} \right. \Longleftrightarrow \left\{ \begin{array}{l} b_{11}(0) = 0 \\ b'_{11}(1) = 1 \end{array} \right. \Leftrightarrow \left\{ \begin{array}{l} d = 0 \\ c = 1 \end{array} \right.$$

whence

$$b_{11}(x) = x.$$

So,

$$(B_1 f)(x) = f(0) + x f'(1) = 1 + \frac{1}{2} x.$$

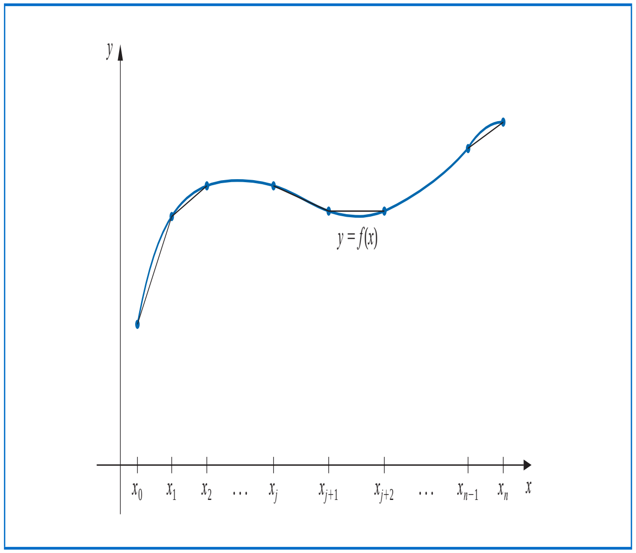
Example 12 *Considering* $f'(0) = 1$, $f(1) = 2$ *and* $f'(2) = 1$. *Find the approximative value of* $f(\frac{1}{2})$.

COURSE 5

2.5. Cubic spline interpolation

Lagrange, Hermite, Birkhoff interpolants of large degrees could oscillate widely; a minor fluctuation over a small portion of the interval can induce large fluctuations over the entire interval.

An alternative: to divide the interval into a collection of subintervals and construct a (generally) different approximating polynomial on each subinterval. This is called **piecewise-polynomial approximation.**



Let $f : [a, b] \to \mathbb{R}$ be the approximating function. Examples of piecewise-polynomial interpolation:

- piecewise-linear interpolation: consists of joining a set of data points $\{(x_0, f(x_0)), (x_1, f(x_1)), ..., (x_n, f(x_n))\}$ by a series of straight lines

  Disadvantage: there is likely no differentiability at the endpoints of the subintervals, (the interpolating function is not "smooth"). Often, from physical conditions, that smoothness is required.

- Hermite interpolation when values of f and f' are known at the points $x_0 < x_1 < ... < x_n$;

  Disadvantage: we need to know f' and this is frequently unavailable.

- spline interpolation: piecewise polynomials that require no specific derivative information, except perhaps at the endpoints of the interval.

Definition 1 *The piecewise-polynomial approximation that uses cubic spline polynomials between each successive pair of nodes is called* **cubic spline interpolation.**

(The word “spline” was used to refer to a long flexible strip, generally of metal, that could be used to draw continuous smooth curves by forcing the strip to pass through specified points and tracing along the curve.)

Definition 2 *Let* $f : [a, b] \to \mathbb{R}$ *and the nodes* $a = x_0 < x_1 < ... < x_n = b$, *a* **cubic spline interpolant** S *for* f *is the function that satisfies the following conditions:*

(a) $S(x)$ *is a cubic polynomial, denoted* $S_j(x)$ *on the subinterval* $[x_j, x_{j+1}]$, $\forall j = 0, 1, ..., n-1$, *i.e.,*

$$S(x) = \begin{cases} S_0(x), & x \in [x_0, x_1] \\ S_1(x), & x \in [x_1, x_2] \\ ... & \\ S_{n-1}(x), & x \in [x_{n-1}, x_n] \end{cases}$$

(b) $S_j(x_j) = f(x_j)$ *and* $S_j(x_{j+1}) = f(x_{j+1})$, $\forall j = 0, 1, ..., n-1$;

(c) $S_j(x_{j+1}) = S_{j+1}(x_{j+1})$, $\forall j = 0, 1, ..., n-2$;

(d) $S_j^{'}(x_{j+1}) = S_{j+1}^{'}(x_{j+1})$, $\forall j = 0, 1, ..., n-2$;

(e) $S_j^{''}(x_{j+1}) = S_{j+1}^{''}(x_{j+1})$, $\forall j = 0, 1, ..., n-2$;

(f) *One of the following boundary conditions is satisfied:*

(i) $S''(x_0) = S''(x_n) = 0$ *(*$\Longleftrightarrow S''_0(x_0) = S''_{n-1}(x_n) = 0$ *natural (or free) boundary)* **natural spline***;*

(ii) $S^{'}(x_0) = f'(x_0)$ *and* $S'(x_n) = f'(x_n)$ *(*$\Longleftrightarrow S^{'}_0(x_0) = f'(x_0)$ *and* $S'_{n-1}(x_n) = f'(x_n)$ *clamped boundary)* **clamped spline***;*

(iii) $S_1(x) = S_2(x)$ *and* $S_{n-2} = S_{n-1}$ *(***de Boor spline***).*

Remark 3 *A cubic spline function defined on an interval divided into* n *subintervals will require determining* $4n$ *constants.*

We have the following expression of a cubic spline:

$$S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3, \quad \forall j = 0, 1, ..., n-1. \quad (1)$$

Theorem 4 *If* f *is defined at* $a = x_0 < x_1 < ... < x_n = b$*, then* f *has an unique natural spline interpolant* S *on the nodes* x_0*,* $x_1, ..., x_n$*; that satisfies the natural boundary conditions* $S''(a) = 0$ *and* $S''(b) = 0$*.*

Theorem 5 *If f is defined at $a = x_0 < x_1 < ... < x_n = b$ and differentiable at a and b, then f has an unique clamped spline interpolant S on the nodes $x_0, x_1, ..., x_n$; that satisfies the clamped boundary conditions $S'(a) = f'(a)$ și $S'(b) = f'(b)$.*

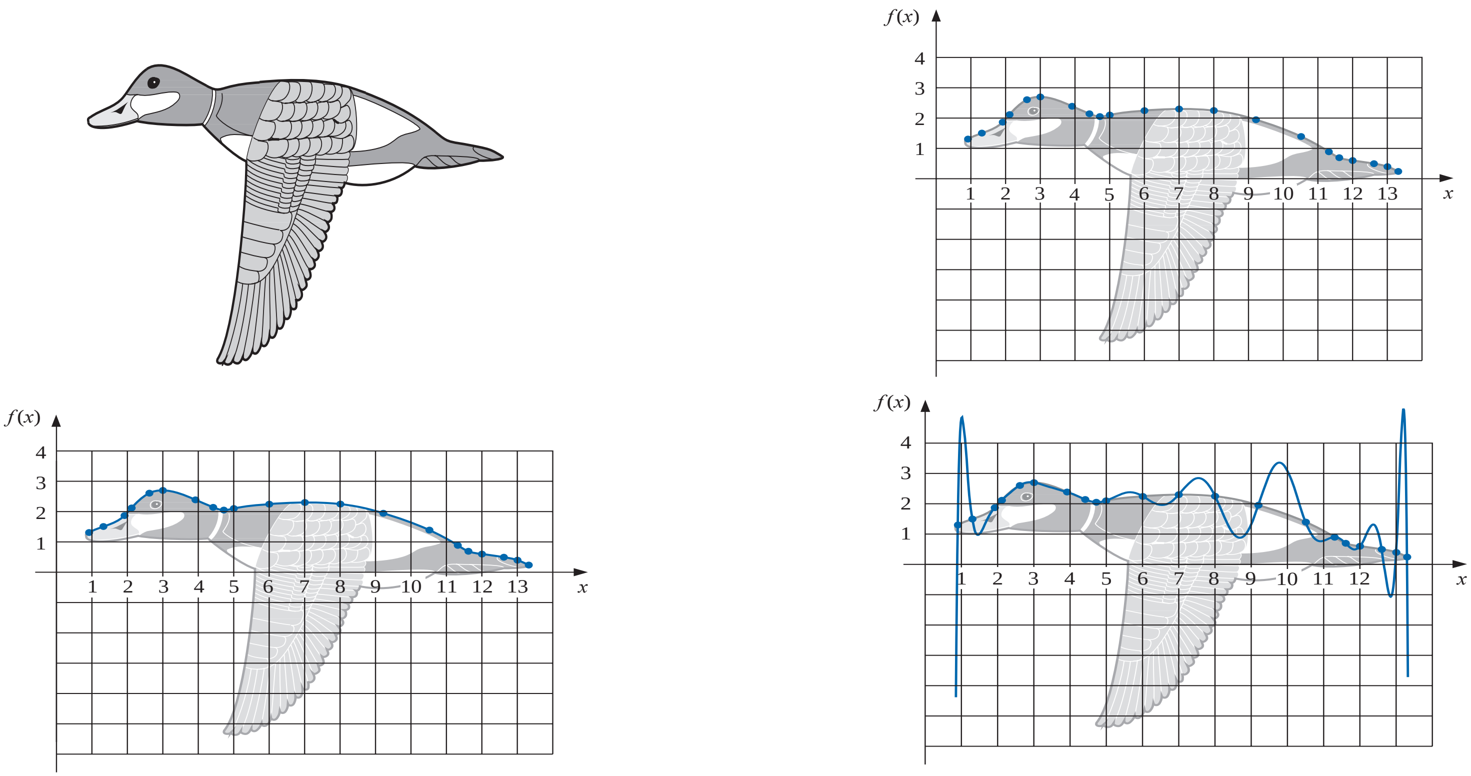
Theorem 6 *Let $f \in C^4[a, b]$ with $\max\limits_{a \leq x \leq b} |f^{(4)}(x)| = M$. If S is the unique clamped cubic spline interpolant to f with respect to the nodes $a = x_0 < x_1 < \cdots < x_n = b$, then for all x in $[a, b]$,*

$$|f(x) - S(x)| \leq \frac{5M}{384} \max_{0 \leq j \leq n-1} (x_{j+1} - x_j)^4.$$

Remark 7 *A fourth-order error-bound result also holds in the case of natural boundary conditions, but it is more difficult to express.*

Remark 8 *The natural boundary conditions will generally give less accurate results than the clamped conditions near the ends of the interval $[x_0, x_n]$ unless the function f happens to nearly satisfy $f''(x_0) = f''(x_n) = 0$.*

Illustration. To approximate the top profile of a duck, we have chosen 21 points along the curve through which we want the approximating curves to pass.



1) The duck in flight. 2) The points. 3) The natural cubic spline. 4) The Lagrange interpolation polynomial.

Example 9 *Construct* **a natural cubic spline** *that passes through the points* $(1, 2)$, $(2, 3)$ *and* $(3, 5)$.

Sol. (Sketch of the solution) *We follow Definition 2:*

Here $S(x)$ *consists of two cubic splines,* $S_j(x)$ *on the subinterval* $[x_j, x_{j+1}]$, $\forall j = 0, 1$, *i.e.,*

$$S(x) = \begin{cases} S_0(x), & x \in [x_0, x_1] \\ S_1(x), & x \in [x_1, x_2] \end{cases}$$

given by (*1*),

$$\begin{aligned} S_0(x) &= a_0 + b_0(x-1) + c_0(x-1)^2 + d_0(x-1)^3, \\ S_1(x) &= a_1 + b_1(x-2) + c_1(x-2)^2 + d_1(x-2)^3. \end{aligned}$$

There are 8 constants ($a_i, b_i, c_i, d_i,\ i = 0, 1$) *to be determined, which requires 8 conditions, that come from (b),(c),(d),(e),(i).*

Example 10 *Construct* **a clamped spline** S *that passes through the points* $(1, 2)$, $(2, 3)$ *and* $(3, 5)$ *and that has* $S'(1) = 2$ *and* $S'(3) = 1$.

2.6. Least squares approximation

- It is an extension of the interpolation problem.

- More desirable when the data are contaminated by errors.

- To estimate values of parameters of a mathematical model from measured data, which are subject to errors.

When we know $f(x_i)$, $i = 0, ..., m$, an interpolation method can be used to determine an approximation φ of the function f, such that

$$\varphi\left(x_i\right) = f\left(x_i\right),\ i = 0, ..., m.$$

If only approximations of $f\left(x_i\right)$ are available or the number of interp. conditions is too large, instead of requiring that the approx. function reproduces $f(x_i)$ exactly, we ask only that it fits the data "as closely as possible".

It seems that the least squares method was first introduced by C. F. Gauss in 1795. In 1801 he used it for making the best prediction for the orbital position of the planet Ceres (dwarf planet, lies between Mars and Jupiter, first considered planet and further reclassified as an asteroid) using the measurements of G. Piazzi (who was the first that discovered it).

The first clear and concise exposition of the method of least squares was first published by A. M. Legendre in 1805. P. S. Laplace and R. Adrain have also contributed to the development of this theory.

In 1809 C. F. Gauss applied the method in calculating the orbits of some celestial bodies. In that work he claimed and proved that he have been in possession of the method since 1795. The least squares approximation φ is determined such that:

- in the discrete case:

$$\left(\sum_{i=0}^{m} \left[f\left(x_i\right) - \varphi\left(x_i\right)\right]^2\right)^{1/2} \to \text{min},$$

- in the continuous case:

$$\left(\int_a^b \left[f\left(x\right)-\varphi\left(x\right)\right]^2 dx\right)^{1/2} \to \text{min},$$

Remark 11 *Notice that the interpolation is a particular case of the least squares approximation, with*

$$f\left(x_i\right)-\varphi\left(x_i\right)=0, \quad i=0,...,m.$$

Linear least square. Consider the data

x	1	2	3	4	5
$f(x)$	1	1	2	2	4

The problem consists in finding a function φ that "best" represents the data.

Plot the data and try to recognize the shape of a "guess function φ" such that $f \approx \varphi$.

For this example, a resonable guess may be a linear one, $\varphi(x) = ax + b$. The problem: find a and b that makes φ the best function to fit the data. The least squares criterion consists in minimizing the sum

$$E(a,b) = \sum_{i=0}^{4} \left[f\left(x_i\right) - \varphi\left(x_i\right)\right]^2 = \sum_{i=0}^{4} \left[f\left(x_i\right) - \left(ax_i + b\right)\right]^2.$$

The minimum of the sum is obtained when

$$\frac{\partial E(a,b)}{\partial a} = 0$$
$$\frac{\partial E(a,b)}{\partial b} = 0.$$

We get

$$\begin{aligned} 55a + 15b &= 37 \\ 15a + 5b &= 10 \end{aligned}$$

and further $\varphi(x) = 0.7x - 0.1$.

Consider a more general problem with the data from the table

x	x_0	x_1	...	x_m
$f(x)$	y_0	y_1	...	y_m

and the approximating linear function $\varphi(x) = ax + b$. We have to find a and b.

We have to minimize the sum

$$E(a,b) = \sum_{i=0}^{m} \left[f\left(x_i\right) - \varphi\left(x_i\right)\right]^2 = \sum_{i=0}^{m} \left[f\left(x_i\right) - \left(ax_i + b\right)\right]^2 . \qquad (2)$$

The minimum of the sum is obtained by

$$\frac{\partial E(a,b)}{\partial a} = 2\sum_{i=0}^{m} \left[f\left(x_i\right) - \left(ax_i + b\right)\right] \cdot \left(-x_i\right) = 0$$

$$\frac{\partial E(a,b)}{\partial b} = 2\sum_{i=0}^{m} \left[f\left(x_i\right) - \left(ax_i + b\right)\right] \cdot (-1) = 0$$

These are called **normal equations**. Further,

$$\sum_{i=0}^{m} x_i f\left(x_i\right) = a\sum_{i=0}^{m} x_i^2 + b\sum_{i=0}^{m} x_i$$

$$\sum_{i=0}^{m} f\left(x_i\right) = a\sum_{i=0}^{m} x_i + (m+1)b.$$

The solution is

$$a = \frac{(m+1)\sum\limits_{i=0}^{m} x_i f\left(x_i\right) - \sum\limits_{i=0}^{m} x_i \sum\limits_{i=0}^{m} f\left(x_i\right)}{(m+1)\sum\limits_{i=0}^{m} x_i^2 - (\sum\limits_{i=0}^{m} x_i)^2} \tag{3}$$

$$b = \frac{\sum\limits_{i=0}^{m} x_i^2 \sum\limits_{i=0}^{m} f\left(x_i\right) - \sum\limits_{i=0}^{m} x_i f\left(x_i\right) \sum\limits_{i=0}^{m} x_i}{(m+1)\sum\limits_{i=0}^{m} x_i^2 - (\sum\limits_{i=0}^{m} x_i)^2}.$$

Example 12 *Having the data*

x	0	1	2	3
$f(x)$	-4	0	4	-2

find the corresponding least squares polynomial of the first degree.

Sol. We have

$$E(a,b) = \sum_{i=0}^{3} \left[f\left(x_i\right) - \varphi\left(x_i\right)\right]^2 = \sum_{i=0}^{3} \left[f\left(x_i\right) - \left(ax_i + b\right)\right]^2 \tag{4}$$

and we have to find a and b from the system

$$\begin{cases} \frac{\partial E(a,b)}{\partial a} = 2 \sum\limits_{i=0}^{3} \left[f\left(x_i\right) - \left(ax_i + b\right)\right] \cdot x_i = 0 \\ \frac{\partial E(a,b)}{\partial b} = 2 \sum\limits_{i=0}^{3} \left[f\left(x_i\right) - \left(ax_i + b\right)\right] = 0 \end{cases}$$

$$\begin{cases} \sum\limits_{i=0}^{3} \left[f\left(x_i\right) - \left(ax_i + b\right)\right] \cdot x_i = 0 \\ \sum\limits_{i=0}^{3} \left[f\left(x_i\right) - \left(ax_i + b\right)\right] = 0 \end{cases}$$

Polynomial least squares. In many experimental results the data are not linear or can be better estimated by a polynomial. Suppose that

$$\varphi(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0.$$

Consider $m+1$ points (x_i, y_i), $i = 0, \ldots, m$.

We have to find $a_i, i = 0, ..., n$, that minimize the sum

$$\begin{aligned} E(a_0, ..., a_n) &= \sum_{i=0}^{m} \left[f\left(x_i\right) - \varphi\left(x_i\right)\right]^2 \qquad (5) \\ &= \sum_{i=0}^{m} \left[f\left(x_i\right) - \sum_{k=0}^{n} a_k x_i^k\right]^2 . \end{aligned}$$

Denoting $y_i = f(x_i)$ we have

$$\begin{aligned}
E(a_0, ..., a_n) &= \sum_{i=0}^{m} y_i^2 - 2\sum_{i=0}^{m} y_i \varphi(x_i) + \sum_{i=0}^{m} (\varphi(x_i))^2 \\
&= \sum_{i=0}^{m} y_i^2 - 2\sum_{i=0}^{m} \left(\sum_{j=0}^{n} a_j x_i^j\right) y_i + \sum_{i=0}^{m} \left(\sum_{j=0}^{n} a_j x_i^j\right)^2 \\
&= \sum_{i=0}^{m} y_i^2 - 2\sum_{j=0}^{n} a_j \left(\sum_{i=0}^{m} x_i^j y_i\right) + \sum_{j=0}^{n}\sum_{k=0}^{n} a_j a_k \left(\sum_{i=0}^{m} x_i^{j+k}\right).
\end{aligned}$$

The minimum is obtained when

$$\frac{\partial E(a_0, ..., a_n)}{\partial a_j} = 0, \quad j = 0, ...n,$$

which are **the normal equations** and have a unique solution.

It is obtain

$$\frac{\partial E}{\partial a_j} = -2\sum_{i=0}^{m} x_i^j y_i + 2\sum_{k=0}^{n} a_k \left(\sum_{i=0}^{m} x_i^{j+k}\right) = 0$$

which gives $n+1$ unknowns a_j, $j \in \{0, 1, \ldots, n\}$ and $n+1$ equations

$$\sum_{i=0}^{m} x_i^j y_i = \sum_{k=0}^{n} a_k \left(\sum_{i=0}^{m} x_i^{j+k} \right), \quad \text{for each } j \in \{0, 1, \ldots, n\}. \tag{6}$$

We have the system

$$\begin{aligned}
&a_0 \sum_{i=0}^{m} x_i^0 + a_1 \sum_{i=0}^{m} x_i^1 + a_2 \sum_{i=0}^{m} x_i^2 + \cdots + a_n \sum_{i=0}^{m} x_i^n = \sum_{i=0}^{m} x_i^0 y_i, \\
&a_0 \sum_{i=0}^{m} x_i^1 + a_1 \sum_{i=0}^{m} x_i^2 + a_2 \sum_{i=0}^{m} x_i^3 + \cdots + a_n \sum_{i=0}^{m} x_i^{n+1} = \sum_{i=0}^{m} x_i^1 y_i \\
&\ldots \\
&a_0 \sum_{i=0}^{m} x_i^n + a_1 \sum_{i=0}^{m} x_i^{n+1} + a_2 \sum_{i=0}^{m} x_i^{n+2} + \cdots + a_n \sum_{i=0}^{m} x_i^{2n} = \sum_{i=0}^{m} x_i^n y_i
\end{aligned}$$

General case. Solution of the least squares problem is

$$\varphi(x) = \sum_{i=1}^{n} a_i g_i(x),$$

where $\{g_i, \ i = 1, ..., n\}$ is a basis of the space and the coefficients a_i are obtained solving **the normal equations**:

$$\sum_{i=1}^{n} a_i \langle g_i, g_k \rangle = \langle f, g_k \rangle \,, \qquad k = 1, ..., n.$$

In the discrete case

$$\langle f, g \rangle = \sum_{k=0}^{m} w\left(x_k\right) f\left(x_k\right) g\left(x_k\right)$$

and in the continuous case

$$\langle f, g \rangle = \int_a^b w\left(x\right) f\left(x\right) g\left(x\right),$$

where w is a weight function.

Example 13 *Fit the data in table*

x	-3	-1	2
$f(x)$	-4	-2	3

a) with the best least squares line;

b) with the best least squares polynomial of degree at most 2.

COURSE 6

3. Numerical integration of functions

The need: for evaluating definite integrals of functions that has no explicit antiderivatives or whose antiderivatives are not easy to obtain.

Let $f : [a, b] \to \mathbb{R}$ be an integrable function, x_k, $k = 0, ..., m$, distinct nodes from $[a, b]$.

Definition 1 *A formula of the form*

$$\int_a^b f(x)dx = \sum_{k=0}^{m} A_k f(x_k) + R(f),$$

is **a numerical integration formula** *or* **a quadrature formula**.

A_k - **the coefficients**; $x_k -$**the nodes**; $R(f)$ - **the remainder (the error)**.

Definition 2 Degree of exactness (degree of precision) *of a quadrature formula is r if and only if the error is zero for all the polynomials of degree $k = 0, 1, ..., r$, but is not zero for at least one polynomial of degree $r + 1$.*

From the linearity of R we have that the degree of exactness is r if and only if $R(e_i) = 0$, $i = 0, ..., r$ and $R(e_{r+1}) \neq 0$, where $e_i(x) = x^i$, $\forall i \in \mathbb{N}$.

3.1. Interpolatory quadrature formulas

Definition 3 *A quadrature formula*

$$\int_a^b f(x)dx = \sum_{k=0}^{m} A_k f(x_k) + R(f),$$

is **an interpolatory quadrature formula** *if it is obtained by integrating each member of an interpolation formula regarding the function f and the nodes x_k.*

Remark 4 *An interpolatory quadrature formula has its degree of exactness at least the degree of the corresponding interpolation polynomial.*

Consider Lagrange interpolation formula regarding the nodes $x_k \in [a,b]$, $k = 0, ..., m$:

$$f(x) = \sum_{k=0}^{m} \ell_k(x) f(x_k) + (R_m f)(x).$$

Integrating the two parts of this formula one obtains

$$\int_a^b f(x)dx = \sum_{k=0}^{m} A_k f(x_k) + R_m(f), \tag{1}$$

where

$$A_k = \int_a^b \ell_k(x)dx$$

and

$$R_m(f) = \int_a^b (R_m f)(x)dx. \tag{2}$$

If the nodes are equidistant, i.e., $x_k = a + kh$, $h = \frac{b-a}{m}$ then

$$A_k = (-1)^{m-k} \frac{h}{k!(m-k)!} \int_0^m \frac{t(t-1)...(t-m)}{(t-k)} dt, \;\; k = 0, ..., m. \qquad (3)$$

The remainder from the Lagrange interpolation formula can be written as:

$$(R_m f)(x) = \frac{u(x)}{(m+1)!} f^{(m+1)}(\xi(x)),$$

where $u(x) = \prod\limits_{k=0}^{m} (x - x_k)$, so the remainder of the quadrature formula may be written as

$$R_m(f) = \frac{1}{(m+1)!} \int_a^b u(x) f^{(m+1)}(\xi(x)) dx. \qquad (4)$$

Definition 5 *The quadrature formulas with equidistant nodes are called* **Newton-Cotes formulas.**

Consider the case $m = 1$ $(x_0 = a, x_1 = b, h = b - a)$.

Lagrange polynomial is

$$(L_1 f)(x) = \frac{x-b}{a-b} f(a) + \frac{x-a}{b-a} f(b)$$

and the remainder in interpolation formula is

$$(R_1 f)(x) = \frac{(x-a)(x-b)}{2} f''(\xi(x)).$$

Integrating the interpolation formula $f(x) = (L_1 f)(x) + (R_1 f)(x)$ one obtains

$$\begin{aligned}\int_a^b f(x)dx = &\int_a^b \left[\frac{x-b}{a-b} f(a) + \frac{x-a}{b-a} f(b)\right] dx \\ &+ \int_a^b \frac{(x-a)(x-b)}{2} f''(\xi(x)) dx.\end{aligned}$$

As $(x-a)(x-b)$ does not change the sign, by *Mean Value Th.* (If f:[a, b]→R is continuous and g is an integrable function that does not change sign on [a, b], then there exists c in (a, b) such that $\int_a^b f(x)g(x)dx = f(c)\int_a^b g(x)dx$), we have that there exist $\xi \in (a,b)$ such that

$$\int_a^b f(x)dx = \left[\frac{(x-b)^2}{2(a-b)}f(a) + \frac{(x-a)^2}{2(b-a)}f(b)\right]\Bigg|_a^b + \frac{f''(\xi)}{2}\left[\frac{x^3}{3} - \frac{(a+b)x^2}{2} + abx\right]\Bigg|_a^b$$

We obtain **the trapezium's quadrature formula**

$$\int_a^b f(x)dx = \frac{b-a}{2}[f(a)+f(b)] - \frac{(b-a)^3}{12}f''(\xi). \tag{5}$$

This formula is called the trapezium's formula because the integral is approximated by the area of a trapezium.

Remark 6 *The error from (5) involves f'', so the rule gives exact result when is applied to function whose second derivative is zero (polynomial of first degree or less). So its degree of exactness is 1.*

Example 7 *Approximate the integral $\int_1^3 (2x+1)dx$ using the trapezium's formula.*

(*Remark.* The result is the exact value of the integral because $f(x) = 2x + 1$ is a linear function and the degree of exactness of the trapezium's formula is 1.)

For $m = 2$ $((x_0 = a, x_1 = a + \frac{b-a}{2}, x_2 = b, h = \frac{b-a}{2})$ one obtains **the Simpson's quadrature formula**

$$\int_a^b f(x)dx = \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right] + R_2(f), \qquad (6)$$

where

$$R_2(f) = -\frac{(b-a)^5}{2880} f^{(4)}(\xi),\ a \le \xi \le b. \qquad (7)$$

Remark 8 *The error from (6) involves $f^{(4)}$, so the rule gives exact result when is applied to any polynomial of third degree or less. So degree of exactness of Simpson's formula is 3.*

Remark 9 *A Newton-Cotes quadrature formula has degree of exactness equal to* $\begin{cases} m, & \textit{if } m \textit{ is an odd number} \\ m+1, & \textit{if } m \textit{ is an even number.} \end{cases}$

Remark 10 *The coefficients of the Newton-Cotes quadrature formulas have the symmetry property:*

$$A_i = A_{m-i}, i = 0, ..., m.$$

Example 11 *Compare the trapezium's rule and Simpson's rule approximations for*

$$\int_0^2 x^2 dx.$$

Sol. *The exact value is* 2.667*; for trapezium rule the value is* 4*, for Simpson's rule the value is* 2.667*. (The approximation from Simpson's rule is exact because the error involves* $f^{(4)}(x) = 0$*.)*

COURSE 7

3.2. Repeated quadrature formulas

Example 1 *Approximate the integral using Simpson' s formula*

$$I = \int_0^4 e^x dx.$$

(The real value is $e^4 - 1 = 53.59$.*)*

Sol. We have $I \approx \frac{4}{6}\left[e^0 + 4e^2 + e^4\right] = 56.76.$

If we apply Simpson's formula twice we get

$$I \approx \int_0^2 e^x dx + \int_2^4 e^x dx \approx \frac{2}{6}\left[e^0 + 4e + e^2\right] + \frac{2}{6}\left[e^2 + 4e^3 + e^4\right] = 53.86$$

and if we apply four times we get

$$I \approx \sum_{i=0}^{3} \int_i^{i+1} e^x dx = 53.61,$$

so it follows the utility of using repeated formulas.

In practice, the problem of approximating $I = \int_a^b f(x)dx$ can be set in the following way: approximate the integral I with an absolute error not larger than a given bound ε.

By the trapezium's formula, for example, it follows that

$$|R_1(f)| = \frac{(b-a)^3}{12} |f''(\xi)| \geq \frac{(b-a)^3}{12} m_2 f$$

where $m_2 f = \min\limits_{a \leq x \leq b} |f''(x)|$. Therefore, if

$$\varepsilon < \frac{(b-a)^3}{12} m_2 f$$

then the problem cannot be solved by the trapezium's formula.

A solution: use formula with higher degree of exactness (e.g., the Simpson formula, etc.). But as m increases, the application of the formula becomes more difficult (computation, evaluation of the remainders (appear the derivatives of order $(m+1)$ or $(m+2)$ of f)).

An efficient way of constructing a practical quadrature formula: repeated application of a simple formula.

Let $x_k = a + kh,\ k = 0, ..., n$ with $h = \frac{b-a}{n}$, be the nodes of a uniform grid of $[a, b]$. By the additivity property of the integral we have

$$\int_a^b f(x)dx = \sum_{k=1}^{n} I_k, \text{ with } I_k = \int_{x_{k-1}}^{x_k} f(x)dx$$

Applying a quadrature formula to I_k, one obtains **the repeated quadrature formula.**

Applying to each integral I_k the trapezium's formula, we get

$$\int_a^b f(x)dx = \sum_{k=1}^{n} \left\{ \frac{x_k - x_{k-1}}{2} \left[f(x_{k-1}) + f(x_k) \right] - \frac{(x_k - x_{k-1})^3}{12} f"(\xi_k) \right\},$$

where $x_{k-1} \leq \xi_k \leq x_k$, or

$$\int_a^b f(x)dx = \frac{b-a}{2n} \left[f(a) + f(b) + 2 \sum_{k=1}^{n-1} f(x_k) \right] + R_n(f), \qquad (1)$$

with

$$R_n(f) = -\frac{(b-a)^3}{12n^3}\sum_{k=1}^{n} f''(\xi_k).$$

There exists $\xi \in (a,b)$ such that

$$\frac{1}{n}\sum_{k=1}^{n} f''(\xi_k) = f''(\xi).$$

So **the repeated trapezium's quadrature formula** is

$$\int_a^b f(x)dx = \frac{b-a}{2n}\left[f(a)+f(b)+2\sum_{k=1}^{n-1} f(x_k)\right] + R_n(f), \qquad (2)$$

with

$$R_n(f) = -\frac{(b-a)^3}{12n^2}f''(\xi),\ \ a<\xi<b \qquad (3)$$

We have

$$|R_n(f)| \leq \frac{(b-a)^3}{12n^2}M_2 f,$$

where $M_2f = \max\limits_{a\leq x\leq b} |f''(x)|$. By

$$|R_n(f)| \leq \frac{(b-a)^3}{12n^2} M_2f, \tag{4}$$

it follows that the repeated trapezium quadrature formula allows the approx. of an integral with arbitrary small given error, if n is taken sufficiently large. If we want that the absolute error to be smaller than ε, we determine the smallest solution n of the inequation

$$\frac{(b-a)^3}{12n^2} M_2f < \varepsilon,\ n \in \mathbb{N},$$

and using this value in (1), leads to desired approximation.

Similarly, there is obtained **the repeated Simpson's quadrature formula**

$$\int_a^b f(x)dx = \tfrac{b-a}{6n}\left[f(a)+f(b)+4\sum_{k=1}^{n} f\left(\tfrac{x_{k-1}+x_k}{2}\right)+2\sum_{k=1}^{n-1} f(x_k)\right]+R_n(f) \tag{5}$$

where

$$R_n(f) = -\frac{(b-a)^5}{2880n^4}f^{(4)}(\xi),\ a<\xi<b,$$

and

$$|R_n(f)| \leq \frac{(b-a)^5}{2880n^4}M_4f.$$

Example 2 *Approximate the integral $\int_1^3 (2x+1)dx$ with repeated trapezium's formula for $n=2$.*

(*Remark.* The result is the exact value of the integral because $f(x)=2x+1$ is a linear function and the degree of exactness of the trapezium's formula is 1.)

Example 3 *Approximate $\frac{\pi}{4}$ with repeated trapezium's formula, considering precision $\varepsilon = 10^{-2}$.*

Sol. We have

$$\frac{\pi}{4} = arctg(1) = \int_0^1 \frac{dx}{1+x^2},$$

so $f(x) = \frac{1}{1+x^2}$. Using (4), we get

$$|R_n(f)| \leq \frac{(1-0)^3}{12n^2} M_2 f.$$

We have

$$f'(x) = \frac{-2x}{(1+x^2)^2}$$
$$f''(x) = \frac{6x^2-2}{(1+x^2)^3}$$

and

$$M_2 f = \max_{x \in [0,1]} |f''(x)| = 2,$$

so

$$|R_n(f)| \leq \frac{1}{6n^2} < 10^{-2} \Rightarrow n^2 > \frac{10^2}{6} = 16.66 \Rightarrow n = 5.$$

We have $x_0 = 0, x_1 = \frac{1}{5}, x_2 = \frac{2}{5}, x_3 = \frac{3}{5}, x_4 = \frac{4}{5}, x_5 = 1$ ($h = \frac{1}{5}$). The integral will be

$$\int_a^b f(x)dx \approx \frac{1}{10}\left\{f(0) + f(1) + 2\left[f(\frac{1}{5}) + f(\frac{2}{5}) + f(\frac{3}{5}) + f(\frac{4}{5})\right]\right\} = 0.7837.$$

(The real value is 0.7854.)

Example 4 *Approximate*

$$\ln 2 = \int_0^1 \frac{1}{1+x}dx,$$

with precision $\varepsilon = 10^{-3}$, *using the repeated Simpson's formula.*

3.3. The Romberg's iterative generation method of a repeated quadrature formula

The presence of derivatives in the remainder $\Rightarrow$ difficulties in applicability to practical problems and to computer programs. There are preferred, in this sense, the iterative quadratures.

Consider the iterative generation method of a repeated formula by *the Romberg's method.*

In the case of the trapezium formula we have

$$Q_{T_0}(f) = \frac{h}{2}\left[f(a) + f(b)\right], \ h = b - a,$$

$Q_{T_0}(f)$ being the first element of the sequence.

We divide the interval $[a, b]$ in two equal parts, of length $\frac{h}{2}$ and applying to each subinterval $[a, a + \frac{h}{2}]$ and $[a + \frac{h}{2}, b]$ the trapezium formula we get

$$Q_{T_1}(f) = \frac{h}{4}\left[f(a) + 2f\left(a + \frac{h}{2}\right) + f(b)\right]$$

or

$$Q_{T_1}(f) = \frac{1}{2} Q_{T_0}(f) + hf\left(a + \frac{h}{2}\right).$$

Dividing now each previous divisions $[a, \frac{a+b}{2}]$ and $[\frac{a+b}{2}, b]$ in two equal parts, we obtain a division of the initial interval in $4 = 2^2$ equal parts, each of length $\frac{h}{4}$. Applying the repeated trapezium formula, we get

$$\begin{aligned} Q_{T_2}(f) &= \frac{h}{8}\left[f(a) + 2\sum_{i=1}^{3} f\left(a + \frac{ih}{4}\right) + f(b)\right] \qquad (6) \\ &= \frac{1}{2} Q_{T_1}(f) + \frac{h}{2^2}\left[f\left(a + \frac{1}{2^2}h\right) + f\left(a + \frac{3}{2^2}h\right)\right]. \end{aligned}$$

Continuing in an analogous manner, we get

$$Q_{T_k}(f) = \frac{1}{2} Q_{T_{k-1}}(f) + \frac{h}{2^k} \sum_{j=1}^{2^{k-1}} f\left(a + \frac{2j-1}{2^k}h\right),\ k = 1, 2, ... \qquad (7)$$

We obtain the sequence

$$Q_{T_0}(f),\ Q_{T_1}(f), ..., Q_{T_k}(f), ... \qquad (8)$$

which converges to the value $I = \int_a^b f(x)dx$.

We approximate the error by $\left|Q_{T_n}(f) - Q_{T_{n-1}}(f)\right|$. If we want to approximate I with error less than ε, we compute successively the elements of (8) until the first index for which

$$\left|Q_{T_n}(f) - Q_{T_{n-1}}(f)\right| \leq \varepsilon,$$

$Q_{T_n}(f)$ being the required value.

Similarly, one may iteratively generate the repeated Simpson's formula. Denoting by $Q_{S_k}(f)$ the Simpson's formula repeated k times, we have

$$Q_{S_k}(f) = \frac{1}{3}\left[4Q_{T_{k+1}}(f) - Q_{T_k}(f)\right],\ k = 0, 1, ...$$

where

$$Q_{S_0}(f) = \frac{h}{6}\left[f(a) + 4f\left(a + \frac{h}{2}\right) + f(b)\right]$$

is the Simpson's quadrature formula.

Another **Romberg's algorithm**, based on Aitken scheme:

$$\begin{array}{llll} T_{00} & & & \\ T_{10} & T_{11} & & \\ \dots & & & \\ T_{i0} & T_{i1} & \dots & T_{ii} \end{array} \tag{9}$$

where the first column is computed by repeated trapezium rule and the other rows are computed by

$$T_{i,j} = \frac{4^{-j} T_{i-1,j-1} - T_{i,j-1}}{4^{-j} - 1}. \tag{10}$$

The columns, rows and diagonal all converge to the value of the integral; for smooth functions, the diagonal converges fastest.

The Romberg scheme computed using formula (10) contains in its first column the values of the repeated trapezium rule and in its second column the values of the Simpson's rule.

If we want to approximate I with error less than ε, we compute successively the lines of (9) until

$$\left|T_{i,i} - T_{i-1,i-1}\right| \leq \varepsilon,$$

$T_{i,i}$ being the required value.

3.4. Adaptive quadrature methods

The repeated integration methods require equidistant nodes. There are problems where the function contains both regions with large variations and with small variations. It is needed a smaller step for the regions with large variations than for the regions with small variations in order that the error to be uniformly distributed.

Such methods, which adapt the size of the step in accordance with the need, are called **adaptive quadrature methods**.

We present the method based on the repeated Simpson's quadrature formula.

Suppose we want to approximate with precision ε the integral

$$I = \int_a^b f(x)\,dx.$$

First step: we apply the Simpson's formula with the step $h = \frac{b-a}{2}$ $(x_0 = a,\, x_1 = a + h,\, x_2 = b)$:

$$\begin{gathered} \textstyle\int_a^b f(x)\,dx = \frac{b-a}{6}\Big(f(a) + 4f(a+\frac{h}{2}) + f(b)\Big) - \frac{(b-a)^5}{2880} f^{(4)}(\xi) = \\ := S(a,b) - \frac{h^5}{90} f^{(4)}(\xi) \end{gathered} \tag{11}$$

Then, we apply the Simpson's formula with the step $\frac{(b-a)}{4} = \frac{h}{2}$:

$$\begin{gathered} \textstyle\int_a^b f(x)\,dx = \frac{h}{6}[f(a) + 4f(a+\frac{h}{2}) + 2f(a+h) + 4f(a+\frac{3h}{2}) \\ +f(b)] - \frac{h^5}{2^4 \cdot 90} f^{(4)}(\theta) \end{gathered}$$

$$\int_a^b f(x)\,dx = S\left(a, \tfrac{a+b}{2}\right) + S\left(\tfrac{a+b}{2}, b\right) - \tfrac{1}{16}\tfrac{h^5}{90} f^{(4)}(\theta) \tag{12}$$

We estimate the error without determining $f^{(4)}(\xi)$. We suppose $f^{(4)}(\xi) \simeq f^{(4)}(\theta)$. We get

$$S(a,b) - \tfrac{h^5}{90} f^{(4)}(\xi) = S\left(a, \tfrac{a+b}{2}\right) + S\left(\tfrac{a+b}{2}, b\right) - \tfrac{1}{16}\tfrac{h^5}{90} f^{(4)}(\xi),$$

whence

$$S\left(a,b\right)-S\left(a,\tfrac{a+b}{2}\right)-S\left(\tfrac{a+b}{2},b\right)=\tfrac{15}{16}\tfrac{h^5}{90}f^{(4)}\left(\xi\right),$$

or

$$\tfrac{h^5}{90}f^{(4)}\left(\xi\right)=\tfrac{16}{15}\left(S\left(a,b\right)-S\left(a,\tfrac{a+b}{2}\right)-S\left(\tfrac{a+b}{2},b\right)\right)$$

Replacing in (12) we get

$$\left|\int_a^b f\left(x\right)dx-S\left(a,\tfrac{a+b}{2}\right)-S\left(\tfrac{a+b}{2},b\right)\right|=\tfrac{1}{16}\tfrac{h^5}{90}f^{(4)}\left(\theta\right)$$
$$=\tfrac{1}{15}\left|S\left(a,b\right)-S\left(a,\tfrac{a+b}{2}\right)-S\left(\tfrac{a+b}{2},b\right)\right|$$

So the remainder of the approximation of I by $S\left(a,\tfrac{a+b}{2}\right)+S\left(\tfrac{a+b}{2},b\right)$ is 15 times smaller than the expression $\left|S\left(a,b\right)-S\left(a,\tfrac{a+b}{2}\right)-S\left(\tfrac{a+b}{2},b\right)\right|$. Hence, if

$$\left|S\left(a,b\right)-S\left(a,\tfrac{a+b}{2}\right)-S\left(\tfrac{a+b}{2},b\right)\right|<15\varepsilon,\text{ then}\tag{13}$$

$$\left|I-S\left(a,\tfrac{a+b}{2}\right)-S\left(\tfrac{a+b}{2},b\right)\right|<\varepsilon.$$

When (13) does not hold, the procedure is applied individually on $[a, (a + b)/2]$ and $[(a + b)/2, b]$ in order to determine if the approx. of the integral on each two subintervals is performed with error $\varepsilon/2$. If yes, the sum of these two approx. offer an approx. of I with precision ε. If on a subinterval it is not obtained the error $\varepsilon/2$, then we divide that subinterval and we analyze if the approx. on the resulted two subintervals has precision $\varepsilon/4$, and so on. This procedure of halfing is continued until the corresponding error is attained on each subinterval.

Algorithm: (the idea: ”divide and conquer”)

```
function I=adquad(a,b,er)
      I1=Simpson(a,b)
       I2=Simpson(a,$\frac{a+b}{2}$)+Simpson($\frac{a+b}{2}$,b)
   if |I1-I2|<15*er
          I=I2
          return
   else
         I=adquad(a,$\frac{a+b}{2}$,$\frac{er}{2}$)+adquad($\frac{a+b}{2}$,b,$\frac{er}{2}$)
   end
```

Remark 5 *For example, for evaluating the integral* $\int_1^3 \frac{100}{x^2} \sin \frac{10}{x} dx$ *with* $\varepsilon = 10^{-4}$, *repeated Simpson formula requires* 177 *function evaluations, nearly twice as many as adaptive quadrature.*

COURSE 8

3.5. General quadrature formulas

Using the interpolation formulas, there are obtained a large variety of quadrature formulas.

In the case of some concrete applications, the choosing of the quadrature formula is made according to the information about the function f.

A general quadrature formula is given by:

$$\int_a^b f(x)dx = \sum_{k=0}^{m} \sum_{j\in I_k} A_{kj} f^{(j)}(x_k) + R(f).$$

For example, consider the Hermite interpolation formula for $f : [a,b] \to \mathbb{R}$, the nodes $x_k \in [a,b]$, $k = 0, ..., m$ multiple of orders $r_0, ..., r_m \in \mathbb{N}$,

$$f = H_n f + R_n f, \tag{1}$$

with $\dot{n} = m + r_0 + ... + r_m$, and

$$(H_n f)(x) = \sum_{k=0}^{m} \sum_{j=0}^{r_k} h_{kj} f^{(j)}(x_k),$$

h_{jk} being the Hermite fundamental polynomials.

Formula (1) generates the quadrature formula with multiple nodes:

$$\int_a^b f(x)dx = \sum_{k=0}^{m} \sum_{j=0}^{r_k} A_{kj} f^{(j)}(x_k) + R_n(f),$$

with

$$A_{kj} = \int_a^b h_{kj}(x)dx.$$

Similarly, there are obtained quadrature formulas starting to Birkhoff interpolation formulas.

For example, if we know only the values of $f'(a)$ and $f(b)$ which is the corresponding quadrature formula?

Sol. We have $f(x) = (B_1 f)(x) + (R_1 f)(x)$ with

$$(B_1 f)(x) = b_{01}(x) f'(a) + b_{10}(x) f(b). \tag{2}$$

Formula (2) generates the quadrature formula

$$\int_a^b f(x)dx = \sum_{k=0}^{1} \sum_{j=I_k} A_{kj} f^{(j)}(x_k) + R_1(f),$$

with

$$A_{01} = \int_a^b b_{01}(x)dx = -\frac{(b-a)^2}{2}$$
$$A_{10} = \int_a^b b_{10}(x)dx = (b-a).$$

Example 1 *Find the coefficients A, B and C of the following quadrature formula:*

$$\int_0^1 f(x)dx = Af(0) + Bf'(0) + Cf(1) + R(f).$$

3.6. Quadrature formulas of Gauss type

All the previous rules can be written in the form

$$\int_a^b f(x)dx = \sum_{k=1}^{m} A_k f(x_k) + R_m(f), \tag{3}$$

where the coefficients A_k, $k = 1, ..., m$, do not depend on the function f. We have picked the nodes x_k, $k = 1, ..., m$ equispaced and have then calculated the coefficients A_k, $k = 1, ..., m$. This guarantees that the rule is exact for polynomials of degree $\leq m$.

It is possible to make such a rule exact for polynomials of degree $\leq 2m - 1$, by choosing also the nodes appropriately. This is the basic idea of the gaussian rules.

Let $f : [a, b] \to \mathbb{R}$ be an integrable function and $w : [a, b] \to \mathbb{R}_+$ a weight function, integrable on $[a, b]$.

Definition 2 *A formula of the following form*

$$\int_a^b w(x)f(x)dx = \sum_{k=1}^{m} A_k f(x_k) + R_m(f) \tag{4}$$

is called **a quadrature formula of Gauss type** *or* **with maximum degree of exactness** *if the coefficients A_k and the nodes x_k, $k = 1, ..., m$ are determined such that the formula has the maximum degree of exactness.*

Remark 3 *The coefficients and the nodes are determined such that to minimize the error, to produce exact results for the largest class of polynomials.*

A_k and x_k, $k = 1, ..., m$ from (4) are $2m$ unknown parameters $\Rightarrow$ $2m$ equations obtained such that the formula (4) is exact for any polynomial degree at most $2m - 1$.

It is often possible to rewrite the integral $\int_a^b g(x)dx$ as $\int_a^b w(x)f(x)dx$, where $w(x)$ is a nonnegative integrable function, and $f(x) = \frac{g(x)}{w(x)}$ is smooth, or it is possible to consider the simple choice $w(x) = 1$.

For the general case, consider the elementary polynomials $e_k(x) = x^k$; $k = 0, ..., 2m-1$ and obtain the system s.t. $R_m(e_k) = 0$:

$$\begin{cases} \sum\limits_{k=1}^{m} A_k e_0(x_k) = \int_a^b w(x) e_0\,(x)\,dx \\ \sum\limits_{k=1}^{m} A_k e_1(x_k) = \int_a^b w(x) e_1\,(x)\,dx \\ ... \\ \sum\limits_{k=1}^{m} A_k e_{2m-1}(x_k) = \int_a^b w(x) e_{2m-1}\,(x)\,dx \end{cases}$$

$\Longleftrightarrow$

$$\begin{cases} A_1 + A_2 + ... + A_m = \mu_0 \\ A_1 x_1 + A_2 x_2 + ... + A_m x_m = \mu_1 \\ ... \\ A_1 x_1^{2m-1} + A_2 x_2^{2m-1} + ... + A_m x_m^{2m-1} = \mu_{2m-1} \end{cases} \tag{5}$$

with

$$\mu_k = \int_a^b w(x) x^k dx.$$

As, the system (5) is difficult to solve, there have been found other ways to find the unknown parameters.

If $w(x) = 1$ (this case was studied by Gauss), then the nodes are the roots of Legendre orthogonal polynomial

$$u(x) = \frac{m!}{(2m)!}\left[(x-a)^m(x-b)^m\right]^{(m)}$$

and for finding the coefficients we use the first m equations from the system (5).

Example 4 *Consider $m = 1$ and obtain the following Gauss type quadrature formula*

$$\int_a^b f(x)dx = A_1 f(x_1) + R_1(f).$$

The system (5) becomes

$$\begin{cases} A_1 = \int_a^b dx = b-a \\ A_1 x_1 = \int_a^b x dx = \frac{b^2-a^2}{2}. \end{cases}$$

The unique solution of this system is $A_1 = b-a$, $x_1 = \frac{a+b}{2}$.

The same result is obtained considering x_1 the root of the Legendre polynomial of the first degree,

$$u(x) = \frac{1}{2}[(x-a)(x-b)]' = x - \frac{a+b}{2}.$$

The Gauss type quadrature formula with one node is

$$\int_a^b f(x)dx = (b-a)f\left(\frac{a+b}{2}\right) + R_1(f),$$

with

$$R_1(f) = \frac{(b-a)^3}{24}f''(\xi), \qquad \xi \in [a,b]$$

which is called **the rectangle quadrature rule** (also called **the midpoint rule).**

The repeated rectangle (midpoint) quadrature formula is

$$\int_a^b f(x)dx = \frac{b-a}{n}\sum_{i=1}^{n} f(x_i) + R_n(f),$$
$$R_n(f) = \frac{(b-a)^3}{24n^2} f''(\xi), \qquad \xi \in [a,b]$$

with $x_1 = a + \frac{b-a}{2n}$, $x_i = x_1 + (i-1)\frac{b-a}{n}$, $i = 2, ..., n$.

We have

$$|R_n(f)| \ \leq \frac{(b-a)^3}{24n^2} M_2 f, \quad \text{with } M_2 f = \max_{x\in[a,b]} |f''(x)|.$$

Remark 5 *Another* **rectangle rule** *is the following*:

$$\int_a^b f(x)dx = (b-a)f\,(a) + R(f),$$

with

$$R(f) = \frac{(b-a)^2}{2} f'(\xi), \qquad \xi \in [a,b].$$

Romberg's algorithm for the rectangle (midpoint) quadrature formula. Applying successively the rectangle formula on $[a, b]$, we get

$$Q_{D_0}(f) = (b-a)f(x_1), \quad x_1 = \frac{a+b}{2}$$

$$Q_{D_1}(f) = \frac{1}{3}Q_{D_0}(f) + \frac{b-a}{3}\left[f(x_2) + f\left(x_3\right)\right],$$

$$x_2 = a + \frac{b-a}{6},\ x_3 = b - \frac{b-a}{6}.$$

Continuing in an analogous manner, we obtain the sequence

$$Q_{D_0}(f),\ Q_{D_1}(f), ..., Q_{D_k}(f), ... \tag{6}$$

which converges to the value I of the integral $\int_a^b f(x)dx$.

If we want to approximate the integral I with error less than ε, we compute successively the elements of (6) until the first index for which

$$\left|Q_{D_m}(f) - Q_{D_{m-1}}(f)\right| \leq \varepsilon,$$

$Q_{D_m}(f)$ being the required value.

Example 6 *Approximate* $\ln 2 = \int_1^2 \frac{1}{x}dx$, *with* $\varepsilon = 10^{-2}$, *using the repeated rectangle (midpoint) method.*

Solution. We have

$$\int_a^b f(x)dx = \frac{b-a}{n}\sum_{i=1}^{n} f(x_i) + R_n(f),$$

$$R_n(f) = \frac{(b-a)^3}{24n^2}f''(\xi), \qquad \xi \in [a,b].$$

$$\ln 2 = \int_1^2 \frac{dx}{x},$$

so $f(x) = \frac{1}{x}$ and we get

$$\ln 2 = \frac{b-a}{n}\left[f(a+\frac{b-a}{2n}) + \sum_{i=2}^{n} f(a+\frac{b-a}{2n}+(i-1)\frac{b-a}{n})\right] + \frac{(b-a)^3}{24n^2}f''(\xi)$$

We have $f(x) = \frac{1}{x}$, $f'(x) = -\frac{1}{x^2}$, $f''(x) = \frac{2}{x^3}$, and $|f''(\xi)| \leq 2$, for $\xi \in [1,2]$ so it follows

$$|R_n(f)| \leq \frac{1}{24n^2}2 < 10^{-2} \Rightarrow 12n^2 > 100 \Rightarrow n = 3.$$

Therefore,

$$\ln 2 \approx \frac{1}{3}\left(\frac{1}{1+\frac{1}{6}}+\frac{1}{1+\frac{1}{6}+\frac{1}{3}}+\frac{1}{1+\frac{1}{6}+\frac{2}{3}}\right)=\frac{1}{3}\left(\frac{6}{7}+\frac{6}{9}+\frac{6}{11}\right)=0.6897$$

(real value is 0.693...)

Example 7 *For* $m = 2$, *Gauss quadrature formula is*

$$\int_a^b f(x)dx = A_1 f(x_1) + A_2 f(x_2) + R_2(f).$$

Find A_1, A_2, x_1, x_2.

Sol. The corresponding Legendre polynomial is

$$\begin{aligned} u(x) &= \frac{2}{4!}\left[(x-a)^2(x-b)^2\right]'' \\ &= x^2 - (a+b)x + \frac{1}{6}(a^2+b^2+4ab), \end{aligned}$$

with the roots

$$x_1 = \frac{a+b}{2} - \frac{(b-a)\sqrt{3}}{6},$$
$$x_2 = \frac{a+b}{2} + \frac{(b-a)\sqrt{3}}{6}.$$

For finding A_1 and A_2 we use the first two equations:

$$\begin{cases} A_1 + A_2 = b - a \\ A_1 x_1 + A_2 x_2 = (b^2 - a^2)/2. \end{cases}$$

We get

$$A_1 = A_2 = (b-a)/2,$$

so the quadrature formula of Gauss type with two nodes is

$$\int_a^b f(x)dx = \frac{b-a}{2}\left[f\left(\frac{a+b}{2} - \frac{b-a}{6}\sqrt{3}\right) + f\left(\frac{a+b}{2} + \frac{b-a}{6}\sqrt{3}\right)\right] + R_2(f).$$

For the interval $[-1,1]$ we get $A_1 = A_2 = 1$ and $x_1 = -\frac{\sqrt{3}}{3}, x_2 = \frac{\sqrt{3}}{3}$, which gives the fomula

$$\int_{-1}^{1} f(x)dx \simeq f(-\frac{\sqrt{3}}{3}) + f(\frac{\sqrt{3}}{3}).$$

This formula has degree of precision 3, i.e., it gives exact result for every polynomial of the 3−rd degree or less.

Remark 8 *The resulting rules look more complicated than the interpolatory rules. Both nodes and weights for gaussian rules are, in general, irrational numbers. But, on a computer, it usually makes no difference whether one evaluates a function at $x = 3$ or at $x = 1/\sqrt{3}$. Once the nodes and weights of such a rule are stored, these rules are as easily used as the trapezium rule or Simpson's rule. At the same time, these gaussian rules are usually much more accurate when compared with the last ones on the basis of number of function values used.*

Remark 9 *a) The coefficients A_k, $k = 1, ..., m$ of a Gauss type formula are positive.*

b) The coefficients A_k, $k = 1, ..., m$ and the roots of the Legendre polynomials can be found in tables, for $a = -1$, $b = 1$. For example, for $m = 2$ and $m = 3$:

m	*nodes*	*coefficients*
2	0.577	1
	−0.577	1
3	0.774	0.555
	0	0.888
	−0.774	0.555

c) For different weight functions, tables are available for both the nodes and the coefficients.

Example 10 *Approximate $\int_{-1}^{1} e^x \cos x dx$ using a Gauss type quadrature formula for $m = 3$.*

Sol.

$$\begin{aligned}\int_{-1}^{1} e^x \cos x dx &\simeq 0.55e^{0.77} \cos 0.77 + 0.88 \cos 0 + 0.55e^{-0.77} \cos(-0.77) \\ &= 1.9333904\end{aligned}$$

(Exact value is 1.9334214.) Absolute error is $<3.2 \cdot 10^{-5}$.

The integral $\int_a^b f(x)dx$ for an arbitrary interval $[a, b]$ could be transfomed in an integral on $[-1, 1]$ using the change of variable

$$t = \frac{2x - a - b}{b - a} \Leftrightarrow x = \frac{1}{2}\left[(b - a)t + a + b\right].$$

The Gauss type quadrature formulas may be applied on the following way:

$$\int_a^b f(x)dx = \int_{-1}^1 f\left(\frac{(b-a)t + (b+a)}{2}\right)\frac{(b-a)}{2}dt. \tag{7}$$

Example 11 *Consider the integral* $\int_1^3 (x^6 - x^2 \sin(2x))dx = 317.3442466$.

a) Compare the result obtained for Newton-Cotes type formula with $m = 1$ *(trapezium formula) and Gauss-Legendre formula for* $m = 2$;

b) Compare the result obtained for Newton-Cotes type formula with $m = 2$ *(Simpson formula) and Gauss-Legendre formula for* $m = 3$.

Sol. a) Each formula needs 2 evaluations of the function $f(x) = x^6 - x^2 \sin(2x)$. We have

$$\text{Trapezium formula } (m=1): \frac{2}{2}\left[f(1)+f(3)\right] = 731.6054420;$$

and a Gauss type formula for $m = 2$, using (7):

$$\begin{aligned}\int_1^3 (x^6 - x^2\sin(2x))dx &= \int_{-1}^1 ((t+2)^6 - (t+2)^2 \sin(2(t+2)))dt \\ &\simeq f(-0.577+2) + f(0.577+2) = 306.8199344.\end{aligned}$$

b) Each formula needs 3 evaluations of the function. We have

$$\text{F. Simpson } (m=2): \frac{1}{3}\left[f(1)+4f(2)+f(3)\right] = 333.23;$$

and a Gauss type formula for $m = 3$, using (7):

$$\begin{aligned}\int_1^3 (x^6 - x^2\sin(2x))dx &= \int_{-1}^1 ((t+2)^6 - (t+2)^2 \sin(2(t+2)))dt \\ &\simeq 0.55f(-0.77+2) + 0.88f(2) + 0.55f(0.77+2) \\ &= 317.2641516\end{aligned}$$

COURSE 9

4. Numerical methods for solving linear systems

Practical solving of many problems eventually leads to solving linear systems.

Real-World Application:

1. Price of Fruits: Peter buys two apples and three bananas for $4. Nadia buys four apples and six bananas for $8 from the same store. How much does one banana and one apple costs?

2. A baker sells plain cakes for $7 and decorated cakes for $11. On a busy Saturday the baker started with 120 cakes, and sold all but three. His takings for the day were $991. How many plain cakes did he sell that day, and how many were decorated before they were sold?

3. Twice John's age plus five times Claire's age is 204. Nine times John's age minus three times Claire's age is also 204. How old are John and Claire?

4. Assume an electric network consisting of two voltage sources and three resistors. Applying Kirchhoff's laws it is determined the current going through each resistor.

5. Network analysis: networks composed of branches and junctions are used as models in such fields as economics, traffic analysis, and electrical engineering.

Classification of the methods:

- *direct methods - with low number of unknowns* (up to several tens of thousands); they provide the exact solution of the system in a finite number of steps.

- *iterative methods - with medium number of unknowns*; it is obtained an approximation of the solution as the limit of a sequence.

- *semiiterative methods - with large number of unknowns*; it is obtained an approximation of the solution.

4.1. Perturbation of linear systems.

Consider the linear system

$$Ax = b.$$

Definition 1 *The number* $\text{cond}\,(A) = \|A\| \left\|A^{-1}\right\|$ *is called* **conditioning number** *of the matrix A. It measures the sensibility of the solution x of the system $Ax = b$ to the perturbation of A and b.*

The system is **good conditioned** *if* $\text{cond}\,(A)$ *is small (<1000) or it is* **ill conditioned** *if* $\text{cond}\,(A)$ *is great.*

Remark 2 *1.* $\text{cond}\,(A) \geq 1$*.*

2. $\text{cond}(A)$ *depends on the norm used.*

Consider an example

$$\begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 32 \\ 23 \\ 33 \\ 31 \end{pmatrix},$$

with the solution $\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$.

We perturbate the right hand side:

$$\begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix} \begin{pmatrix} x_1 + \delta x_1 \\ x_2 + \delta x_2 \\ x_3 + \delta x_3 \\ x_4 + \delta x_4 \end{pmatrix} = \begin{pmatrix} 32.1 \\ 22.9 \\ 33.1 \\ 30.9 \end{pmatrix},$$

and obtain the exact solution $\begin{pmatrix} 9.2 \\ -12.6 \\ 4.5 \\ -1.1 \end{pmatrix}$.

Consider, for example,

$$\left|\frac{b_2-(b_2+\delta b_2)}{b_2}\right| = \left|\frac{\delta b_2}{b_2}\right| = \frac{1}{229} \approx \frac{1}{200},$$

where δb_i, $i = \overline{1,3}$ denote the perturbations of b, and

$$\left|\frac{x_2-(x_2+\delta x_2)}{x_2}\right| = \left|\frac{\delta x_2}{x_2}\right| = 13.6 \approx 10.$$

Thus, a relative error of order $\frac{1}{200}$ on the right hand side (precision of $\frac{1}{200}$ for the data in a linear system) attracts a relative error of order 10 on the solution, 2000 times larger.

Consider the same system, and perturb the matrix A:

$$\begin{pmatrix} 10 & 7 & 8.1 & 7.2 \\ 7.08 & 5.04 & 6 & 5 \\ 8 & 5.98 & 9.89 & 9 \\ 6.99 & 4.99 & 9 & 9.98 \end{pmatrix} \begin{pmatrix} x_1+\delta x_1 \\ x_2+\delta x_2 \\ x_3+\delta x_3 \\ x_4+\delta x_4 \end{pmatrix} = \begin{pmatrix} 32 \\ 23 \\ 33 \\ 31 \end{pmatrix},$$

with exact solution $\begin{pmatrix} -81 \\ 137 \\ -34 \\ 22 \end{pmatrix}$.

The matrix A seems to have good properties (symmetric, with determinant 1), and the inverse $A^{-1} = \begin{pmatrix} 25 & -41 & 10 & -6 \\ -41 & 68 & -17 & 10 \\ 10 & -17 & 5 & -3 \\ -6 & 10 & -3 & 2 \end{pmatrix}$ is also with integer numbers.

This example is very concerning as such orders of the errors in many experimental sciences are considered as satisfactory.

Remark 3 *For this example we have* cond$\,(A) = 2984$ *(in euclidian norm).*

Analyze the phenomenon:

⧫ In the first case, when b is perturbed, we compare de exact solutions x and $x + \delta x$ of the systems

$$Ax = b$$

and

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

Let $\|\cdot\|$ be a norm on $\mathbb{R}^n$ and $\|\cdot\|$ the induced matrix norm.

We have the systems

$$Ax = b$$

and

$$Ax + A\delta x = b + \delta b \Longleftrightarrow A\delta x = \delta b.$$

From $\delta x = A^{-1}\delta b$ we get $\|\delta x\| \leq \left\|A^{-1}\right\| \|\delta b\|$

and from $b = Ax$ we get $\|b\| \leq \|A\| \, \|x\| \Leftrightarrow \frac{1}{\|x\|} \leq \frac{\|A\|}{\|b\|}$,

so the relative error of the result is bounded by

$$\frac{\|\delta x\|}{\|x\|} \leq \left(\|A\| \left\|A^{-1}\right\|\right) \frac{\|\delta b\|}{\|b\|} \overset{denoted}{=} \text{cond}\,(A)\, \frac{\|\delta b\|}{\|b\|}. \qquad (1)$$

⧫ In the second case, when the matrix A is perturbed, we compare the exact solutions of the linear systems

$$Ax = b$$

and

$$\begin{aligned}(A + \delta A)(x + \delta x) = b &\Longleftrightarrow Ax + A\delta x + \delta A x + \delta A \delta x = b \\ &\Longleftrightarrow A\delta x = -\delta A(x + \delta x).\end{aligned}$$

From $\delta x = -A^{-1}\delta A(x + \delta x)$, we get $\|\delta x\| \leq \left\|A^{-1}\right\| \|\delta A\| \, \|x + \delta x\|$, or

$$\frac{\|\delta x\|}{\|x + \delta x\|} \leq \left\|A^{-1}\right\| \|\delta A\| = \left(\|A\| \left\|A^{-1}\right\|\right) \frac{\|\delta A\|}{\|A\|} = \text{cond}\,(A)\, \frac{\|\delta A\|}{\|A\|}. \qquad (2)$$

4.2. Direct methods for solving linear systems

Why Cramer's method is not suitable for solving linear systems for $n \geq 100$ and it will not be in near future?

For applying Cramer's method for a $n \times n$ system we need in a rough evaluation the following number of operations:

$$\begin{cases} (n+1)! & \text{aditions} \\ (n+2)! & \text{multiplications} \\ n & \text{divisions} \end{cases}$$

Consider, hypothetically, a volume $V = 1$ km^3 of cubic processors of each having the side $l = 10^{-8}$ cm (radius of an atom), the time for execution of an operation is supposed to be equal to the time needed for the light to pass through an atom. (Light speed is 300.000 km/s.)

In this hypothetically case, the time necessary for solving the $n \times n$ system, $n \geq 100$, will be more than 10^{94} years!

4.2.1. Gauss method for solving linear systems

Consider the linear system $Ax = b$, i.e.,

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}. \tag{3}$$

The method consists of two stages:

- reducing the system (3) to an equivalent one, $Ux = d$, with U an upper triangular matrix.

- solving of the upper triangular linear system $Ux = d$ by backward substitution.

At least one of the elements on the first column is nonzero, otherwise A is singular. We choose one of these nonzero elements (using some criterion) and this will be called the first elimination **pivot.**

If the case, we change the line of the pivot with the first line, both in A and in b, and next we successively make zeros under the first pivot:

$$\begin{pmatrix} a_{11}^1 & a_{12}^1 & \dots & a_{1n}^1 \\ 0 & a_{22}^1 & \dots & a_{2n}^1 \\ \vdots & \vdots & & \vdots \\ 0 & a_{n2}^1 & \dots & a_{nn}^1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1^1 \\ b_2^1 \\ \vdots \\ b_n^1 \end{pmatrix}.$$

Analogously, after k steps we obtain the system

$$\begin{pmatrix} a_{11}^1 & a_{12}^1 & \dots & a_{1k}^1 & a_{1,k+1}^1 & \dots & a_{1n}^1 \\ 0 & a_{22}^2 & \dots & a_{2k}^2 & a_{2,k+1}^2 & \dots & a_{2n}^2 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & a_{kk}^k & a_{k,k+1}^k & \dots & a_{kn}^k \\ 0 & 0 & \dots & 0 & a_{k+1,k+1}^k & \dots & a_{k+1,n}^k \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 & a_{n,k+1}^k & \dots & a_{nn}^k \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \\ x_{k+1} \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1^1 \\ b_2^2 \\ \vdots \\ b_k^k \\ b_{k+1}^k \\ \vdots \\ b_n^k \end{pmatrix}.$$

If $a_{kk}^k \neq 0$, denote $m_{ik} = \frac{a_{ik}^k}{a_{kk}^k}$ and we get

$$\begin{aligned} a_{ij}^{k+1} &= a_{ij}^k - m_{ik} a_{kj}^k, \quad j = k, ..., n \\ b_i^{k+1} &= b_i^k - m_{ik} b_k^k, \quad i = k+1, ..., n. \end{aligned}$$

After $n-1$ steps we obtain the system

$$\left(\begin{array}{cccc} a_{11}^1 & a_{12}^1 & ... & a_{1n}^1 \\ 0 & a_{22}^2 & ... & a_{2n}^2 \\ 0 & 0 & ... & a_{3n}^3 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & ... & a_{nn}^{n-1} \end{array} \right) \left(\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{array} \right) = \left(\begin{array}{l} b_1^1 \\ b_2^2 \\ b_3^3 \\ \vdots \\ b_n^{n-1} \end{array} \right).$$

Remark 4 *The total number of elementary operations is of order* $\frac{2}{3}n^3$.

Example 5 *Consider the system*

$$\left(\begin{array}{cc} 0.0001 & 1 \\ 1 & 1 \end{array} \right) \left(\begin{array}{c} x \\ y \end{array} \right) = \left(\begin{array}{c} 1 \\ 2 \end{array} \right).$$

Gauss algorithm yields: $m_{21} = \frac{a_{21}}{a_{11}} = \frac{1}{0.0001}$

$$\left(\begin{array}{cc} 0.0001 & 1 \\ 1 - 0.0001 * m_{21} = 0 & 1 - 1 * m_{21} = -9999 \end{array} \right) \left(\begin{array}{c} x \\ y \end{array} \right)$$
$$= \left(\begin{array}{c} 1 \\ 2 - 1 * m_{21} = -9998 \end{array} \right)$$

$$\Rightarrow y = \frac{9998}{9999} = 0.(9998) \approx 1.$$

Replacing in the first equation we get

$$x = 1.000(1000) \approx 1.$$

By division with a pivot of small absolute value there could be induced errors. For avoiding this there are two ways:

A) Partial pivoting: finding an index $p \in \{k, ..., n\}$ such that:

$$\left| a_{p,k}^{k} \right| = \max_{i=\overline{k,n}} \left| a_{i,k}^{k} \right|.$$

B) Total pivoting: finding $p, q \in \{k, ..., n\}$ such that:

$$\left|a_{p,q}^{k}\right| = \max_{i,j=\overline{k,n}} \left|a_{ij}^{k}\right|,$$

Example 6 *Solve the following system of equations using partial pivoting:*

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 3 & 1 & 5 \\ -1 & 1 & -5 & 3 \\ 3 & 1 & 7 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 10 \\ 31 \\ -2 \\ 18 \end{bmatrix}.$$

The pivot is a_{41}. We interchange the $1-$st line and the $4-$th line. We have

$$\begin{bmatrix} 3 & 1 & 7 & -2 \\ 2 & 3 & 1 & 5 \\ -1 & 1 & -5 & 3 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 18 \\ 31 \\ -2 \\ 10 \end{bmatrix},$$

then

$$\begin{array}{r|cccc|c|}
\textit{pivot element}\rightarrow & \mathbf{3} & 1 & 7 & -2 & 18 \\
m_{21} = \frac{2}{3} & 0 & 2.33 & -3.66 & 6.33 & 19 \\
m_{31} = -\frac{1}{3} & 0 & 1.33 & -2.66 & 2.33 & 4 \\
m_{41} = \frac{1}{3} & 0 & 0.66 & -1.33 & 1.66 & 4
\end{array}.$$

Subtracting multiplies of the first equation from the three others gives

$$\begin{array}{r|cccc|c|}
 & 3 & 1 & 7 & -2 & 18 \\
\textit{pivot element}\rightarrow & 0 & \mathbf{2.33} & -3.66 & 6.33 & 19 \\
m_{32} = \frac{1.33}{2.33} & 0 & 1.33 & -2.66 & 2.33 & 4 \\
m_{42} = \frac{0.66}{2.33} & 0 & 0.66 & -1.33 & 1.66 & 4
\end{array}.$$

Subtracting multiplies, of the second equation from the last two equations, gives

$$\begin{array}{r|cccc|c|}
 & 3 & 1 & 7 & -2 & 18 \\
 & 0 & 2.33 & -3.66 & 6.33 & 19 \\
\textit{pivot element}\rightarrow & 0 & 0 & \mathbf{-0.57} & -1.28 & -6.85 \\
m_{43} = \frac{0.28}{0.57} & 0 & 0 & -0.28 & -0.14 & -1.42
\end{array}.$$

Subtracting multiplies, of the third equation form the last one, gives the upper triangular system

$$\left[\begin{array}{cccc|c} 3 & 1 & 7 & -2 & 18 \\ 0 & 2.33 & -3.66 & 6.33 & 19 \\ 0 & 0 & \mathbf{-0.57} & -1.28 & -6.85 \\ 0 & 0 & 0 & 0.5 & 2 \end{array}\right].$$

The process of the back substitution algorithm applied to the triangular system produces the solution

$$\begin{aligned} x_4 &= \frac{2}{0.5} = 4 \\ x_3 &= \frac{-6.85 + 1.28x_4}{-0.57} = 3 \\ x_2 &= \frac{19 + 3.66x_3 - 6.33x_4}{2.33} = 2 \\ x_1 &= \frac{18 - x_2 - 7x_3 + 2x_4}{3} = 1. \end{aligned}$$

Example 7 *Solve the system:*

$$\begin{cases} 2x + y = 3 \\ 3x - 2y = 1 \end{cases}$$

Sol.

$$\begin{cases} 2x + y = 3 \\ 3x - 2y = 1 \end{cases}$$

The extended matrix is

$$\left[\begin{array}{ccc} 2 & 1 & | & 3 \\ 3 & -2 & | & 1 \end{array}\right]$$

and the pivot is 3. We interchange the lines:

$$\left[\begin{array}{ccc} 3 & -2 & | & 1 \\ 2 & 1 & | & 3 \end{array}\right]$$

We have $L_2 - \frac{2}{3}L_1 \to L_2$ and obtain

$$\left[\begin{array}{ccc} 3 & -2 & | & 1 \\ 0 & \frac{7}{3} & | & \frac{7}{3} \end{array}\right]$$

so the system becames

$$\begin{cases} 3x - 2y = 1 \\ \frac{7}{3}y = \frac{7}{3} \end{cases} .$$

Solution is

$$\begin{cases} x = 1 \\ y = 1 \end{cases}.$$

Example 8 *Solve the following system using Gauss elimination method:*

$$\begin{cases} x_1 + x_2 + x_3 = 4 \\ 2x_1 - 2x_2 + 3x_3 = 5 \\ x_1 - x_2 + 4x_3 = 5. \end{cases}$$

4.2.2. Gauss-Jordan method ("total elimination" method)

Consider the linear system $Ax = b$, i.e.,

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}. \tag{4}$$

We make transformations, like in Gauss elimination method, to make zeroes in the lines $i+1, i+2, ..., n$ and then, also in the lines $1, 2, ..., i-1$ such that the system to be reducing to:

$$\begin{pmatrix} a_{11}^1 & 0 & \cdots & 0 \\ 0 & a_{22}^2 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & a_{nn}^n \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1^1 \\ b_2^2 \\ b_3^3 \\ \vdots \\ b_n^n \end{pmatrix}.$$

The solution is obtained by

$$x_i = \frac{b_i^i}{a_{ii}^i}, \quad i = 1, ..., n.$$

Definition 9 *A $n \times n$ matrix A is* **strictly diagonally dominant** *if*

$$|a_{ii}| > \sum_{j=1, j\neq i}^{n} \left|a_{ij}\right|, \text{ for } \ i = 1, 2, ..., n.$$

Theorem 10 *If A is a strictly diagonally dominant matrix, then A is nonsingular and moreover, Gaussian elimination can be performed on any linear system $Ax = b$ without rows or columns interchanges, and the computations are stable with respect to the growth of rounding errors.*

COURSE 10

4.2.3. Factorization methods - LU methods

The matrix A can be factored into the product of a lower triangular matrix L and an upper triangular matrix U, namely $A = LU$.

$$Ax = b \Longleftrightarrow LUx = b,$$

where

$$L = \begin{pmatrix} l_{11} & 0 & \dots & 0 \\ l_{21} & l_{22} & \dots & 0 \\ \vdots & & & \\ l_{n1} & l_{n2} & \dots & l_{nn} \end{pmatrix} \qquad U = \begin{pmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{2n} \\ \vdots & & & \\ 0 & 0 & \dots & u_{nn} \end{pmatrix}.$$

We solve the systems in two stages:

First stage: Solve $Lz = b$,

Second stage: Solve $Ux = z$.

Methods for computing matrices L and U : **Doolittle method** where all diagonal elements of L have to be 1; **Crout method** where all diagonal elements of U have to be 1 and **Choleski method** where $l_{ii} = u_{ii}$ for $i = 1, ..., n$.

Remark 1 *LU factorizations are modified forms of Gauss elimination method.*

Doolittle method

We consider that A is a strictly diagonally dominant matrix, so $a_{kk} \neq 0$, $k = \overline{1, n-1}$. Denote

$$l_{i,k} := \frac{a_{i,k}^{(k-1)}}{a_{k,k}^{(k-1)}}, \quad i = \overline{k+1, n}$$

$$t^{(k)} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ l_{k+1,k} \\ \cdots \\ l_{n,k} \end{bmatrix},$$

having zeros for the first k-th lines, and

$$M_k = I_n - t^{(k)} e_k \in \mathcal{M}_{n\times n}(\mathbb{R}) \tag{1}$$

where $e_k = \begin{pmatrix} 0 & ... & 1 & ... & 0 \end{pmatrix}$ is the k-unit vector of dimension n,(has 1 on the k-th position) and $I_n = \begin{pmatrix} 1 & 0 & ... & 0 \\ 0 & 1 & ... & 0 \\ ... & & & \\ 0 & 0 & ... & 1 \end{pmatrix}$ is the identity matrix of order n.

$a_{i,k}^{(0)}$ are elements of A; $a_{i,k}^{(1)}$ are elements of $M_1 \cdot A$; ...; $a_{i,k}^{(k-1)}$ are elements of $M_{k-1} \ldots \cdot M_1 \cdot A$.

Definition 2 *The matrix M_k is called* **the Gauss matrix**, *the components $l_{i,k}$ are called* **the Gauss multiplies** *and the vector $t^{(k)}$ is* **the Gauss vector**.

Remark 3 *If $A \in \mathcal{M}_{n\times n}(\mathbb{R})$, then the Gauss matrices $M_1, \ldots, M_{n-1}$ can be determined such that*

$$U = M_{n-1} \cdot M_{n-2} \ldots M_2 \cdot M_1 \cdot A$$

is an upper triangular matrix. Moreover, if we choose

$$L = M_1^{-1} \cdot M_2^{-1} \ldots M_{n-1}^{-1}$$

then

$$A = L \cdot U.$$

Example 4 *Find LU factorization for the matrix*

$$A = \left(\begin{array}{cc} 2 & 1 \\ 6 & 8 \end{array} \right).$$

Solve the system $\left\{ \begin{array}{c} 2x_1 + x_2 = 3 \\ 6x_1 + 8x_2 = 9 \end{array} \right. .$

Sol.

$$M_1 = I_2 - t^{(1)} e_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 \\ \frac{6}{2} \end{pmatrix} \begin{pmatrix} 1 & 0 \end{pmatrix}$$

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

We have

$$U = M_1 A = \begin{pmatrix} 1 & 0 \\ -3 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 \\ 6 & 8 \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 0 & 5 \end{pmatrix}$$

$$L = M_1^{-1} = \begin{pmatrix} 1 & 0 \\ 3 & 1 \end{pmatrix}.$$

So

$$A = \begin{pmatrix} 2 & 1 \\ 6 & 8 \end{pmatrix} = L \cdot U = \begin{pmatrix} 1 & 0 \\ 3 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 \\ 0 & 5 \end{pmatrix}.$$

We have

$$L \cdot U \cdot x = \begin{pmatrix} 3 \\ 9 \end{pmatrix}$$
$$Ux = z$$

and

$$\begin{pmatrix} 1 & 0 \\ 3 & 1 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} 3 \\ 9 \end{pmatrix} \Rightarrow z = \begin{pmatrix} 3 \\ 0 \end{pmatrix}$$
$$\begin{pmatrix} 2 & 1 \\ 0 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 3 \\ 0 \end{pmatrix} \Rightarrow x = \begin{pmatrix} 1.5 \\ 0 \end{pmatrix}.$$

4.3. Iterative methods for solving linear systems

Because of round-off errors, direct methods become less efficient than iterative methods for large systems (>100 000 variables).

An iterative scheme for linear systems consists of converting the system

$$Ax = b \tag{2}$$

to the form

$$x = \tilde{b} - Bx.$$

After an initial guess for $x^{(0)}$, the sequence of approximations of the solution $x^{(0)}, x^{(1)}, ..., x^{(k)}, ...$is generated by computing

$$x^{(k)} = \tilde{b} - Bx^{(k-1)}, \quad \text{for } k = 1, 2, 3, ...$$

4.3.1. Jacobi iterative method

Consider the $n \times n$ linear system,

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n = b_2 \\ \qquad\qquad\qquad ... \\ a_{n1}x_1 + a_{n2}x_2 + ... + a_{nn}x_n = b_n, \end{cases}$$

where we assume that the diagonal terms a_{11}, a_{22}, ..., a_{nn} are all nonzero.

We begin our iterative scheme by solving each equation for one of the variables:

$$\begin{cases} x_1 = u_{12}x_2 + ... + u_{1n}x_n + c_1 \\ x_2 = u_{21}x_1 + ... + u_{2n}x_n + c_2 \\ ... \\ x_n = u_{n1}x_1 + ... + u_{nn-1}x_{n-1} + c_n, \end{cases}$$

where $u_{ij} = -\frac{a_{ij}}{a_{ii}}$, $c_i = \frac{b_i}{a_{ii}}$, $i = 1, ..., n$.

Let $x^{(0)} = (x_1^{(0)}, x_2^{(0)}, ..., x_n^{(0)})$ be an initial approximation of the solution. The $k+1$-th approximation is:

$$\begin{cases} x_1^{(k+1)} = u_{12}x_2^{(k)} + ... + u_{1n}x_n^{(k)} + c_1 \\ x_2^{(k+1)} = u_{21}x_1^{(k)} + u_{23}x_3^{(k)} + ... + u_{2n}x_n^{(k)} + c_2 \\ ... \\ x_n^{(k+1)} = u_{n1}x_1^{(k)} + ... + u_{nn-1}x_{n-1}^{(k)} + c_n, \end{cases}$$

for $k = 0, 1, 2, ...$

An algorithmic form:

$$x_i^{(k)} = \frac{b_i - \sum\limits_{j=1, j\neq i}^{n} a_{ij}x_j^{(k-1)}}{a_{ii}}, \; i = 1, 2, ..., n, \text{ for } k \geq 1.$$

The iterative process is terminated when a convergence criterion is satisfied.

Stopping criterions: $\left|x^{(k)} - x^{(k-1)}\right| < \varepsilon$ or $\frac{\left|x^{(k)} - x^{(k-1)}\right|}{\left|x^{(k)}\right|} < \varepsilon$, with $\varepsilon > 0$ - a prescribed tolerance.

Example 5 *Solve the following system using the Jacobi iterative method. Use $\varepsilon = 10^{-3}$ and $x^{(0)} = (0\ 0\ 0\ 0)$ as the starting vector.*

$$\left\{\begin{array}{rcrcrcrcr}
7x_1 & - & 2x_2 & + & x_3 & & & = & 17 \\
x_1 & - & 9x_2 & + & 3x_3 & - & x_4 & = & 13 \\
2x_1 & & & + & 10x_3 & + & x_4 & = & 15 \\
x_1 & - & x_2 & + & x_3 & + & 6x_4 & = & 10.
\end{array}\right.$$

These equations can be rearranged to give

$$\begin{aligned}
x_1 &= (17 + 2x_2 - x_3)/7 \\
x_2 &= (-13 + x_1 + 3x_3 - x_4)/9 \\
x_3 &= (15 - 2x_1 - x_4)/10 \\
x_4 &= (10 - x_1 + x_2 - x_3)/6
\end{aligned}$$

and, for example,

$$\begin{aligned}
x_1^{(1)} &= (17 + 2x_2^{(0)} - x_3^{(0)})/7 \\
x_2^{(1)} &= (-13 + x_1^{(0)} + 3x_3^{(0)} - x_4^{(0)})/9 \\
x_3^{(1)} &= (15 - 2x_1^{(0)} - x_4^{(0)})/10 \\
x_4^{(1)} &= (10 - x_1^{(0)} + x_2^{(0)} - x_3^{(0)})/6.
\end{aligned}$$

Substitute $x^{(0)} = (0,0,0,0)$ *into the right-hand side of each of these equations to get*

$$\begin{aligned}
x_1^{(1)} &= (17 + 2 \cdot 0 - 0)/7 = 2.428\ 571\ 429 \\
x_2^{(1)} &= (-13 + 0 + 3 \cdot 0 - 0)/9 = -1.444\ 444\ 444 \\
x_3^{(1)} &= (15 - 2 \cdot 0 - 0)/10 = 1.5 \\
x_4^{(1)} &= (10 - 0 + 0 - 0)/6 = 1.666\ 666\ 667
\end{aligned}$$

and so $x^{(1)} = (2.428\ 571\ 429, -1.444\ 444\ 444, 1.5, 1.666\ 666\ 667)$.
The Jacobi iterative process:

$$\begin{aligned}
x_1^{(k+1)} &= \left(17 + 2x_2^{(k)} - x_3^{(k)}\right)/7 \\
x_2^{(k+1)} &= \left(-13 + x_1^{(k)} + 3x_3^{(k)} - x_4^{(k)}\right)/9 \\
x_3^{(k+1)} &= \left(15 - 2x_1^{(k)} - x_4^{(k)}\right)/10 \\
x_4^{(k+1)} &= \left(10 - x_1^{(k)} + x_2^{(k)} - x_3^{(k)}\right)/6, \qquad k \geq 1.
\end{aligned}$$

We obtain a sequence that converges to

$$\mathbf{x}^{(9)} = (2.000127203, -1.000100162, 1.000118096, 1.000162172).$$

4.3.2. Gauss-Seidel iterative method

Almost the same as Jacobi method, except that each x-value is improved using the most recent approx. of the other variables.

For a $n \times n$ system, the $k+1$-th approximation is:

$$\begin{cases} x_1^{(k+1)} = u_{12}x_2^{(k)} + ... + u_{1n}x_n^{(k)} + c_1 \\ x_2^{(k+1)} = u_{21}x_1^{(k+1)} + u_{23}x_3^{(k)} + ... + u_{2n}x_n^{(k)} + c_2 \\ ... \\ x_n^{(k+1)} = u_{n1}x_1^{(k+1)} + ... + u_{nn-1}x_{n-1}^{(k+1)} + c_n, \end{cases}$$

with $k = 0, 1, 2, ...;$ $u_{ij} = -\frac{a_{ij}}{a_{ii}}$, $c_i = \frac{b_i}{a_{ii}}$, $i = 1, ..., n$ (as in Jacobi method).

Algorithmic form:

$$x_i^{(k)} = \frac{b_i - \sum\limits_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum\limits_{j=i+1}^{n} a_{ij}x_j^{(k-1)}}{a_{ii}}$$

for each $i = 1, 2, ...n,$ and for $k \geq 1$.

Stopping criterions: $\left|x^{(k)} - x^{(k-1)}\right| < \varepsilon$, or $\frac{\left|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\right|}{\left|\mathbf{x}^{(k)}\right|} < \varepsilon$, with ε - a prescribed tolerance, $\varepsilon > 0$.

Remark 6 *Because the new values can be immediately stored in the location that held the old values, the storage requirements for* $\mathbf{x}$ *with the Gauss-Seidel method is half than that for Jacobi method and the rate of convergence is faster.*

Example 7 *Solve the following system using the Gauss-Seidel iterative method. Use* $\varepsilon = 10^{-3}$ *and* $\mathbf{x}^{(0)} = (0\ 0\ 0\ 0)$ *as the starting vector.*

$$\begin{cases} 7x_1 & - & 2x_2 & + & x_3 & & & = & 17 \\ x_1 & - & 9x_2 & + & 3x_3 & - & x_4 & = & 13 \\ 2x_1 & & & + & 10x_3 & + & x_4 & = & 15 \\ x_1 & - & x_2 & + & x_3 & + & 6x_4 & = & 10 \end{cases}$$

We have

$$
\begin{aligned}
x_1 &= (17 + 2x_2 - x_3)/7 \\
x_2 &= (-13 + x_1 + 3x_3 - x_4)/9 \\
x_3 &= (15 - 2x_1 - x_4)/10 \\
x_4 &= (10 - x_1 + x_2 - x_3)/6,
\end{aligned}
$$

and, for example,

$$
\begin{aligned}
x_1^{(1)} &= (17 + 2x_2^{(0)} - x_3^{(0)})/7 \\
x_2^{(1)} &= (-13 + x_1^{(1)} + 3x_3^{(0)} - x_4^{(0)})/9 \\
x_3^{(1)} &= (15 - 2x_1^{(1)} - x_4^{(0)})/10 \\
x_4^{(1)} &= (10 - x_1^{(1)} + x_2^{(1)} - x_3^{(1)})/6,
\end{aligned}
$$

which provide the following Gauss-Seidel iterative process:

$$\begin{aligned}
x_1^{(k+1)} &= \left(17 + 2x_2^{(k)} - x_3^{(k)}\right)/7 \\
x_2^{(k+1)} &= \left(-13 + x_1^{(k+1)} + 3x_3^{(k)} - x_4^{(k)}\right)/9 \\
x_3^{(k+1)} &= \left(15 - 2x_1^{(k+1)} - x_4^{(k)}\right)/10 \\
x_4^{(k+1)} &= \left(10 - x_1^{(k+1)} + x_2^{(k+1)} - x_3^{(k+1)}\right)/6, \quad \textit{for } k \geq 1.
\end{aligned}$$

Substitute $\mathbf{x}^{(0)} = (0, 0, 0, 0)$ *into the right-hand side of each of these equations to get*

$$\begin{aligned}
x_1^{(1)} &= (17 + 2 \cdot 0 - 0)/7 = 2.428\ 571\ 429 \\
x_2^{(1)} &= (-13 + 2.428\ 571\ 429 + 3 \cdot 0 - 0)/9 = -1.1746031746 \\
x_3^{(1)} &= (15 - 2 \cdot 2.428\ 571\ 429 - 0)/10 = 1.0142857143 \\
x_4^{(1)} &= (10 - 2.428\ 571\ 429 - 1.1746031746 - 1.0142857143)/6 \\
&= 0.8970899472
\end{aligned}$$

and so

$$\mathbf{x}^{(1)} = (2.428571429 - 1.1746031746, 1.0142857143, 0.8970899472).$$

Similar procedure generates a sequence that converges to

$$\mathbf{x}^{(5)} = (2.000025, -1.000130, 1.000020.0.999971).$$

4.3.3. Relaxation method

In case of convergence, the Gauss-Seidel method is faster than Jacobi method. The convergence can be more improved using **relaxation method (SOR method) (**SOR=Succesive Over Relaxation)

Algorithmic form of the method:

$$x_i^{(k)} = \frac{\omega}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)} \right) + (1-\omega) x_i^{(k-1)}$$

for each $i = 1, 2, ...n,$ and for $k \geq 1$.

For $0 < \omega < 1$ the procedure is called **under relaxation method,** that can be used to obtain convergence for systems which are not convergent by Gauss-Siedel method.

For $\omega > 1$ the procedure is called **over relaxation method,** that can be used to accelerate the convergence for systems which are convergent by Gauss-Siedel method.

By Kahan's Theorem follows that the method converges for $0 < \omega < 2$.

Remark 8 *For $\omega = 1$, relaxation method is Gauss-Seidel method.*

Example 9 *Solve the following system, using relaxation iterative method. Use $\varepsilon = 10^{-3}$, $\mathbf{x}^{(0)} = (1\ 1\ 1)$ and $\omega = 1.25$,*

$$\begin{array}{rcrcrcr} 4x_1 & + & 3x_2 & & & = & 24 \\ 3x_1 & + & 4x_2 & - & x_3 & = & 30 \\ & - & x_2 & + & 4x_3 & = & -24 \end{array}$$

We have

$$\begin{aligned} x_1^{(k)} &= 7.5 - 0.937x_2^{(k-1)} - 0.25x_1^{(k-1)} \\ x_2^{(k)} &= 9.375 - 9.375x_1^{(k)} + 0.3125x_3^{(k-1)} - 0.25x_2^{(k-1)} \\ x_3^{(k)} &= -7.5 + 0.3125x_2^{(k)} - 0.25x_3^{(k-1)}, \quad \textit{for } k \geq 1. \end{aligned}$$

The solution is $(3, 4, -5)$.

4.3.4 The matriceal formulations of the iterative methods

Split the matrix A into the sum

$$A = D + L + U,$$

where D is the diagonal of A, L the lower triangular part of A, and U the upper triangular part of A. That is,

$$D = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a_{nn} \end{bmatrix}, \qquad L = \begin{bmatrix} 0 & \cdots & & 0 \\ a_{21}\vdots & & & \ddots \\ \vdots & \ddots & \ddots & \\ a_{n1} & \cdots & a_{n,n-1} & 0 \end{bmatrix},$$

$$U = \begin{bmatrix} 0 & a_{12} & \cdots & a_{1n} \\ \vdots & \ddots & \ddots & \vdots \\ & & \ddots & a_{n-1,n} \\ 0 & \cdots & & 0 \end{bmatrix}$$

The system $Ax = b$ can be written as

$$(D + L + U)\mathbf{x} = \mathbf{b}.$$

The **Jacobi method** in matriceal form is given by:

$$D\mathbf{x}^{(k)} = -(L+U)\mathbf{x}^{(k-1)} + \mathbf{b}$$

the **Gauss-Seidel method** in matriceal form is given by:

$$(D+L)\mathbf{x}^{(k)} = -U\mathbf{x}^{(k-1)} + \mathbf{b}$$

and **the relaxation method** in matriceal form is given by:

$$(D+\omega L)\mathbf{x}^{(k)} = ((1-\omega)D - \omega U)\mathbf{x}^{(k-1)} + \omega\mathbf{b}$$

Convergence of the iterative methods

Remark 10 *The convergence (or divergence) of the iterative process in the Jacobi and Gauss-Seidel methods does not depend on the initial guess, but depends only on the character of the matrices themselves. However, a good first guess in case of convergence will make for a relatively small number of iterations.*

A sufficient condition for convergence:

Theorem 11 (**Convergence Theorem**) *If A is strictly diagonally dominant, then the Jacobi, Gauss-Seidel and relaxation methods converge for any choice of the starting vector $\mathbf{x}^{(0)}$.*

Example 12 *Consider the system of equations*

$$\begin{bmatrix} 3 & 1 & 1 \\ -2 & 4 & 0 \\ -1 & 2 & -6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 4 \\ 1 \\ 2 \end{bmatrix}.$$

The coefficient matrix of the system is strictly diagonally dominant since

$$\begin{aligned} |a_{11}| &= |3| = 3 > |1| + |1| = 2 \\ |a_{22}| &= |4| = 4 > |-2| + |0| = 2 \\ |a_{33}| &= |-6| = 6 > |-1| + |2| = 3. \end{aligned}$$

Hence, if the Jacobi or Gauss-Seidel method are used to solve the system of equations, they will converge for any choice of the starting vector $\mathbf{x}^{(0)}$.

Example 13 *Consider the linear system*

$$\begin{aligned} 4x_1 + x_2 &= 3 \\ 2x_1 + 5x_2 &= 1. \end{aligned}$$

Perform two iterations of Jacobi, Gauss-Seidel and relaxation methods to this system, beginning with the vector $x = [3, 11]$ *and for* $\omega = 1.25$.

(Solutions of the system are $7/9$ *and* $-1/9$*).*

COURSE 11

5. Numerical methods for solving nonlinear equations in $\mathbb{R}$

Let $f : \Omega \to \mathbb{R}$, $\Omega \subset \mathbb{R}$. Consider the equation

$$f(x) = 0, \quad x \in \Omega. \tag{1}$$

Example. Kepler's Equation: consider a two-body problem like a satellite orbiting the earth or a planet revolving around the sun. Kepler discovered that the orbit is an ellipse and the central body F (earth, sun) is in a focus of the ellipse. The speed of the satellite P is not uniform: near the earth it moves faster than far away. It is used Kepler's law to predict where the satellite will be at a given time. If we want to know the position of the satellite for t = 9 minutes, then we have to solve the equation $f(E) = E - 0.8 sinE - 2\pi/10 = 0$.

We attach a mapping $F : D \to D,\ D \subset \Omega^n$ to this equation.

Let $(x_0, ..., x_{n-1}) \in D$. Using F and the numbers $x_0, x_1, ..., x_{n-1}$ we construct iteratively the sequence

$$x_0, x_1, ..., x_{n-1}, x_n, ... \tag{2}$$

with

$$x_i = F\left(x_{i-n}, ..., x_{i-1}\right), \quad i = n, \ldots . \tag{3}$$

The problem consists in choosing F and $x_0, ..., x_{n-1} \in D$ such that the sequence (2) to be convergent to the solution of the equation (1).

Definition 1 *The procedure of approximation the solution of equation (1) by the elements of the sequence (2), computed as in (3), is called* F-**method.**

The numbers $x_0, x_1, ..., x_{n-1}$ *are called* **the starting points** *and the* k*-th element of the sequence (2) is called an approximation of* k*-th order of the solution.*

If the set of starting points has only one element then the F-method is **an one-step method;** if it has more than one element then the F-method is **a multistep method**.

Definition 2 *If the sequence (2) converges to the solution of the equation (1) then the F-method is convergent, otherwise it is divergent.*

Definition 3 *Let $\alpha \in \Omega$ be a solution of the equation (1) and let $x_0, x_1, ..., x_{n-1}, x_n, ...$ be the sequence generated by a given F-method. The number p having the property*

$$\lim_{x_i \to \alpha} \frac{\alpha - F(x_{i-n}, ..., x_i)}{(\alpha - x_i)^p} = C \neq 0, \quad C = constant,$$

is called the order *of the F-method.*

We construct some classes of F-methods based on the interpolation procedures.

Let $\alpha \in \Omega$ be a solution of the equation (1) and $V(\alpha)$ a neighborhood of α. Assume that f has inverse on $V(\alpha)$ and denote $g := f^{-1}$. Since

$$f(\alpha) = 0$$

it follows that

$$\alpha = g(0).$$

This way, the approximation of the solution α is reduced to the approximation of $g(0)$.

Definition 4 *The approximation of g by means of an interpolating method, and of α by the value of g at the point zero is called* **the inverse interpolation procedure.**

5.1. One-step methods

Let F be a one-step method, i.e., for a given x_i we have $x_{i+1} = F\left(x_i\right).$

Remark 5 *If $p = 1$ the convergence condition is $\left|F'\left(x\right)\right| < 1$.*

If $p > 1$ there always exists a neighborhood of α where the F-method converges.

All information on f are given at a single point, the starting value ⇒ we are lead to Taylor interpolation.

Theorem 6 *Let α be a solution of equation (1), $V(\alpha)$ a neighborhood of α, $x, x_i \in V(\alpha)$, f fulfills the necessary continuity conditions. Then we have the following method, denoted by F_m^T, for approximating α:*

$$F_m^T(x_i) = x_i + \sum_{k=1}^{m-1} \tfrac{(-1)^k}{k!}[f(x_i)]^k g^{(k)}(f(x_i)), \tag{4}$$

where $g = f^{-1}$.

Proof. There exists $g = f^{-1} \in C^m[V(0)]$. Let $y_i = f(x_i)$ and consider Taylor interpolation formula

$$g(y) = (T_{m-1}g)(y) + (R_{m-1}g)(y),$$

with

$$(T_{m-1}g)(y) = \sum_{k=0}^{m-1} \tfrac{1}{k!}(y - y_i)^k g^{(k)}(y_i),$$

and $R_{m-1}g$ is the corresponding remainder.

Since $\alpha = g(0)$ and $g \approx T_{m-1}g$, it follows

$$\alpha \approx (T_{m-1}g)(0) = x_i + \sum_{k=1}^{m-1} \tfrac{(-1)^k}{k!} y_i^k g^{(k)}(y_i).$$

Hence,

$$x_{i+1} := F_m^T(x_i) = x_i + \sum_{k=1}^{m-1} \tfrac{(-1)^k}{k!}[f(x_i)]^k g^{(k)}(f(x_i))$$

is an approximation of α, and F_m^T is an approximation method for the solution α. ■

Concerning the order of the method F_m^T we state:

Theorem 7 *If* $g = f^{-1}$ *satisfies condition* $g^{(m)}(0) \neq 0$, *then* $\mathrm{ord}(F_m^T) = m$.

Remark 8 *We have an upper bound for the absolute error in approximating* α *by* x_{i+1} :

$$\left|\alpha - F_m^T(x_i)\right| \leq \frac{1}{m!}[f(x_i)]^m M_m g, \quad \textit{with } M_m g = \max_{y \in V(0)} \left|g^{(m)}(y)\right|.$$

Particular cases.

1) Case $m = 2$.

$$F_2^T(x_i) = x_i - \frac{f(x_i)}{f\,'(x_i)}.$$

This method is called **Newton's method** (**the tangent method**). Its order is 2.

2) Case $m = 3$.

$$F_3^T(x_i) = x_i - \frac{f(x_i)}{f\,'(x_i)} - \frac{1}{2}\left[\frac{f(x_i)}{f\,'(x_i)}\right]^2 \frac{f\,''(x_i)}{f\,'(x_i)},$$

with $\text{ord}(F_3^T) = 3$. So, this method converges faster than F_2^T.

3) Case $m = 4$.

$$F_4^T(x_i) = x_i - \frac{f(x_i)}{f\,'(x_i)} - \frac{1}{2}\frac{f''(x_i)f^2(x_i)}{[f\,'(x_i)]^3} + \frac{\left(f'''(x_i)f'(x_i) - 3[f\,''(x_i)]^2\right)f^3(x_i)}{3![f\,'(x_i)]^5}.$$

Remark 9 *The higher the order of a method is, the faster the method converges. Still, this doesn't mean that a higher order method is more efficient (computation requirements). By the contrary, the most efficient are the methods of relatively low order, due to their low complexity (methods F_2^T and F_3^T).*

5.1.1. Newton's method

Newton's method (Newton-Raphson method) named after Isaac Newton (1642–1726) and Joseph Raphson (1648–1715), is a root-finding algorithm which produces successively better approximations to the roots of a real-valued function.

The traces of this methods can be found in ancient times (Babylon and Egypt, 1800 B.C.), as it appears in the computation of the square root of a number. This method is so efficient in computing $\sqrt{a}$, that it is a choice even today in modern codes.

In solving nonlinear problems, Newton ($\approx$1669) and subsequently Raphson (1690) have dealt only with polynomial equations ($x^3 - 2x - 5 = 0$ is "the classical equation where the Newton method is applied"). Newton has also considered such iterations in solving Kepler's equation $x - e \sin x = M$.

According to Remark 5, there always exists a neighborhood of α where the $F-$method is convergent. Choosing x_0 in such a neighborhood allows approximating α by terms of the sequence

$$x_{i+1} = F_2^T (x_i) = x_i - \tfrac{f(x_i)}{f\,'(x_i)}, \quad i = 0, 1, ...,$$

with a prescribed error ε.

If α is a solution of equation (1) and $x_{n+1} = F_2^T (x_n)$, for approximation error, Remark 8 gives

$$\left|\alpha - x_{n+1}\right| \leq \tfrac{1}{2}[f\,(x_n)]^2 M_2 g.$$

Lemma 10 *Let $\alpha \in (a,b)$ be a solution of equation (1) and let $x_n = F_2^T(x_{n-1})$. Then*

$$|\alpha - x_n| \leq \frac{1}{m_1}|f(x_n)|, \quad \textit{with } m_1 \leq m_1 f = \min_{a\leq x\leq b}\left|f'(x)\right|.$$

Proof. We use the mean formula

$$f(\alpha) - f(x_n) = f'(\xi)(\alpha - x_n),$$

with $\xi \in$ to the interval determined by α and $x_{n.}$ From $f(\alpha) = 0$ and $|f'(x)| \geq m_1$ for $x \in (a,b)$, it follows $|f(x_n)| \geq m_1 |\alpha - x_n|$, that is

$$|\alpha - x_n| \leq \frac{1}{m_1}|f(x_n)|.$$

■

In practical applications the following evaluation is more useful:

Lemma 11 *If* $f \in C^2[a,b]$ *and* F_2^T *is convergent, then there exists* $n_0 \in \mathbb{N}$ *such that*

$$|x_n - \alpha| \leq |x_n - x_{n-1}|, \quad n > n_0.$$

Proof. We start with Taylor formula

$$f(x_n) = f(x_{n-1}) + (x_n - x_{n-1}) f'(x_{n-1}) + \tfrac{1}{2}(x_n - x_{n-1})^2 f''(\xi),$$

where ξ belongs to the interval determined by x_{n-1} and x_n.

Since $x_n = F_2^T(x_{n-1})$, it follows that

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})} \Longleftrightarrow f(x_{n-1}) + (x_n - x_{n-1}) f'(x_{n-1}) = 0,$$

thus we obtain

$$f(x_n) = \tfrac{1}{2}(x_n - x_{n-1})^2 f''(\xi).$$

Consequently,

$$|f(x_n)| \leq \tfrac{1}{2}(x_n - x_{n-1})^2 M_2 f,$$

and Lemma 10 yields $|\alpha - x_n| \leq \frac{1}{m_1} |f(x_n)|$ so

$$|\alpha - x_n| \leq \frac{1}{2m_1} (x_n - x_{n-1})^2 M_2 f.$$

Since F_2^T is convergent, there exists $n_0 \in \mathbb{N}$ such that

$$\frac{1}{2m_1} |x_n - x_{n-1}| M_2 f < 1, \quad n > n_0.$$

Hence,

$$|\alpha - x_n| \leq |x_n - x_{n-1}|, \quad n > n_0.$$

■

Remark 12 *The starting value is chosen randomly. If, after a fixed number of iterations the required precision is not achieved, i.e., condition $|x_n - x_{n-1}| \leq \varepsilon$, does not hold for a prescribed positive ε, the computation has to be started over with a new starting value.*

A modified form of Newton's method: - the same value during the computation of f':

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_0)}, \quad k = 0, 1, \ldots.$$

It is very useful because it doesn't request the computation of f' at x_j, $j = 1, 2, \ldots$ but the order is no longer equal to 2.

Another way for obtaining Newton's method.

We start with x_0 as an initial guess, sufficiently close to the α. Next approximation x_1 is the point at which the tangent line to f at $(x_0, f(x_0))$ crosses the Ox-axis. The value x_1 is much closer to the root α than x_0.

We write the equation of the tangent line at $(x_0, f(x_0))$:

$$y - f(x_0) = f'(x_0)(x - x_0).$$

If $x = x_1$ is the point where this line intersects the Ox-axis, then $y = 0$

$$-f(x_0) = f'(x_0)(x_1 - x_0),$$

and solving for x_1 gives

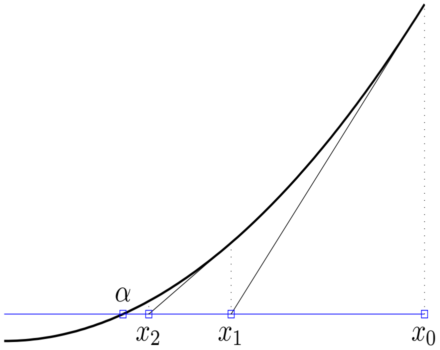
$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

By repeating the process using the tangent line at $(x_1, f(x_1))$, we obtain for x_2

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

For the general case we have

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \;\; n \geq 0. \tag{5}$$



The algorithm:

Let x_0 be the initial approximation.

for $n = 0, 1, ..., ITMAX$

$$x_{n+1} \leftarrow x_n - \frac{f(x_n)}{f'(x_n)}.$$

A stopping criterion is:

$$|f(x_n)| \leq \varepsilon \text{ or } \left|x_{n+1} - x_n\right| \leq \varepsilon \text{ or } \frac{\left|x_{n+1} - x_n\right|}{\left|x_{n+1}\right|} \leq \varepsilon,$$

where ε is a specified tolerance value.

Example 13 *Use Newton's method to compute a root of* $x^3 - x^2 - 1 = 0$*, to an accuracy of* 10^{-4}*. Use* $x_0 = 1$*.*

Sol. *The derivative of f is* $f'(x) = 3x^2 - 2x$. *Using* $x_0 = 1$ *gives* $f(1) = -1$ *and* $f'(1) = 1$ *and so the first Newton's iterate is*

$$x_1 = 1 - \frac{-1}{1} = 2 \text{ and } f(2) = 3,\ f'(2) = 8.$$

The next iterate is

$$x_2 = 2 - \frac{3}{8} = 1.625.$$

Continuing in this manner we obtain the sequence of approximations which converges to 1.465571.

5.2. Multistep methods

Lagrange inverse interpolation

Let $y_k = f(x_k)$, $k = 0, ..., n$, hence $x_k = g(y_k)$. We attach the Lagrange interpolation formula to y_k and $g(y_k)$, $k = 0, ..., n$:

$$g = L_n g + R_n g, \tag{6}$$

where

$$(L_n g)(y) = \sum_{k=0}^{n} \frac{(y-y_0)...(y-y_{k-1})(y-y_{k+1})...(y-y_n)}{(y_k-y_0)...(y_k-y_{k-1})(y_k-y_{k+1})...(y_k-y_n)} g(y_k). \tag{7}$$

Taking

$$F_n^L(x_0, ..., x_n) = (L_n g)(0),$$

F_n^L is a $(n+1)-$ steps method defined by

$$\begin{aligned} F_n^L(x_0, ..., x_n) &= \sum_{k=0}^{n} \frac{y_0 \cdots y_{k-1} y_{k+1} \cdots y_n}{(y_k-y_0)...(y_k-y_{k-1})(y_k-y_{k+1})...(y_k-y_n)} (-1)^n g(y_k) \\ &= \sum_{k=0}^{n} \frac{y_0 \cdots y_{k-1} y_{k+1} \cdots y_n}{(y_k-y_0)...(y_k-y_{k-1})(y_k-y_{k+1})...(y_k-y_n)} (-1)^n x_k. \end{aligned}$$

Concerning the convergence of this method we state:

Theorem 14 *If $\alpha \in (a,b)$ is solution of equation (**??**), f' is bounded on (a,b), and the starting values satisfy $|\alpha - x_k| < 1/c, \quad k = 0, ..., n,$ with $c =$constant, then the sequence*

$$x_{i+1} = F_n^L\left(x_{n-i}, ..., x_i\right), \quad i = n, n+1, ...$$

converges to α.

Remark 15 *The order $ord(F_n^L)$ is the positive solution of the equation*

$$t^{n+1} - t^n - ... - t - 1 = 0.$$

Particular cases.

1) For $n = 1$, the nodes x_0, x_1, we get **the secant method**

$$F_1^L\left(x_0, x_1\right) = x_1 - \frac{\left(x_1 - x_0\right) f\left(x_1\right)}{f\left(x_1\right) - f\left(x_0\right)},$$

Thus,

$$x_{k+1} := F_1^L\left(x_{k-1}, x_k\right) = x_k - \frac{\left(x_k - x_{k-1}\right) f\left(x_k\right)}{f\left(x_k\right) - f\left(x_{k-1}\right)}, \quad k = 1, 2, ...$$

is the new approximation obtained using the previous approximations x_{k-1}, x_k.

The *order* of this method is the positive solution of equation:

$$t^2 - t - 1 = 0,$$

so $ord(F_1^L) = \frac{(1+\sqrt{5})}{2}$.

A modified form of the secant method: if we keep x_1 fixed and we change every time the same interpolation node, i.e.,

$$x_{k+1} = x_k - \frac{\left(x_k - x_1\right) f\left(x_k\right)}{f\left(x_k\right) - f\left(x_1\right)}, \quad k = 2, 3, \ldots.$$

2) For $n = 2$, the nodes x_0, x_1, x_2 and we get

$$F_2^L(x_0, x_1, x_2) = \frac{x_0 f(x_1) f(x_2)}{[f(x_0) - f(x_1)][f(x_0) - f(x_2)]} + \frac{x_1 f(x_0) f(x_2)}{[f(x_1) - f(x_0)][f(x_1) - f(x_2)]} + \frac{x_2 f(x_0) f(x_1)}{[f(x_2) - f(x_0)][f(x_2) - f(x_1)]}.$$

The *order* of this method is the positive solution of equation:

$$t^3 - t^2 - t - 1 = 0,$$

so $ord(F_2^L) = 1.8394$.

Comparing the Newton's method and secant method with respect to the time needed for finding a root with some given precision, we have:

-Newton's method has more computation at one step: it is necessary to evaluate $f(x)$ and $f'(x)$. Secant method evaluates just $f(x)$ (supposing that $f(x_{previous})$ is stored.)

-The number of iterations for Newton's method is smaller (its order is $p_N = 2$). Secant method has order $p_S = 1.618$ and we have that three steps of this method are equivalent with two steps of Newton's method.

- It is proved that if the time for computing $f'(x)$ is greater than (0.44×the time for computing $f(x)$), then the secant method is faster.

Remark 16 *The computation time is not the unique criterion in choosing the method! Newton's method is easier to apply. If $f(x)$ is not explicitly known (for example, it is the solution of the numerical integration of a differential equation), then its derivative is computed numerically. If we consider the following expression for the numerical computation of derivative:*

$$f'(x) \approx \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}} \tag{8}$$

then the Newton's method becomes the secant method.

Another way of obtaining secant method.

Based on approx. the function by a straight line connecting two points on the graph of f (not required f to have opposite signs at the initial points).

The first point, x_2, of the iteration is taken to be the point of intersection of the Ox-axis and the secant line connecting two starting points

$(x_0, f(x_0))$ and $(x_1, f(x_1))$. The next point, x_3, is generated by the intersection of the new secant line, joining $(x_1, f(x_1))$ and $(x_2, f(x_2))$ with the Ox-axis. The new point, x_3, together with x_2, is used to generate the next point, x_4, and so on.

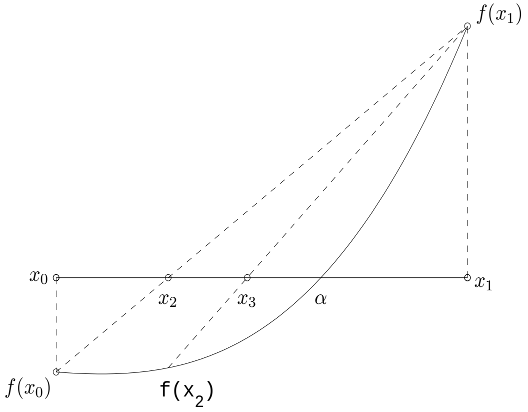
The formula for x_{n+1} is obtained by setting $x = x_{n+1}$ and $y = 0$ in the equation of the secant line from $(x_{n-1}, f(x_{n-1}))$ to $(x_n, f(x_n))$:

$$\frac{x - x_n}{x_{n-1} - x_n} = \frac{y - f(x_n)}{f(x_{n-1}) - f(x_n)} \Leftrightarrow x = x_n + \frac{(x_{n-1} - x_n)(y - f(x_n))}{f(x_{n-1}) - f(x_n)},$$

we get

$$x_{n+1} = x_n - f(x_n)\left[\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}\right]. \tag{9}$$

Note that x_{n+1} depends on the two previous elements of the sequence $\Rightarrow$ two initial guesses, x_0 and x_1,for generating $x_2, x_3, \ldots$.



The algorithm:

Let x_0 and x_1 be two initial approximations.

for $n = 1, 2, ..., ITMAX$

$$x_{n+1} \leftarrow x_n - f(x_n)\left[\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}\right].$$

A suitable stopping criterion is

$$|f(x_n)| \leq \varepsilon \text{ or } \left|x_{n+1} - x_n\right| \leq \varepsilon \text{ or } \frac{\left|x_{n+1} - x_n\right|}{\left|x_{n+1}\right|} \leq \varepsilon,$$

where ε is a specified tolerance value.

Example 17 *Use the secant method with* $x_0 = 1$ *and* $x_1 = 2$ *to solve* $x^3 - x^2 - 1 = 0$, *with* $\varepsilon = 10^{-4}$.

Sol. *With* $x_0 = 1$, $f(x_0) = -1$ *and* $x_1 = 2$, $f(x_1) = 3$, *we have*

$$x_2 = 2 - \frac{(2-1)(3)}{3-(-1)} = 1.25$$

from which $f(x_2) = f(1.25) = -0.609375$. *The next iterate is*

$$x_3 = 1.25 - \frac{(1.25-2)(-0.609375)}{-0.609375-3} = 1.3766234.$$

Continuing in this manner the iterations lead to the approximation 1.4655713.

Examples of other multi-step methods

1. THE BISECTION METHOD

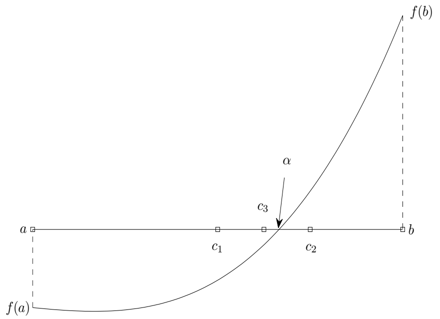
Let f be a given function, continuous on an interval $[a, b]$, such that

$$f(a)f(b) < 0. \tag{10}$$

By Mean Value Theorem, it follows that there exists at least one zero α of f in (a, b).

The bisection method is based on halving the interval $[a, b]$ to determine a smaller and smaller interval within α must lie.

First we give the midpoint of $[a, b]$, $c = (a + b)/2$ and then compute the product $f(c)f(b)$. If the product is negative, then the root is in the interval $[c, b]$ and we take $a_1 = c$, $b_1 = b$. If the product is positive, then the root is in the interval $[a, c]$ and we take $a_1 = a$, $b_1 = c$. Thus, a new interval containing α is obtained.



Bisection method

The algorithm:

Suppose $f(a)f(b) \leq 0$. Let $a_o = a$ and $b_o = b$.

```
for $n = 0, 1, ...,$ITMAX
$c \leftarrow \frac{a_n+b_n}{2}$
if $f(a_n)f(c) \leq 0$, set $a_{n+1} = a_n, b_{n+1} = c$
else, set $a_{n+1} = c, b_{n+1} = b_n$
```

The process of halving the new interval continues until the root is located as accurately as desired, namely

$$\frac{|a_n - b_n|}{|a_n|} < \varepsilon,$$

where a_n and b_n are the endpoints of the n-th interval $[a_n, b_n]$ and ε is a specified precision. The approximation of the solution will be $\frac{a_n+b_n}{2}$.

Some other stopping criterions: $|a_n - b_n| < \varepsilon$ or $|f(a_n)| < \varepsilon$.

Example 18 *The function $f(x) = x^3 - x^2 - 1$ has one zero in $[1, 2]$. Use the bisection algorithm to approximate the zero of f with precision 10^{-4}.*

Sol. *Since $f(1) = -1 < 0$ and $f(2) = 3 > 0$, then (10) is satisfied. Starting with $a_0 = 1$ and $b_0 = 2$, we compute*

$$c_0 = \frac{a_0 + b_0}{2} = \frac{1+2}{2} = 1.5 \text{ and } f(c_0) = 0.125.$$

Since $f(1.5)f(2) > 0$, the function changes sign on $[a_0, c_0] = [1, 1.5]$.

To continue, we set $a_1 = a_0$ *and* $b_1 = c_0$; *so*

$$c_1 = \frac{a_1 + b_1}{2} = \frac{1 + 1.5}{2} = 1.25 \text{ and } f(c_1) = -0.609375$$

Again, $f(1.25)f(1.5) < 0$ *so the function changes sign on* $[c_1, b_1] = [1.25,\ 1.5]$. *Next we set* $a_2 = c_1$ *and* $b_2 = b_1$. *Continuing in this manner we obtain a sequence* $(c_i)_{i>0}$ *which converges to* 1.465454, *the solution of the equation.*

2. THE METHOD OF FALSE POSITION

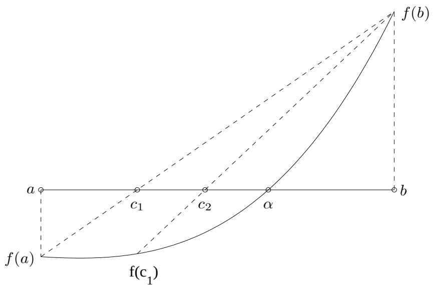
This method is also known as *regula falsi*, is similar to the Bisection method but has the advantage of being slightly faster than the latter. The function have to be continuous on $[a, b]$ with

$$f(a)f(b) < 0.$$

The point c is selected as point of intersection of the Ox-axis, and the straight line joining the points $(a,\ f(a))$ and $(b, f(b))$. From the equation of the secant line, it follows that

$$c = b - f(b)\frac{b-a}{f(b)-f(a)} = \frac{af(b)-bf(a)}{f(b)-f(a)} \qquad (11)$$

Compute $f(c)$ and repeat the procedure between the values at which the function changes sign, that is, if $f(a)f(c) < 0$ set $b = c$, otherwise set $a = c$. At each step we get a new interval that contains a root of f and the generated sequence of points will eventually converge to the root.



Method of false position.

The algorithm:

Given a function f continuous on $[a_0, b_0]$, with $f(a_0)f(b_0) < 0$,

input: a_0, b_0

for $n = 0, 1, ..., ITMAX$

$c \leftarrow \frac{f(b_n)a_n - f(a_n)b_n}{f(b_n) - f(a_n)}$

if $f(a_n)f(c) < 0$, set $a_{n+1} = a_n, b_{n+1} = c$ **else** set $a_{n+1} = c, b_{n+1} = b_n$.

Stopping criterions: $|f(a_n)| \leq \varepsilon$ or $|a_n - a_{n-1}| \leq \varepsilon$, where ε is a specified tolerance value.

Remark 19 *The bisection and the false position methods converge at a very low speed compared to the secant method.*

Example 20 *The function $f(x) = x^3 - x^2 - 1$ has one zero in $[1, 2]$. Use the method of false position to approximate the zero of f with precision 10^{-4}.*

Sol. *A root lies in the interval $[1, 2]$ since $f(1) = -1$ and $f(2) = 3$. Starting with $a_0 = 1$ and $b_0 = 2$, we get using (11)*

$$c_0 = 2 - \frac{3(2 - 1)}{3 - (-1)} = 1.25 \textit{ and } f(c_0) = -0.609375.$$

Here, $f(c_0)$ has the same sign as $f(a_0)$ and so the root must lie on the interval $[c_0, b_0] = [1.25, 2]$. Next we set $a_1 = c_0$ and $b_1 = b_0$ to get the next approximation

$$c_1 = 2 - \frac{3 - (2 - 1.25)}{3 - (-0.609375)} = 1.37662337 \textit{ and } f(c_1) = -0.2862640.$$

Now $f(x)$ change sign on $[c_1, b_1] = [1.37662337, 2]$. Thus we set $a_2 = c_1$ and $b_2 = b_{1.}$ Continuing in this manner the iterations lead to the approximation 1.465558.

Example 21 *Compare the false position method, the secant method and Newton's method for solving the equation $x = cosx$, having as starting points $x_0 = 0.5$ și $x_1 = \pi/4$, respectively $x_0 = \pi/4$.*

n	**(a) x_n False position**	**(b) x_n Secant**	**(c) x_n Newton**
0	0.5	0.5	0.5
1	0.785398163397	0.785398163397	0.785398163397
2	0.736384138837	0.736384138837	0.739536133515
3	0.739058139214	0.739058139214	0.739085178106
4	0.739084863815	0.739085149337	0.739085133215
5	0.739085130527	0.739085133215	0.739085133215
6	0.739085133188	0.739085133215	
7	0.739085133215		

The extra condition from the false position method usually requires more computation than the secant method, and the simplifications from the secant method come with more iterations than in the case of Newton's method.

Example 22 *Consider the equation $x^2 - x - 3 = 0$. Give the next two iterations for approximating the solution of this equation using:*

a) Newton's method starting with $x_0 = 0$.

b) secant, false position and bisection methods starting with $x_0 = 0$ and $x_1 = 4$.