Numerical Analysis
An Advanced Course

Gheorghe Coman, Ioana Chiorean, Teodora Cătinaş

Contents

Preface

Numerical Analysis is a branch of Mathematics which deals with the constructive solutions of many problems formulated and studied by other fields of Mathematics. As illustrative examples, one may consider the problem of solving operatorial equations, the polynomial approximation of real-valued functions, the numerical integration of functions, and so on.

At the same time, Numerical Analysis has a theoretical aspect of its own. When analyzing a numerical method, one studies the error for the numerical solution, and obtains results concerning convergence and error bounds.

This book provides both theoretical and numerical aspects for basic problems of Numerical Analysis, with a special emphasis on optimal approximation.

It is addressed, first of all, to the undergraduate students, master students and Ph.D. students in Mathematics, but also to the students of Computer Science, Engineering, and to all the people interested in Numerical Analysis and Approximation Theory.

The book is structured in 7 chapters.

Chapter 1 is dedicated to some preliminary notions, linear spaces, Peano kernel theorems, best approximation and orthogonal polynomials.

Chapter 2 treats the functions approximation by univariate interpolation operators, respectively by linear and positive operators. Also, there are presented some elements of multivariate approximation.

Chapter 3 provides some applications to numerical integration of functions.

In Chapters 4-6 there are studied several methods of solving some operatorial equations (linear systems, nonlinear equations on R,nonlinear equations on R^n).

Chapter 7 is a brief introduction to parallel calculus.

The content of the book is essentially based on the authors work and research.

The authors

Chapter 1

Preliminaries

1.1 Linear spaces

Definition 1.1.1. *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)

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) $1 \cdot v = v, \quad \forall v \in V.$

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

Definition 1.1.2. *Let V be a K-linear space. A subset $V_1 \subset V$ is called* a subspace *of V if V_1 is stable with respect to the internal, respectively, to the external operations, i.e.:*

$$\forall v_1, v_2 \in V_1 \Longrightarrow v_1 + v_2 \in V_1,$$
$$\forall \alpha \in K, \ \forall v \in V_1 \Longrightarrow \alpha v \in V_1,$$

and V_1 is a K-linear space with respect to the induced operations.

Remark 1.1.3. 1) If V_1 is a subspace of V then

$$0 \in V_1$$

and

$$\forall v \in V_1 \Longrightarrow -v \in V_1;$$

2) For all linear spaces the subsets $\{0\}$ and V are subspaces;
3) If V_1 and V_2 are subspaces of V then

$$V_1 + V_2 = \{v_1 + v_2 \mid v_1 \in V_1, v_2 \in V_2\} \text{ is a subspace.}$$

Definition 1.1.4. *Let V and V' be two K-linear spaces. A function $f : V \to V'$ is called* linear transformation *or* linear operator *if:*

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

Remark 1.1.5. 1) $f : V \to V'$ is a linear operator if and only if

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

2) f linear operator implies

$$f(0) = 0 \quad \text{and} \quad f(-v) = -f(v), \quad v \in V.$$

3) If $f : V \to V'$ is a linear operator then

$$\operatorname{Ker} f = \{v \in V \mid f(v) = 0\}$$

is a subspace of V called *the Kernel* or *the null space* of f; and

$$\operatorname{Im} f := f(V) = \{f(v) \mid v \in V\}$$

is a subspace of V', called *the image* of f.

Remark 1.1.6. A $\mathbb{R}$-linear space ($K = \mathbb{R}$) is called *the real linear space* and a $\mathbb{C}$-linear space ($K = \mathbb{C}$) is called *the complex linear space.*

Definition 1.1.7. *If V is a K-linear space then an application $f : V \to K$ is called* functional *(real functional if $K = \mathbb{R}$, respectively complex functional if $K = \mathbb{C}$).*

Definition 1.1.8. *A nonnegative real functional p, defined on a linear space V, i.e.,*

$$p : V \to [0, \infty],$$

with the properties

$$\begin{aligned} &1)\ p(v_1 + v_2) \le p(v_1) + p(v_2), \quad \forall v_1, v_2 \in V \\ &2)\ p(\alpha v) = |\alpha|\, p(v), \quad \forall \alpha \in R \text{ and } v \in V \end{aligned}$$

is called a seminorm *on V. If, in addition,*

$$3)\ p(v) = 0 \Longrightarrow v = 0$$

then it is called a norm *and V is called* a normed linear space.

A norm is denoted by $\|\cdot\|$.

Remark 1.1.9. If V is a normed linear space then we have:
a) The properties:

$$\begin{aligned} &1) \text{ and } 3) \Longrightarrow \|v\| = 0 \Leftrightarrow v = 0, \\ &2) \Longrightarrow |\|v_1\| - \|v_2\|| \le \|v_1 - v_2\|; \end{aligned}$$

b) $d(v_1, v_2) = \|v_1 - v_2\|$, for $v_1, v_2 \in V$, is *a distance* in V.
c) A sequence $(v_n)_{n\in\mathbb{N}}$ of elements of V is said to be *convergent* in norm of V to an element $v \in V$, if

$$\lim_{n\to\infty} \|v_n - v\| = 0.$$

d) A sequence $(v_n)_{n\in\mathbb{N}}$ is called *a Cauchy sequence* if for any $\varepsilon > 0$ there exists a natural number $N = N_\varepsilon$ such that

$$\|v_m - v_n\| < \varepsilon, \quad \text{for } m, n > N_\varepsilon.$$

Remark 1.1.10. A normed linear space is *a metric space.*

Remark 1.1.11. A convergent sequence is a Cauchy sequence.

Definition 1.1.12. *In a normed linear space V, if any Cauchy sequence is convergent, then V is called* a complete space.

Definition 1.1.13. *A complete normed linear space is called* a Banach space.

Definition 1.1.14. *Let V be a K-linear space. An application*

$$\langle v_1, v_2\rangle : V \times V \to K,$$

with the properties:

$$\begin{aligned}
&1)\ \langle v_1, v_2\rangle = \overline{\langle v_2, v_1\rangle},\\
&2)\ \langle v_1 + v_2, v_3\rangle = \langle v_1, v_3\rangle + \langle v_2, v_3\rangle,\\
&3)\ \langle \alpha v_1, v_2\rangle = \alpha \langle v_1, v_2\rangle, \quad \alpha \in K\\
&4)\ v \neq 0 \Longrightarrow \langle v, v\rangle > 0,
\end{aligned}$$

is called the inner-product *of v_1 and v_2.*

Remark 1.1.15. The following statements are fulfilled:

$$\begin{aligned}
&1) \text{ and } 4) \Longrightarrow \langle v, v\rangle \text{ is real and strict positive;}\\
&1) \text{ and } 3) \Longrightarrow \langle 0, v\rangle = \langle v, 0\rangle = 0; \text{ in particular } \langle 0, 0\rangle = 0;\\
&1) \text{ and } 2) \Longrightarrow \langle v_1, v_2 + v_3\rangle = \overline{\langle v_2 + v_3, v_1\rangle} = \overline{\langle v_2, v_1\rangle} + \overline{\langle v_3, v_1\rangle}\\
&\qquad\qquad\qquad\qquad = \langle v_1, v_2\rangle + \langle v_1, v_3\rangle;\\
&1) \text{ and } 3) \Longrightarrow \langle v_1, \alpha v_2\rangle = \bar{\alpha} \langle v_1, v_2\rangle.
\end{aligned}$$

If $K = \mathbb{R}$ then

$$\langle \alpha v_1, v_2\rangle = \langle v_1, \alpha v_2\rangle = \alpha \langle v_1, v_2\rangle.$$

Definition 1.1.16. *A linear space with an inner-product is called* an inner-product space.

Definition 1.1.17. *In an inner-product space, the norm defined by*

$$\|v\| = \sqrt{\langle v, v\rangle},$$

is called the norm induced by the inner-product.

Definition 1.1.18. *A normed linear space with the norm induced by an inner-product is called* a pre-Hilbert space.

In any pre-Hilbert space, the following inequality (called the Schwarz inequality*) holds, with respect to the inner-product norm:*

$$|\langle v_1, v_2\rangle| \leq \|v_1\| \cdot \|v_2\|.$$

Also, the inner-product norm satisfies the triangle inequality*:*

$$\|v_1 + v_2\| \leq \|v_1\| + \|v_2\|.$$

Definition 1.1.19. *A complete pre-Hilbert space is called* a Hilbert space.

A Banach space with the norm induced by an inner-product is a Hilbert space.

In any Hilbert space V the following equality holds:

$$\|v_1 + v_2\|^2 + \|v_1 - v_2\|^2 = 2(\|v_1\|^2 + \|v_2\|^2), \quad \forall v_1, v_2 \in V, \tag{1.1.1}$$

that is called *the parallelogram identity* or *parallelogram law*.

Moreover, a Banach space whose norm satisfies the parallelogram identity is a Hilbert space.

Definition 1.1.20. *Two elements v_1, v_2 in an inner-product space V are* orthogonal *if*

$$\langle v_1, v_2 \rangle = 0.$$

A set $V' \subset V$ is called an orthogonal set *if*

$$\langle v_1, v_2 \rangle = 0, \quad \forall v_1, v_2 \in V', \quad v_1 \neq v_2.$$

If, in addition,

$$\langle v, v \rangle = 1, \quad \forall v \in V',$$

then V' is called an orthonormal set.

Definition 1.1.21. *Let V be a linear space over $\mathbb{R}$ or $\mathbb{C}$. A linear operator $P : V \to V$ is called* a projector *if*

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

For example, *the identity operator*

$$I : V \to V, \quad I(v) = v$$

and *the null operator*

$$0 : V \to V, \quad 0(v) = 0$$

are projectors.

Definition 1.1.22. *Let $P_1, P_2 : V \to V$ be projectors. If*

$$P_1 \circ P_2 = P_2 \circ P_1$$

then P_1 and P_2 are called commuting projectors.

Remark 1.1.23. 1) " $\circ$ " is also denoted by " $\cdot$ " or it is just missing, i.e., $P_1 \circ P_2$ is denoted by $P_1 \cdot P_2$ or P_1P_2.

2) P is a projector implies that $P^C := I - P$ is a projector.

Indeed,

$$(P^C)^2 = (I-P)(I-P) = I^2 - P - P + P^2 = I - P = P^C.$$

3) The projectors P and P^C are commuting projectors.

We have

$$PP^C = P(I-P) = PI - P = P - P = 0,$$
$$P^CP = (I-P)P = IP - P^2 = P - P = 0,$$

hence,

$$PP^C = 0 = P^CP.$$

4) $(P^C)^C = P$.

5) P is projector implies $\operatorname{Im} P^C = \operatorname{Ker} P$.

6) P and Q are commuting projectors implies PQ is projector.

Indeed,

$$(PQ)^2 = PQPQ = PPQQ = P^2Q^2 = PQ.$$

7) P, Q are commuting projectors implies P^C, Q^C are commuting projectors.

We have

$$P^CQ^C = (I-P)(I-Q) = I^2 - IQ - PI + PQ = I - Q - P + PQ,$$
$$Q^CP^C = (I-Q)(I-P) = I^2 - IP - QI + PQ = I - P - Q + PQ.$$

It follows that

$$P^CQ^C = Q^CP^C.$$

8) P, Q are commuting projectors implies P^CQ^C and $(P^CQ^C)^C$ are projectors, with

$$(P^CQ^C)^C = P + Q - PQ := P \oplus Q,$$

which is *the boolean sum* of P and Q.

Indeed, we already have seen that

$$P^CQ^C = I - P - Q + PQ,$$

so,

$$(P^CQ^C)^C = I - (I - P - Q + PQ) = P + Q - PQ.$$

Let V be a linear space over $\mathbb{R}$ or $\mathbb{C}$ and

$$\mathcal{L}= \{P_j \mid j \in J\}$$

be a set of commuting projectors of V, i.e., $P_j : V \to V$ are linear operators and

$$\begin{aligned} &P_j^2 = P_j; \\ &P_j P_k = P_k P_j, \quad \text{for } j,k \in J. \end{aligned}$$

Proposition 1.1.24. *The set*

$$\mathcal{L}' = \{P_j P_k \mid j,k \in J\}.$$

includes $\mathcal{L}$, *i.e.,*

$$\mathcal{L} \subseteq \mathcal{L}', \tag{1.1.2}$$

and it is made up by commuting projectors.

Proof. For all $j \in J$, we have $P_j^2 = P_j$, i.e.,

$$P_j P_j = P_j,$$

which implies $P_j \in \mathcal{L}'$, so (1.1.2) is proved.

For any $j,k \in J$ denote by

$$P_{jk} := P_j P_k,$$

which implies that P_{jk} is a projector, as a composition of two commuting projectors. For every $j,k,i,m \in J$ we have

$$\begin{aligned} P_{jk}P_{im} &= P_j P_k P_i P_m = P_j P_i P_k P_m = P_i P_j P_k P_m \\ &= P_i P_j P_m P_k = P_i P_m P_j P_k = P_{im} P_{jk}, \end{aligned}$$

i.e., the projectors from $\mathcal{L}'$ are commuting. ■

Proposition 1.1.25. *The set*

$$\mathcal{L}'' = \{P_{jk} \oplus P_{im} \mid j,k,i,m \in J\}$$

includes $\mathcal{L}'$, *i.e.,*

$$\mathcal{L}' \subseteq \mathcal{L}'', \tag{1.1.3}$$

and it is made up by commuting projectors.

Proof. For $\forall j,k \in J$, we have

$$\begin{aligned} P_{jk} \oplus P_{jk} &= P_{jk} + P_{jk} - P_{jk}\ P_{jk} = P_{jk} + P_{jk} - P_{jk}^2 \\ &= P_{jk} + P_{jk} - P_{jk} = P_{jk}, \end{aligned}$$

which implies $P_{jk} \in \mathcal{L}''$, so (1.1.3) is proved. From the permutability of the projectors from $\mathcal{L}'$, it follows that the elements of $\mathcal{L}''$ are projectors. For the commutability of the projectors of $\mathcal{L}''$ we have to prove that if P_i, $i = 1, ..., 4$ are commuting projectors then

$$(P_1 \oplus P_2)(P_3 \oplus P_4) = (P_3 \oplus P_4)(P_1 \oplus P_2). \tag{1.1.4}$$

Indeed, we get

$$\begin{aligned} (P_1 \oplus P_2)(P_3 \oplus P_4) =& (P_1 + P_2 - P_1P_2)(P_3 + P_4 - P_3P_4) \\ =& P_1P_3 + P_1P_4 - P_1P_3P_4 + P_2P_3 + P_2P_4 \\ & - P_2P_3P_4 - P_1P_2P_3 - P_1P_2P_4 + P_1P_2P_3P_4. \end{aligned}$$

On the other hand,

$$\begin{aligned} (P_3 \oplus P_4)(P_1 \oplus P_2) =& (P_3 + P_4 - P_3P_4)(P_1 + P_2 - P_1P_2) \\ =& P_3P_1 + P_3P_2 - P_3P_1P_2 + P_4P_1 + P_4P_2 \\ & - P_4P_1P_2 - P_3P_4P_1 - P_3P_4P_2 + P_3P_4P_1P_2. \end{aligned}$$

As, P_i, $i = 1, ..., 4$ are commuting projectors, we have

$$\begin{aligned} (P_3 \oplus P_4)(P_1 \oplus P_2) =& P_1P_3 + P_1P_4 - P_1P_3P_4 + P_2P_3 + P_2P_4 \\ & - P_2P_3P_4 - P_1P_2P_3 - P_1P_2P_4 + P_1P_2P_3P_4, \end{aligned}$$

and (1.1.4) is proved. ■

Remark 1.1.26. From the definitions of $\mathcal{L}'$ and $\mathcal{L}''$, it follows that if $\mathcal{L}$ is finite then $\mathcal{L}'$ and $\mathcal{L}''$ are also finite.

Starting from a set $\mathcal{L}$ of commuting projectors, we have built $\mathcal{L}'$ and $\mathcal{L}''$, which are sets of commuting projectors and

$$\mathcal{L} \subseteq \mathcal{L}' \subseteq \mathcal{L}''.$$

Denoting

$$\mathcal{L}_1 = \mathcal{L}''$$

and inductively defining

$$\mathcal{L}_{n+1} = \mathcal{L}_n'', \quad (n \geq 1),$$

one obtains an increasing sequence of sets of commuting projectors:

$$\mathcal{L}_1 \subseteq \mathcal{L}_2 \subseteq \ldots \subseteq \mathcal{L}_n \subseteq \mathcal{L}_{n+1} \subseteq \ldots. \tag{1.1.5}$$

Let us denote

$$\bar{\mathcal{L}} = \bigcup_{n=1}^{\infty} \mathcal{L}_n. \tag{1.1.6}$$

Proposition 1.1.27. *The set of commuting projectors $\bar{\mathcal{L}}$, is stable with respect to the composition and boolean sum, i.e.,*

$$\forall P, Q \in \bar{\mathcal{L}} \Longrightarrow PQ \in \bar{\mathcal{L}}$$
$$\forall P, Q \in \bar{\mathcal{L}} \Longrightarrow P \oplus Q \in \bar{\mathcal{L}}.$$

Proof. If $P, Q \in \bar{\mathcal{L}}$ then, from (1.1.6), if follows that $\exists n_1, n_2 \leq n \in \mathbb{N}^*$ such that $P \in \mathcal{L}_{n_1}$, $Q \in \mathcal{L}_{n_2}$. If $n_1 \leq n_2$, from (1.1.5) one obtains

$$\mathcal{L}_{n_1} \subseteq \mathcal{L}_{n_2} \Longrightarrow P, Q \in \mathcal{L}_{n_2} \Longrightarrow PQ \in \mathcal{L}_{n_2}' \leq \mathcal{L}_{n_2}'' \subseteq \mathcal{L}_{n_2+1}$$

and

$$P \oplus Q \in \mathcal{L}_{n_2}'' = \mathcal{L}_{n_2+1}.$$

So, $PQ = QP$ and PQ, $P \oplus Q \in \bar{\mathcal{L}}$. ■

Remark 1.1.28. The set $\bar{\mathcal{L}}$ is the lowest set of commuting projectors that includes $\mathcal{L}$ and which is stable with respect to the operations of composition and boolean sum.

Remark 1.1.29. For $P_1, P_2, P_3 \in \bar{\mathcal{L}}$, we have

$$P_1 P_2 = P_2 P_1,$$
$$P_1 \oplus P_2 = P_2 \oplus P_1,$$
$$(P_1 P_2) P_3 = P_1 (P_2 P_3),$$
$$(P_1 \oplus P_2) \oplus P_3 = P_1 \oplus (P_2 \oplus P_3),$$

which are easy to verify.

Proposition 1.1.30. *The relation* " $\leq$ ", *defined on* $\bar{\mathcal{L}}$ *by*

$$P_1 \leq P_2 \Leftrightarrow P_1P_2 = P_1,$$

is an order relation (reflexive, transitive and antisymmetric).

Proof. Reflexive: $\forall P \in \overline{\mathcal{L}}$, $PP = P$ (P is a projector), so $P \leq P$.
Transitive: $\forall P_1, P_2, P_3 \in \bar{\mathcal{L}}$ we have

$$P_1 \leq P_2,\ P_2 \leq P_3 \Longrightarrow P_1P_2 = P_1,\ P_2P_3 = P_2 \Longrightarrow P_1P_2P_2P_3 = P_1P_2$$
$$\Longrightarrow P_1P_2P_3 = P_1P_2 \Longrightarrow P_1P_3 = P_1 \Longrightarrow P_1 \leq P_3.$$

Antisymmetric: $\forall P_1, P_2$ we have

$$P_1 \leq P_2,\ P_2 \leq P_1 \Longrightarrow P_1P_2 = P_1,$$
$$P_2P_1 = P_2 \Longrightarrow P_1 = P_2 \ \ (P_1P_2 = P_2P_1)$$

■

Proposition 1.1.31. $(\overline{\mathcal{L}}, \leq)$ *is* a lattice *and*

$$\inf(P_1,P_2) = P_1P_2, \tag{1.1.7}$$
$$\sup(P_1, P_2) = P_1 \oplus P_2, \qquad \forall P_1, P_2 \in \overline{\mathcal{L}}. \tag{1.1.8}$$

Proof. We have

$$\left.\begin{array}{l} P_1(P_1P_2) = P_1^2P_2 = P_1P_2 \Longrightarrow P_1P_2 \leq P_1 \\ P_2(P_1P_2) = P_1P_2^2 = P_1P_2 \Longrightarrow P_1P_2 \leq P_2 \end{array}\right\} \Longrightarrow P_1P_2 \text{ is a minor for } P_1 \text{ and } P_2.$$

Let $P_3 \in \overline{\mathcal{L}}$ be another minor of P_1 and P_2. It follows

$$P_3 \leq P_1,\ P_3 \leq P_2 \Longrightarrow P_3P_1 = P_3,\ P_3P_2 = P_3 \Longrightarrow (P_3P_1)(P_3P_2) = P_3^2$$
$$\Longrightarrow P_3^2(P_1P_2) = P_3^2 \Longrightarrow P_3(P_1P_2) = P_3 \Longrightarrow P_3 \leq P_1P_2.$$

Hence, P_1P_2 is the highest minor of P_1 and P_2, what implies that

$$\inf(P_1, P_2) = P_1P_2$$

and (1.1.7) is proved.

To prove (1.1.8), we consider

$$\begin{aligned} P_1(P_1 \oplus P_2) &= P_1(P_1 + P_2 - P_1P_2) = P_1^2 + P_1P_2 - P_1^2P_2 \\ &= P_1 + P_1P_2 - P_1P_2 = P_1 \Longrightarrow P_1 \leq P_1 \oplus P_2. \end{aligned}$$

In the same way it can be proved that

$$P_2 \leq P_1 \oplus P_2.$$

So,

$$\left.\begin{array}{l} P_1 \leq P_1 \oplus P_2 \\ P_2 \leq P_1 \oplus P_2 \end{array}\right\} \Longrightarrow P_1 \oplus P_2 \text{ is a major for } P_1 \text{ and } P_2.$$

Now, let $P_3 \in \overline{\mathcal{L}}$ be another major of P_1 and P_2. We have

$$\begin{aligned} P_1 \leq P_3 &\Longrightarrow P_1P_3 = P_1, \\ P_2 \leq P_3 &\Longrightarrow P_2P_3 = P_2, \end{aligned}$$

respectively,

$$\begin{aligned} (P_1 \oplus P_2)P_3 &= (P_1 + P_2 - P_1P_2)P_3 \\ &= P_1P_3 + P_2P_3 - P_1P_2P_3 \\ &= P_1 + P_2 - P_1P_2 = P_1 \oplus P_2. \end{aligned}$$

Therefore,

$$(P_1 \oplus P_2)P_3 = P_1 \oplus P_2 \Longrightarrow P_1 \oplus P_2 \leq P_3,$$

i.e., $P_1 \oplus P_2$ is the lowest major of P_1 and P_2, and (1.1.8) is also proved.

In concusion, $(\overline{\mathcal{L}}, \leq)$ is a lattice. ■

Proposition 1.1.32. *The lattice $(\overline{\mathcal{L}}, \leq)$ is distributive, i.e.,*

$$P_1(P_2 \oplus P_3) = (P_1P_2) \oplus (P_1P_3), \quad \forall P_1, P_2, P_3 \in \overline{\mathcal{L}}. \tag{1.1.9}$$

Proof. We have that

$$\begin{aligned} &P_1(P_2 \oplus P_3) = P_1(P_2 + P_3 - P_2P_3) = P_1P_2 + P_1P_3 - P_1P_2P_3, \\ &(P_1P_2) \oplus (P_1P_3) = P_1P_2 + P_1P_3 - P_1P_2P_1P_3 = P_1P_2 + P_1P_3 - P_1P_2P_3, \end{aligned}$$

and therefore, the distributivity is proved. ■

Remark 1.1.33. The relation (1.1.9) represents the distributivity of inf with respect to sup. In any lattice this distributivity is equivalent to the distributivity of sup with respect to inf. So, (1.1.9) is equivalent to

$$P_1 \oplus (P_2 P_3) = (P_1 \oplus P_2)(P_1 \oplus P_3), \quad \forall P_1, P_2, P_3 \in \overline{\mathcal{L}}.$$

Remark 1.1.34. The lattice $(\overline{\mathcal{L}}, \leq)$ is generated by $\mathcal{L}$.

Definition 1.1.35. *The elements of the set* $\mathcal{L}$ *are called* the generators *of the lattice* $(\overline{\mathcal{L}}, \leq)$.

1.2 Examples of functions spaces

1. The space $C^p[a,b]$, $p \in \mathbb{N}$ of functions $f : [a,b] \to \mathbb{R}$ continuously differentiable up to order p, inclusively. For $p = 0$ one gets the space $C[a,b]$ of continuous functions on $[a,b]$.

$C^p[a,b]$ is a linear space over $\mathbb{R}$, with respect to the usual addition of functions and multiplication of functions by real numbers. It is also normed, with the norm defined by

$$\|f\| = \sum_{k=0}^{p} \max_{a \leq x \leq b} \left| f^{(k)}(x) \right|.$$

Moreover, $C^p[a,b]$ is a complete space, thus it is a Banach space.

2. The space $\mathbb{P}_m$ of polynomials of degree at most m. One has $\mathbb{P}_m \subset C^\infty[a,b]$.

We denote by $\mathbb{P}_m^n$ the set of polynomials in n variables and of global degree at most m and by $\mathbb{P}_{m_1,...,m_n}^n$ the set of polynomials in n variables $x_1, ..., x_n$ and of degree m_k with respect to the variable x_k, for $k = 1, ..., n$.

3. The space $L^p[a,b]$, $p \in \mathbb{R}$, $p \geq 1$ of functions p-Lebesgue integrable on $[a,b]$.

Recall that a function $f : [a,b] \to \mathbb{R}$ is said to be p-Lebesgue integrable on $[a,b]$ if $|f|^p$ is Lebesgue integrable on $[a,b]$.

Moreover, the class $\hat{f}$ of p-Lebesgue integrable functions on $[a,b]$ is the set of all functions g, which are p-Lebesgue integrable on $[a,b]$ and equivalent to f (denoted $g \sim f$), i.e., $g = f$ almost everywhere on $[a,b]$.

The set of equivalence classes defined above is a linear space with respect to addition:

$$\hat{f}_1 + \hat{f}_2 = \hat{g}, \text{ iff } f_1 + f_2 \sim g,$$

and multiplication:

$$\lambda \hat{f} = \hat{h}, \text{ iff } \lambda f \sim h.$$

Identifying the class $\hat{f}$ with its representant $f \in \hat{f}$, we define the norm

$$\|f\| = \left(\int_a^b |f(x)|^p \, dx \right)^{1/p}. \tag{1.2.1}$$

The space $L^p[a,b]$ is the linear space of the introduced equivalence classes, and it is normed, with the norm given by (1.2.1).

Moreover, $L^p[a,b]$ is a complete space, thus it is a Banach space.

For $p = 2$, i.e., on $L^2[a,b]$ one can define the inner product

$$\langle f, g \rangle_2 = \int_a^b f(x) g(x) dx.$$

It follows that $L^2[a,b]$ is a Hilbert space, with respect to the norm endowed by this inner product, namely,

$$\|f\|_2 = \left(\int_a^b f^2(x) dx \right)^{1/2}.$$

4. Let w be a positive integrable function on $[a,b]$, with

$$\int_a^b w(x) dx < \infty.$$

Let $L^2_w[a,b]$ be the set of functions f such that

$$\int_a^b w(x) |f(x)|^2 dx < \infty.$$

On $L^2_w[a,b]$ it is defined the inner-product

$$\langle f, g \rangle_{w,2} = \int_a^b w(x) f(x) g(x) dx, \qquad f, g \in L^2_w[a,b],$$

and the induced norm

$$\|f\|_{w,2} = \left(\int_a^b w(x) |f(x)|^2 dx \right)^{1/2}.$$

$\left(L^2_w[a,b], \|\cdot\|_{w,2}\right)$ is a Hilbert space.

5. The space $H^m[a,b]$, $m \in \mathbb{N}^*$ of functions $f \in C^{m-1}[a,b]$, with $f^{(m-1)}$ absolutely continuous on $[a,b]$.

Notice that every function $f \in H^m[a,b]$ can be represented using Taylor's formula with remainder in the integral form:

$$f(x) = \sum_{k=0}^{m-1} \frac{(x-a)^k}{k!} f^{(k)}(a) + \int_a^x \frac{(x-t)^{m-1}}{(m-1)!} f^{(m)}(t)dt. \tag{1.2.2}$$

So, it immediately follows that for $f, g \in H^m[a,b]$ and $\lambda \in \mathbb{R}$, one has $f+g$, $\lambda f \in H^m[a,b]$.

$H^m[a,b]$ is a linear space.

6. The space $H^{m,2}[a,b]$, $m \in \mathbb{N}^*$ of functions $f \in H^m[a,b]$, with $f^{(m)} \in L^2[a,b]$.

From (1.2.2) it follows that a function $f \in H^{m,2}[a,b]$ is uniquely determined by the values $f^{(k)}(a)$, $k = 0, ..., m-1$ and $f^{(m)}$. Consequently, one can consider the expression

$$\langle f, g \rangle_{m,2} = \int_a^b f^{(m)}(x) g^{(m)}(x)dx + \sum_{k=0}^{m-1} f^{(k)}(a) g^{(k)}(a),$$

as an inner product, since the four properties of the inner product are verified in this case.

Let $\|\cdot\|_{m,2}$ be the norm endowed by this inner product, namely,

$$\|f\|^2_{m,2} = \left\|f^{(m)}\right\|^2_2 + \sum_{k=0}^{m-1} (f^{(k)}(a))^2.$$

In $H^{m,2}[a,b]$ we consider a Cauchy sequence $\{f_n\}_{n\in\mathbb{N}}$. Since $\{f_n^{(m)}\}_{n\in\mathbb{N}}$ is a Cauchy sequence in $L^2[a,b]$, and $\{f_n^{(k)}(a)\}_{n\in\mathbb{N}}$ is a Cauchy sequence in $\mathbb{R}$, for each $k = 0, ..., m-1$, it follows that $\{f_n^{(m)}\}_{n\in\mathbb{N}}$ converges to a function $g \in L^2[a,b]$, and every sequence $\{f_n^{(k)}(a)\}_{n\in\mathbb{N}}$ converges to a number $c_k \in \mathbb{R}$, $k = 0, ..., m-1$, ($L^2[a,b]$ and $\mathbb{R}$ are complete spaces). Let

$$f(x) = \sum_{k=0}^{m-1} c_k \frac{(x-a)^k}{k!} + \int_a^x \frac{(x-t)^{m-1}}{(m-1)!} g(t)dt.$$

Since $f \in H^m[a,b]$ and $f^{(m)} = g$ ($f^{(m)} \in L^2[a,b]$), it follows that $f \in H^{m,2}[a,b]$. But,

$$\lim_{n\to\infty} \|f_n - f\|_{m,2}^2 = \lim_{n\to\infty} \left(\left\| f_n^{(m)} - f^{(m)} \right\|_2^2 + \sum_{k=0}^{m} [f_n^{(k)}(a) - f^{(k)}(a)]^2 \right) = 0,$$

so,

$$\lim_{n\to\infty} f_n = f,$$

in $\|\cdot\|_{m,2}$ norm.

It yields $H^{m,2}[a,b]$ is complete, thus it is a Hilbert space.

7. The Sard-type space $B_{pq}(a,c)$, $(p,q \in \mathbb{N},\ p+q=m)$ of the functions $f : D \to \mathbb{R}$, $D = [a,b] \times [c,d]$ satisfying:

1. $f^{(p,q)} \in C(D)$;
2. $f^{(m-j,j)}(\cdot,c) \in C[a,b]$, $j < q$;
3. $f^{(i,m-i)}(a,\cdot) \in C[c,d]$, $i < p$.

8. The Sard-type space $B_{pq}^r(a,c)$, $(p,q \in \mathbb{N},\ p+q=m,\ r \in \mathbb{R},\ r \geq 1)$ of the functions $f : D \to \mathbb{R}$, $D = [a,b] \times [c,d]$ satisfying:

1. $f^{(i,j)} \in C(D)$, $i < p$, $j < q$;
2. $f^{(m-j-1,j)}(\cdot,c)$ is absolute continuous on $[a,b]$ and $f^{(m-j,j)}(\cdot,c) \in L^r[a,b]$, $j < q$;
3. $f^{(i,m-i-1)}(a,\cdot)$ is absolute continuous on $[c,d]$ and $f^{(i,m-i)}(a,\cdot) \in L^r[c,d]$, $i < p$;
4. $f^{(p,q)} \in L^r(D)$.

As in the univariate case, we have that if $f \in B_{pq}(a,c)$ then this function admits the Taylor representation:

$$f(x,y) = \sum_{i+j<m} \frac{(x-a)^i}{i!} \frac{(y-c)^j}{j!} f^{(i,j)}(a,c) + (R_m f)(x,y), \tag{1.2.3}$$

with

$$\begin{aligned}(R_m f)(x,y) =& \sum_{j<q} \frac{(y-c)^j}{j!} \int_a^b \frac{(x-s)_+^{m-j-1}}{(m-j-1)!} f^{(m-j,j)}(s,c)ds \\ &+ \sum_{i<p} \frac{(x-a)^i}{i!} \int_c^d \frac{(y-t)_+^{m-i-1}}{(m-i-1)!} f^{(i,m-i)}(a,t)dt \\ &+ \iint_D \frac{(x-s)_+^{p-1}}{(p-1)!} \frac{(y-t)_+^{q-1}}{(q-1)!} f^{(p,q)}(s,t)dsdt,\end{aligned}$$

where

$$z_+ = \begin{cases} z, \text{ when } z \geq 0, \\ 0, \text{ when } z < 0. \end{cases}$$

Remark 1.2.1. The Taylor type formula (1.2.3) holds for any domain Ω having the property that there exists a point $(a, c) \in \Omega$ such that the rectangle $[a, x] \times [c, y] \subseteq \Omega$, for all $(x, y) \in \Omega$. In this case, formula (1.2.3) is written as

$$f(x,y) = \sum_{i+j<m} \frac{(x-a)^i}{i!} \frac{(y-c)^j}{j!} f^{(i,j)}(a,c) + (R_m f)(x,y),$$

with

$$\begin{aligned}(R_m f)(x,y) =& \sum_{j<q} \frac{(y-c)^j}{j!} \int_{I_1} \frac{(x-s)_+^{m-j-1}}{(m-j-1)!} f^{(m-j,j)}(s,c)ds \\ &+ \sum_{i<p} \frac{(x-a)^i}{i!} \int_{I_2} \frac{(y-t)_+^{m-i-1}}{(m-i-1)!} f^{(i,m-i)}(a,t)dt \\ &+ \iint_\Omega \frac{(x-s)_+^{p-1}}{(p-1)!} \frac{(y-t)_+^{q-1}}{(q-1)!} f^{(p,q)}(s,t)dsdt,\end{aligned}$$

where

$$\begin{aligned} I_1 &= \{(x,y) \in \mathbb{R}^2 \mid y = c\} \cap \Omega, \\ I_2 &= \{(x,y) \in \mathbb{R}^2 \mid x = a\} \cap \Omega. \end{aligned}$$

Examples of such domains are the triangle

$$T_h = \{(x,y) \in \mathbb{R}^2 \mid x \geq 0,\ y \geq 0,\ x + y \leq h\}$$

with $(a, c) = (0, 0)$ and any circle with center at (a, c).

1.3 The Peano Kernel Theorems

Theorem 1.3.1. *Let $L : H^m[a, b] \to \mathbb{R}$ be a linear functional which commutes with the defined integral operator, for example, of the form*

$$L(f) = \sum_{i=0}^{m-1} \int_a^b f^{(i)}(x) d\mu_i(x),$$

where μ_i are functions of bounded variation on $[a, b]$.

If $\operatorname{Ker} L = \mathbb{P}_{m-1}$ *then*

$$L(f) = \int_a^b K_m(t) f^{(m)}(t) dt, \tag{1.3.1}$$

where

$$K_m(t) = L^x \left(\frac{(x-t)_+^{m-1}}{(m-1)!} \right).$$

The notation $L^x(f)$ *means that* L *is applied to* f *with respect to the variable* x*.*

Proof. Since $f \in H^m[a,b]$, one can apply Taylor's formula and get

$$f = P_{m-1} + R_{m-1},$$

where $P_{m-1} \in \mathbb{P}_{m-1}$ and

$$R_{m-1}(x) = \int_a^b \frac{(x-t)_+^{m-1}}{(m-1)!} f^{(m)}(t) dt.$$

Thus,

$$L(f) = L(P_{m-1}) + L(R_{m-1}).$$

Since $L(P_{m-1}) = 0$ ($\operatorname{Ker} L = \mathbb{P}_{m-1}$), we obtain

$$L(f) = L^x \left(\int_a^b \frac{(x-t)_+^{m-1}}{(m-1)!} f^{(m)}(t) dt \right),$$

and, from the hypothesis condition, we have that

$$L(f) = \int_a^b L \left(\frac{(\cdot - t)_+^{m-1}}{(m-1)!} \right) f^{(m)}(t) dt.$$

■

The function K is called *the Peano's kernel.*

Corollary 1.3.2. *If the kernel* K *does not change its sign on the interval* $[a,b]$ *and* $f^{(m)}$ *is continuous on* $[a,b]$*, then*

$$L(f) = \tfrac{1}{m!} L(e_m) f^{(m)}(\xi), \quad a \le \xi \le b, \tag{1.3.2}$$

where $e_k(x) = x^k$*.*

Proof. Applying mean value theorem to (1.3.1), one obtains the representation

$$L(f) = f^{(m)}(\xi) \int_a^b K_m(t)dt, \tag{1.3.3}$$

which does not depend on f. Taking $f = e_m$, this yields

$$\int_a^b K_m(t)dt = \tfrac{1}{m!} L(e_m).$$

Substituting the integral in (1.3.3) by the above expression, one gets (1.3.2). ■

Theorem 1.3.3. *Let $f \in B_{pq}(a, c)$ and the linear functional L given by*

$$\begin{aligned} L(f) = &\sum_{i+j<m} \iint_D f^{(i,j)}(x, y)d\mu_{i,j}(x, y) \\ &+ \sum_{j<q} \int_a^b f^{(m-j,j)}(x, c)d\mu_{m-j,j}(x) \\ &+ \sum_{i<p} \int_c^d f^{(i,m-i)}(a, y)d\mu_{i,m-i}(y), \end{aligned} \tag{1.3.4}$$

where $\mu_{m,n}$ are functions with bounded variation on D, $[a, b]$ and, respectively, on $[c, d]$. If $\operatorname{Ker} L = \mathbb{P}^2_{m-1}$ then

$$\begin{aligned} L(f) = &\sum_{j<q} \int_a^b K_{m-j,j}(s) f^{(m-j,j)}(s, c)ds \\ &+ \sum_{i<p} \int_c^d K_{i,m-i}(t) f^{(i,m-i)}(a, t)dt \\ &+ \iint_D K_{p,q}(s, t) f^{(p,q)}(s, t)dsdt, \end{aligned} \tag{1.3.5}$$

where

$$\begin{aligned} K_{m-j,j}(s) &= L^{x,y}\Big(\tfrac{(x-s)_+^{m-j-1}}{(m-j-1)!} \tfrac{(y-c)^j}{j!}\Big), \\ K_{i,m-i}(t) &= L^{x,y}\Big(\tfrac{(x-a)^i}{i!} \tfrac{(y-t)_+^{m-i-1}}{(m-i-1)!}\Big), \\ K_{p,q}(s, t) &= L^{x,y}\Big(\tfrac{(x-s)_+^{p-1}}{(p-1)!} \tfrac{(y-t)_+^{q-1}}{(q-1)!}\Big). \end{aligned}$$

Proof. Since $f \in B_{pq}(a,c)$, we can consider formula (1.2.3). Applying the functional L to each member of this formula and taking into account the hypothesis

$$L(p) = 0, \qquad \forall p \in \mathbb{P}^2_{m-1},$$

we are led to

$$L(f) = L(R_m f).$$

Next, from (1.3.4) we get (1.3.5). ■

1.4 Best approximation in Hilbert spaces

Definition 1.4.1. *Let $\mathcal{B}$ be a normed linear space $\mathcal{A} \subset \mathcal{B}$ and $f \in \mathcal{B}$. The element $g^* \in \mathcal{A}$ with the property that*

$$\|f - g^*\| \leq \|f - g\|, \qquad \forall g \in \mathcal{A}$$

is called the best approximation *of f from $\mathcal{A}$.*

Next, we are interested in some conditions for the existence, uniqueness and for a characterization of the best approximation, in the case when $\mathcal{B}$ is a Hilbert space.

Theorem 1.4.2. *Let $\mathcal{B}$ be a Hilbert space. If $\mathcal{A} \subset \mathcal{B}$ is a nonempty, convex and closed set then for all $f \in \mathcal{B}$ there exists an unique best approximation $g^* \in \mathcal{A}$.*

Proof. If $f \in \mathcal{A}$ then $g^* = f$ and the proof is over.

Suppose that $f \notin \mathcal{A}$. Let

$$d^* = d(f, \mathcal{A}) = \inf_{g \in \mathcal{A}} \|f - g\|.$$

Let $(g_n)_{n \in \mathbb{N}} \subset \mathcal{A}$ be such that

$$\lim_{n \to \infty} \|f - g_n\| = d^*.$$

Using the parallelogram identity (1.1.1), with $v_1 = f - g_n$ and $v_2 = f - g_m$, one obtains:

$$\|2f - (g_m + g_n)\|^2 + \|g_m - g_n\|^2 = 2\left(\|f - g_m\|^2 + \|f - g_n\|^2\right),$$

or

$$\|g_m - g_n\|^2 = 2\left(\|f - g_m\|^2 + \|f - g_n\|^2\right) - 4\left\|f - \tfrac{(g_m+g_n)}{2}\right\|^2.$$

Since $\mathcal{A}$ is a convex set and $g_m, g_n \in \mathcal{A}$, it follows that

$$\tfrac{1}{2}(g_m + g_n) \in \mathcal{A},$$

which implies

$$\left\|f - \tfrac{(g_m+g_n)}{2}\right\|^2 \geq (d^*)^2.$$

So,

$$\|g_m - g_n\|^2 \leq 2\left(\|f - g_m\|^2 + \|f - g_n\|^2\right) - 4(d^*)^2.$$

As,

$$\lim_{m\to\infty} \|f - g_m\| = d^*,$$
$$\lim_{n\to\infty} \|f - g_n\| = d^*,$$

it follows that $(g_n)_{n\in\mathbb{N}}$ is a Cauchy sequence, hence it is a convergent one, in $\mathcal{B}$. But $\mathcal{A}$ is closed, so $g^* \in \mathcal{A}$ and $d^* = \|f - g^*\|$. Hence, g^* is the best approximation to f and the existence is proved.

Suppose now, that there exist two elements $g^*, h^* \in \mathcal{A}$ such that

$$\|f - g^*\| = d^*$$

and

$$\|f - h^*\| = d^*.$$

Since $\mathcal{A}$ is a convex set and $g^*, h^* \in \mathcal{A}$, it follows

$$\tfrac{1}{2}(g^* + h^*) \in \mathcal{A},$$

hence,

$$\|2f - (g^* + h^*)\| \geq 2d^*.$$

Using again the parallelogram identity, with $v_1 = f - g^*$, $v_2 = f - h^*$, one obtains

$$\begin{aligned}\|g^* - h^*\|^2 &= 2\left(\|f - g^*\|^2 + \|f - h^*\|^2\right) - \|2f - (g^* + h^*)\|^2 \\ &\leq 4d^* - 4d^* = 0.\end{aligned}$$

Therefore,

$$\|g^* - h^*\| = 0,$$

i.e.,

$$g^* = h^*.$$

■

Theorem 1.4.3. *In the conditions of Theorem 1.4.2, $g^* \in \mathcal{A}$ is the best approximation to $f \in \mathcal{B}$ if and only if*

$$\langle g^* - f, g - g^* \rangle \geq 0, \qquad \forall g \in \mathcal{A}. \tag{1.4.1}$$

Proof. Suppose that g^* is the best approximation to f, i.e.,

$$\|g^* - f\| \leq \|g - f\|, \qquad \forall g \in \mathcal{A}.$$

The set $\mathcal{A}$ being convex, it follows that $\forall \varepsilon > 0$, $g^* + \varepsilon(g - g^*) \in \mathcal{A}$, hence,

$$\|g^* - f\| \leq \|g^* + \varepsilon(g - g^*) - f\|.$$

It yields

$$\begin{aligned} \|g^* - f\|^2 &\leq \|g^* - f + \varepsilon(g - g^*)\|^2 \\ &= \langle g^* - f + \varepsilon(g - g^*), g^* - f + \varepsilon(g - g^*) \rangle \\ &= \|g^* - f\|^2 + 2\varepsilon \langle g^* - f, g - g^* \rangle + \varepsilon^2 \|g - g^*\|^2, \end{aligned}$$

which implies

$$2 \langle g^* - f, g - g^* \rangle + \varepsilon \|g - g^*\|^2 \geq 0.$$

As this inequality holds for $\forall \varepsilon > 0$, we must have

$$\langle g^* - f, g - g^* \rangle \geq 0.$$

Now, let us suppose that (1.4.1) holds. For $g \in \mathcal{A}$, we have

$$\begin{aligned} \|g - f\|^2 &= \|g^* - f + (g - g^*)\|^2 \\ &= \|g^* - f\|^2 + \|g - g^*\|^2 + 2 \langle g^* - f, g - g^* \rangle, \end{aligned}$$

or

$$\|g - f\|^2 - \|g^* - f\|^2 = \|g - g^*\|^2 + 2 \langle g^* - f, g - g^* \rangle.$$

As (1.4.1) holds, it follows

$$\|g - f\|^2 - \|g^* - f\|^2 \geq 0,$$

i.e.,

$$\|g - f\| \geq \|g^* - f\|, \qquad \forall g \in \mathcal{A},$$

and the theorem is completely proved. ■

Remark 1.4.4. The property (1.4.1) is called the property of *pseudoorthogonality.*

Corollary 1.4.5. *If $\mathcal{A}$ is a linear subspace of $\mathcal{B}$ and $\mathcal{A}_t$ is a translation of $\mathcal{A}$ then $g^* \in \mathcal{A}_t$ is the best approximation for $f \in \mathcal{B}$, if and only if*

$$\langle g^* - f, g \rangle = 0, \qquad \forall g \in \mathcal{A}. \tag{1.4.2}$$

Proof. According to Theorem 1.4.3, $g^* \in \mathcal{A}_t$ is the best approximation for $f \in \mathcal{B}$, if and only if

$$\langle g^* - f, g - g^* \rangle \geq 0, \qquad \forall g \in \mathcal{A}_t.$$

As, $h := g - g^* \in \mathcal{A}$, it follows

$$\langle g^* - f, h \rangle \geq 0, \qquad h \in \mathcal{A}.$$

Suppose that $\langle g^* - f, h \rangle > 0$. We have $h \in \mathcal{A}$ implies $-h \in \mathcal{A}$ ($\mathcal{A}$ being a linear space), so

$$\langle g^* - f, -h \rangle \geq 0, \qquad h \in \mathcal{A}.$$

But, if

$$\langle g^* - f, h \rangle > 0$$

then

$$\langle g^* - f, -h \rangle < 0,$$

which is a contradiction with our assumption, therefore it implies (1.4.2). ■

1.5 Orthogonal polynomials

Let $\mathcal{P} \subset L^2_w[a,b]$ be a set of orthogonal polynomials on $[a,b]$, with respect to the weight function w, and $\tilde{\mathcal{P}}_n \subset \mathcal{P}$ the set of orthogonal polynomials of degree n, with the coefficient of x^n equals to 1 (such a polynomial is denoted by $\tilde{p}_n$).

Theorem 1.5.1. *If $\tilde{p}_n \perp \mathbb{P}_{n-1}$ then $\tilde{p}_n$ has exactly n real and distinct roots in the open interval (a,b).*

Proof. The condition $\tilde{p}_n \perp \mathbb{P}_{n-1}$ is equivalent with

$$\int_a^b w(x)\tilde{p}_n(x)x^i dx = 0, \qquad i = 0,1,...,n-1,$$

hence,

$$\int_a^b w(x)\tilde{p}_n(x)dx = 0.$$

It follows that $\tilde{p}_n$ has $m \geq 1$ roots in which it changes the sign. Let $x_1,...,x_m \in (a,b)$ be these roots and

$$r_m(x) = (x-x_1)...(x-x_m).$$

Therefore, $\tilde{p}_n r_m \geq 0$ on $[a,b]$. As $\tilde{p}_n r_m \neq 0$, we have

$$\langle \tilde{p}_n, r_m \rangle_{w,2} = \int_a^b w(x)\tilde{p}_n(x)r_m(x)dx \geq 0$$

and the condition $\tilde{p}_n \perp \mathbb{P}_{m-1}$ implies $m \geq n$. As a polynomial of degree n cannot have more than n roots, it follows that $m = n$. ■

Theorem 1.5.2. *Let $(p_n)_{n\in\mathbb{N}}$ be a sequence of orthogonal polynomials on $[a,b]$, with respect to the weight function w, and*

$$p_n(x) = a_n x^n + b_n x^{n-1} + ...$$

Then we have the recurrence relation:

$$\frac{a_n}{a_{n+1}}p_{n+1}(x) = \left(\frac{b_{n+1}}{a_{n+1}} - \frac{b_n}{a_n} + x\right)p_n(x) - \frac{a_{n-1}}{a_n}\frac{\|p_n\|^2_{w,2}}{\|p_{n-1}\|^2_{w,2}}p_{n-1}(x), \quad (1.5.1)$$

for $n = 1,2,...$

Proof. We have

$$xp_n(x) = \sum_{k=0}^{n+1} c_{n,k} p_k(x), \tag{1.5.2}$$

where

$$c_{n,k} = \int_a^b w(x) x p_n(x) p_k(x) dx \Big/ \int_a^b w(x) p_k^2(x) dx,$$

or

$$c_{n,k} = \langle e_1 p_n, p_k \rangle_{w,2} \Big/ \|p_k\|_{w,2}^2 , \qquad (e_1(x) = x).$$

We also have

$$c_{n,k} = \langle p_n, e_1 p_k \rangle_{w,2} \Big/ \|p_k\|_{w,2}^2 .$$

As, $\langle p_n, p_i \rangle_{w,2} = 0$, for $i < n$, it follows that $c_{n,k} = 0$, for $k < n-1$ and (1.5.2) becomes

$$xp_n(x) = c_{n,n+1} p_{n+1}(x) + c_{n,n} p_n(x) + c_{n,n-1} p_{n-1}(x). \tag{1.5.3}$$

In (1.5.3), identifying the terms of degree $n+1$, respectively n, one obtains

$$c_{n,n+1} = \frac{a_n}{a_{n+1}},$$
$$c_{n,n} = \frac{b_n}{a_n} - \frac{b_{n+1}}{a_{n+1}}.$$

In order to get $c_{n,n-1}$, one notices that

$$c_{n,k} = \langle e_1 p_n, p_k \rangle_{w,2} \Big/ \|p_k\|_{w,2}^2 ,$$

respectively,

$$c_{k,n} = \langle e_1 p_k, p_n \rangle_{w,2} \Big/ \|p_n\|_{w,2}^2 .$$

So,

$$c_{n,k} \|p_k\|_{w,2}^2 = c_{k,n} \|p_n\|_{w,2}^2 .$$

It follows that

$$c_{n,n-1} = c_{n-1,n} \frac{\|p_n\|_{w,2}^2}{\|p_{n-1}\|_{w,2}^2},$$

respectively,

$$c_{n,n-1} = \frac{a_{n-1}}{a_n} \frac{\|p_n\|_{w,2}^2}{\|p_{n-1}\|_{w,2}^2},$$

and from (1.5.3), the proof follows. ■

Remark 1.5.3. For the polynomials of $\tilde{\mathcal{P}}$, we have

$$\tilde{p}_{n+1}(x) = (x + b_{n+1} - b_n)\tilde{p}_n(x) - \frac{\|\tilde{p}_n\|_{w,2}^2}{\|\tilde{p}_{n-1}\|_{w,2}^2}\tilde{p}_{n-1}(x). \tag{1.5.4}$$

Remark 1.5.4. If $(p_n)_{n\in\mathbb{N}}$ is a sequence of orthonormal polynomials then (1.5.1) becomes

$$\frac{a_n}{a_{n+1}}p_{n+1}(x) = \left(\frac{b_{n+1}}{a_{n+1}} - \frac{b_n}{a_n} + x\right)p_n(x) - \frac{a_{n-1}}{a_n}p_{n-1}(x).$$

Theorem 1.5.5. *If $\tilde{p}_n \perp \mathbb{P}_{n-1}$, on $[a,b]$, with respect to the weight function w, then*

$$\|\tilde{p}_n\|_{w,2} = \min_{p\in\tilde{\mathbb{P}}_n} \|p\|_{w,2}\,.$$

Proof. We have $\tilde{\mathbb{P}}_n = e_n + \mathbb{P}_{n-1}$, i.e., $\tilde{\mathbb{P}}_n$ is a translation of the linear subspace $\mathbb{P}_{n-1}$ of the space $H^{m,2}[a,b]$. In accordance with Corollary 1.4.5, $p^* \in \tilde{\mathbb{P}}_n$ is the best approximation to $0 \in L_w^2[a,b]$ if and only if $p^* \perp \mathbb{P}_{n-1}$. So, $p^* = \tilde{p}_n$. ■

Next, we give a characterization of the orthogonal polynomials.

The recurrence relation of Theorem 1.5.2 is not always the most convenient method for computing the orthogonal polynomials. Some other useful techniques come from the following characterization theorem.

Theorem 1.5.6. *Let $w : [a,b] \to \mathbb{R}$ be any continuous function. The function $\varphi_{k+1} \in C[a,b]$ satisfies the orthogonality conditions,*

$$\int_a^b w(x)\varphi_{k+1}(x)p(x)dx = 0, \qquad p \in \mathbb{P}_k, \tag{1.5.5}$$

if and only if there exists a $k+1$-times differentiable function, $u(x)$, $x \in [a,b]$ that satisfies the relations:

$$w(x)\varphi_{k+1}(x) = u^{(k+1)}(x), \qquad x \in [a,b] \tag{1.5.6}$$

and

$$u^{(i)}(a) = u^{(i)}(b) = 0, \qquad i = 0,1,...,k. \tag{1.5.7}$$

Proof. If equations (1.5.6) and (1.5.7) hold, then the integration by parts gives the following identity:

$$\int_a^b w(x)\varphi_{k+1}(x)p(x)dx = (-1)^{k+1}\int_a^b u(x)p^{(k+1)}(x)dx.$$

Therefore, as $p \in \mathbb{P}_k$, the orthogonality condition (1.5.5) holds.

Conversely, when equation (1.5.5) is satisfied, we let u be defined by expression (1.5.6), where the constants of integration are chosen to give the values

$$u^{(i)}(a) = 0, \qquad i = 0, 1, ..., k.$$

Expression (1.5.6) is substituted in the integral (1.5.5). For each integer j, $j = 0, 1, ..., k$, let $p = p_j$ be the polynomial

$$p_j(x) = (b - x)^j, \qquad x \in [a, b],$$

and we apply integration by parts $j + 1$ times to the left-hand side of expression (1.5.5). Thus, we obtain

$$[(-1)^j u^{(k-j)}(x)p_j^{(j)}(x)]\Big|_a^b + (-1)^{j+1}\int_a^b u^{(k-j)}(x)p_j^{(j+1)}(x) = 0.$$

Because $p_j^{(j+1)}(x) = 0$, it follows that

$$u^{(k-j)}(b) = 0, \qquad j = 0, 1, ..., k,$$

which completes the proof. ■

In order to apply this theorem to generate orthogonal polynomials, it is necessary to identify a function u, satisfying the conditions (1.5.7), such that the function φ_{k+1}, defined by (1.5.6), is a polynomial of $k + 1$ degree. There is no automatic method of identification, but in many important cases the required function u is easy to recognize.

Example 1.5.7. If there are satisfied the relations (1.5.7), by letting u be the function

$$u(x) = (x - a)^{k+1}(x - b)^{k+1}, \qquad x \in [a, b],$$

then it follows that $\varphi_{k+1} \in \mathbb{P}_{k+1}$, when the weight function w is a constant.

In other words, the polynomials

$$\varphi_j(x) = \frac{d^j}{dx^j}[(x-a)^j(x-b)^j], \qquad j = 0, 1, 2, ... \tag{1.5.8}$$

satisfy the orthogonality conditions

$$\int_a^b \varphi_i(x)\varphi_j(x)dx = 0, \qquad i \neq j, \quad i, j = 0, 1, 2, ...$$

Definition 1.5.8. *The polynomial l_n given by (1.5.8), i.e.,*

$$l_n(x) = C\frac{d^n}{dx^n}[(x^2-1)^n]$$

is called the Legendre polynomial *of degree n, relatively to the interval $[-1, 1]$, and*

$$\tilde{l}_n(x) = \frac{n!}{(2n)!}\frac{d^n}{dx^n}[(x^2-1)^n]$$

is the Legendre polynomial with the coefficient of x^n equals to 1.

From (1.5.4) it follows the recurrence formula

$$\tilde{l}_{n+1}(x) = x\tilde{l}_n(x) - \tfrac{n^2}{(2n-1)(2n+1)}\tilde{l}_{n-1}(x), \qquad n = 1, 2, ...,$$

with $\tilde{l}_0(x) = 1$ and $\tilde{l}_1(x) = x$.

Example 1.5.9. Let α and β be real constants that are both greater than -1. The polynomials $(\varphi_i)_{i\in\mathbb{N}}$, that satisfies the orthogonality conditions

$$\int_{-1}^1 (1-x)^\alpha(1+x)^\beta\varphi_i(x)\varphi_j(x)dx = 0, \qquad i \neq j, \quad i, j = 0, 1, 2, ...$$

are called *the Jacobi polynomials.*

In this case we require the function (1.5.6) to be a polynomial of degree $k+1$, multiplied by the weight function

$$w(x) = (1-x)^{\alpha+k+1}(1+x)^{\beta+k+1}, \quad x \in [-1, 1].$$

Because the condition (1.5.7) is satisfied, it follows that the Jacobi polynomials are defined by the equation

$$\varphi_i(x) = (1-x)^{-\alpha}(1+x)^{-\beta}\frac{d^i}{dx^i}[(1-x)^{\alpha+i}(1+x)^{\beta+i}], \quad i = 0, 1, 2, ...,$$

which is called *the Rodrigue's formula.*

Remark 1.5.10. For $\alpha = \beta = 0$ the Jacobi polynomials become *the Legendre polynomials*.

Remark 1.5.11. Each family of orthogonal polynomials is characterized by the weight function w and the interval $[a, b]$.

Example 1.5.12. For the weight function $w(x) = e^{-x}$ and the interval $[0, +\infty)$, one obtains *the Laguerre orthogonal polynomials* g_n, $n = 0, 1, ...$, i.e.,

$$\int_0^\infty e^{-x} g_i(x) g_j(x) dx = \begin{cases} 0, & i \neq j, \\ i!, & j = i. \end{cases}$$

If u is the function

$$u(x) = e^{-x} x^n, \qquad 0 \leq x < \infty,$$

the conditions (1.5.7) are verified and the function φ_{k+1}, defined by equation (1.5.6), is from $\mathbb{P}_{k+1}$. Hence, the Laguerre polynomials are

$$g_n(x) = e^x \frac{d^n}{dx^n}(e^{-x} x^n), \qquad n = 0, 1, 2, ...,$$

with $x \in [0, \infty)$.

The recurrence relation verified by the Laguerre polynomials is:

$$g_{n+1}(x) = \frac{2n + 1 - x}{n + 1} g_n(x) - n g_{n-1}(x), \qquad n = 1, 2, ...,$$

where

$$\begin{aligned} g_0(x) &= 1, \\ g_1(x) &= 1 - x. \end{aligned}$$

Example 1.5.13. *The Hermite polynomials* given by

$$h_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n}(e^{-x^2}), \qquad n = 0, 1, 2, ...$$

fulfill the orthogonality conditions:

$$\int_{-\infty}^\infty e^{-x^2} h_m(x) h_n(x) dx = \begin{cases} 0, & m \neq n \\ 2^n n! \sqrt{\pi}, & m = n. \end{cases}$$

So, h_n, $n = 0, 1, ...$, are orthogonal polynomials on $\mathbb{R}$, with respect to the weight function $w(x) = e^{-x^2}$. They verify the recurrence relation

$$h_{n+1}(x) = 2xh_n(x) - 2nh_{n-1}(x),$$

with

$$\begin{aligned} h_0(x) &= 1, \\ h_1(x) &= 2x. \end{aligned}$$

Example 1.5.14. The polynomial $T_n \in \mathbb{P}_n$ defined by

$$T_n(x) = \cos(n \arccos x), \qquad x \in [-1, 1],$$

is called *the Chebyshev polynomial of the first kind.*

Property 1.5.15. *1)* Algebraic form. *For $x = \cos\theta$ we have*

$$\begin{aligned} T_n(\cos\theta) &= \cos n\theta = \tfrac{1}{2}(e^{in\theta} + e^{-in\theta}) \\ &= \tfrac{1}{2}[(\cos\theta + i\sin\theta)^n + (\cos\theta - i\sin\theta)^n]. \end{aligned}$$

Hence,

$$T_n(x) = \tfrac{1}{2}[(x + \sqrt{x^2-1})^n + (x - \sqrt{x^2-1})^n]. \tag{1.5.9}$$

2) $\tilde{T}_n(x) = \frac{1}{2^{n-1}} T_n(x)$.
From the algebraic form (1.5.9), one obtains

$$\lim_{x\to\infty} \tfrac{T_n(x)}{x^n} = \tfrac{1}{2} \lim_{x\to\infty} \left[\left(1 + \sqrt{1 - \tfrac{1}{x^2}}\right)^n + \left(1 - \sqrt{1 - \tfrac{1}{x^2}}\right)^n \right] = 2^{n-1}.$$

So, the coefficient of x^n in T_n is 2^{n-1}. It follows that

$$\tilde{T}_n = \tfrac{1}{2^{n-1}} T_n.$$

3) The orthogonality property.
Starting with

$$\int_0^\pi \cos m\theta \cos n\theta d\theta = \begin{cases} 0, & m \neq n, \\ \frac{\pi}{2}, & m = n \neq 0, \\ \pi, & m = n = 0, \end{cases}$$

and taking $\cos\theta = x$, *it follows that*

$$\int_{-1}^{1} \frac{1}{\sqrt{1-x^2}} T_m(x)T_n(x)dx = \begin{cases} 0, & m \neq n, \\ c \neq 0, & m = n. \end{cases}$$

So, $(T_n)_{n\in\mathbb{N}}$ *are orthogonal polynomials on* $[-1,1]$, *with regard to the weight function* $w(x) = 1/\sqrt{1-x^2}$.

4) Theorem 1.5.5 implies that

$$\left\|\tilde{T}_n\right\|_{w,2} = \min_{P\in\tilde{\mathbb{P}}_n} \|P\|_{w,2}\,.$$

5) The roots of the polynomial T_n *are*

$$x_k = \cos\tfrac{2k-1}{2n}\pi, \qquad k = 1, ..., n,$$

that are real, distinct and belonging to $(-1,1)$.

6) The recurrence relation:

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \qquad x \in [-1,1],$$

with $T_0(x) = 1$ *and* $T_1(x) = x$.

This is provided by the identity

$$\cos[(n+1)\theta] + \cos[(n-1)\theta] = 2\cos\theta\cos n\theta.$$

7) On the interval $[-1,1]$ *we have*

$$\left\|\tilde{T}_n\right\|_{\infty} = \min_{P\in\tilde{\mathbb{P}}_n} \|P\|_{\infty}\,.$$

Proof. We have

$$\left\|\tilde{T}_n\right\|_{\infty} = \tfrac{1}{2^{n-1}}$$

and

$$\tilde{T}_n(x'_k) = \tfrac{(-1)^k}{2^{n-1}},$$

where

$$x'_k = \cos\tfrac{k\pi}{n}, \qquad k = 0, 1, ..., n,$$

are the roots of the polynomial T'_n, i.e.,

$$T'_n(x) = \frac{n\sin(n\arccos x)}{\sqrt{1-x^2}}.$$

Now, suppose that there exists a polynomial $P_n \in \tilde{\mathbb{P}}_n$, with $\|P_n\|_\infty < \frac{1}{2^{n-1}}$. It follows that

$$P_n = \tilde{T}_n + P_{n-1}, \quad \text{with } P_{n-1} \in \mathbb{P}_{n-1}.$$

Hence,

$$P_{n-1}(x'_k) = P_n(x'_k) - \tfrac{(-1)^k}{2^{n-1}}.$$

As, $\|P_n\| < \frac{1}{2^{n-1}}$, for $k = 0, 1, ..., n-1$ we have

$$\begin{array}{l} P_{n-1}(x'_k) < 0, \quad \text{for } k \text{ even}, \\ P_{n-1}(x'_k) > 0, \quad \text{for } k \text{ odd}. \end{array}$$

It follows that P_{n-1} changes its sign for $n+1$ times in $[-1, 1]$, i.e., it has at least n roots. As the degree of P_{n-1} is at most $n-1$ we obtain that $P_{n-1} = 0$, hence $P_n = T_n$. ■

Example 1.5.16. The polynomial $Q_n \in \mathbb{P}_n$ defined by

$$Q_n(x) = \frac{\sin((n+1)\arccos x)}{\sqrt{1-x^2}}, \qquad x \in [-1, 1]$$

is called *the Chebyshev polynomial of the second kind.*

Remark 1.5.17. The following relation is fulfilled:

$$Q_n(x) = \tfrac{1}{n+1} T'_{n+1}(x), \qquad x \in [-1, 1].$$

As the coefficient of x^n is 2^{n-1}, in T_n, it follows that

$$\tilde{Q}_n = \tfrac{1}{2^n} Q_n.$$

We remark also that

$$\int_{-1}^{1} \sqrt{1-x^2} Q_m Q_n dx = \begin{cases} 0, & m \neq n, \\ \frac{\pi}{2}, & m = n, \end{cases}$$

i.e., $(Q_n)_{n\in\mathbb{N}}$ is an orthogonal sequence on $[-1, 1]$, with respect to the weight function $w(x) = \sqrt{1-x^2}$. From the identity

$$\sin(n+2)\theta + \sin n\theta = 2\cos\theta \sin(n+1)\theta,$$

one obtains the recurrence formula

$$Q_{n+1}(x) = 2xQ_n(x) - Q_{n-1}(x), \qquad n = 1, 2, ...,$$

with $Q_0(x) = 1$ and $Q_1(x) = 2x$.

Also,

$$\left\|\tilde{Q}_n\right\|_{w,2} = \min_{P\in\tilde{\mathbb{P}}_n} \|P\|_{w,2}\,.$$

A very important property of the polynomials Q_n is that

$$\left\|\tilde{Q}_n\right\|_{w,1} = \min_{P\in\tilde{\mathbb{P}}_n} \|P\|_{w,1}\,, \qquad \text{on } [-1,1],$$

i.e., the minimum norm property takes place also in the norm of $L^1_w[-1,1]$.

Chapter 2

Approximation of functions

2.1 Preliminaries

Functions are the basic mathematical tools for describing (modeling) many physical processes. Very frequently these functions are not known explicitly and it is necessary to construct approximations of them, based on limited information about the underlying process. As a physical process depends on one or more parameters, the problem which appears is to approximate a function of one or several variables based on some limited information about it.

The most commonly approach for finding approximations to unknown functions is:

1) choose a reasonable class of functions $\mathcal{A}$, in which to look for the approximations;

2) select an appropriate approximation process for assigning a specific function to a specific problem.

The success of this approach heavily depends on the existence of the convenient set $\mathcal{A}$ of approximating functions.

Such a set $\mathcal{A}$ should possess at least the following properties:

1) the functions in $\mathcal{A}$ should be relatively smooth;

2) the set $\mathcal{A}$ should be large enough so that arbitrary smooth functions can be well approximated by elements of $\mathcal{A}$;

3) the functions in $\mathcal{A}$ should be easy to store, manipulate and evaluate on a digital computer.

The study of various classes of approximating functions is the content of

the approximation theory. The computational aspects of such approximations are a major part of numerical analysis.

The most important classes of approximation functions used in this book are: polynomials, polynomial splines and rational functions.

1) Polynomials ($\mathcal{A} \subset \mathbb{P}$).

The polynomials have always played a central role in approximation theory and numerical analysis. We note that the space $\mathbb{P}_m$, of polynomials of degree at most m, has the following properties:

1. It is a finite dimensional linear space with a convenient basis (e_i, $i = 0, 1, ..., m$, with $e_i(x) = x^i$).

2. Polynomials are smooth functions.

3. The derivatives and primitives of the polynomials are also polynomials whose coefficients can be found algebraically.

4. The number of zeros of a polynomial of degree m cannot exceed m.

5. Various matrices arising in interpolation and approximation by polynomials are usually nonsingular and they have strong sign-regularity properties.

6. The sign structure and shape of a polynomial are strongly related to the sign structure of its set of coefficients.

7. Given any continuous function on an interval $[a, b]$, there exists a polynomial which is uniformly close to it.

8. Precise rates of convergence can be given for approximation of smooth functions by polynomials.

9. Polynomials are easy to store, manipulate and evaluate on computer.

2) Polynomial splines ($\mathcal{A} \subset \mathcal{S}_m$).

The main drawbacks of the space $\mathbb{P}_m$ for approximation of functions is that the set $\mathbb{P}_m$ is relatively inflexible. Polynomials seem to be all right on sufficiently small intervals, but when we go to larger intervals often appear severe oscillations, particulary for large m. This observation suggests that in order to achieve a class of approximating functions with greater flexibility, we should work with polynomials of relatively low degree and divide the interval into smaller pieces. This is the class of polynomial splines.

The space $\mathcal{S}_m$ of polynomial splines of order m have the following properties:

1. Polynomial spline spaces are finite dimensional linear spaces with very convenient bases.

2. Polynomial splines are relatively smooth functions.

3. The derivatives and primitives of polynomial splines are also polynomial splines.

4. Polynomial splines posses nice zeros properties analogous to those for polynomials.

5. Various matrices arising naturally in the use of splines in approximation theory and numerical analysis have convenient sign and properties.

6. The sign structure and shape of a polynomial spline can be related to the sign structure of its coefficients.

7. Every continuous function on an interval $[a, b]$ can be well approximated by polynomial splines with the order m fixed, supposing that a sufficient number of knots are allowed.

8. Precise rates of convergence can be given for approximation of smooth functions by splines.

9. Low-order splines are very flexible and do not exhibit the oscillations usually associated with polynomials.

10. Polynomial splines are easy to store, manipulate and evaluate on computer.

3) Rational functions ($\mathcal{A} = \mathcal{R}_{p,q}$).

Polynomials are not suitable to approximate, for example, a function $f : [a, b] \to \mathbb{R}$ for which a line of equation $x = a - \varepsilon$ or $x = a + \varepsilon$ is an asymptote, or for the case when f is an exponential function and so on.

In such cases a rational function can give better approximation than a polynomial.

In a computational structure sense, a rational function R is the one that can be evaluated as the quotient of two polynomials, i.e.,

$$R(x) = \frac{P(x)}{Q(x)} = \frac{a_0 + a_1 x + ... + a_m x^m}{b_0 + b_1 x + ... + b_n x^n},$$

where $a_m \neq 0$, $b_n \neq 0$. When $n = 0$ and $b_0 \neq 0$, then R reduces to a polynomial.

A rational function R is finitely representable by the coefficients $(a_0, ..., a_m)$ and $(b_0, ..., b_n)$, so it is computable by a finite number of arithmetic operations, as in the polynomial or polynomial spline case.

Definition 2.1.1. *Let $\mathcal{B}$ be a normed linear space of real-valued functions defined on $\Omega \subset \mathbb{R}^n$, $\mathcal{A} \subset \mathcal{B}$ and $f \in \mathcal{B}$.*

- *$P : \mathcal{B} \to \mathcal{A}$ is an approximation operator;*

- Pf *is the approximation of* f *by the operator* P;
- $f = Pf + Rf$ *is the approximation formula generated by* P, *where* R *is the corresponding remainder operator;*
- *if* $P(Pf) = Pf$ *then the operator* P *is* idempotent;
- *if* P *is linear and idempotent then* P *is* a projector;
- *the smallest real number, denoted by* $\|P\|$, *with the property that*

$$\|Pf\| \leq \|P\| \, \|f\|, \text{ for all } f \in \mathcal{B}$$

is the norm *of the operator* P.

Remark 2.1.2. The norm of an approximation operator is also called *the Lebesgue constant* of the operator.

Definition 2.1.3. *The number* $r \in \mathbb{N}$ *for which*

$$Pf = f, \qquad f \in \mathbb{P}_r^n$$

and there exists $g \in \mathbb{P}_{r+1}^n$ *such that* $Pg \neq g$ *is called* the degree of exactness *of the operator* P, *(denoted by* $\operatorname{dex}(P)$*).*

Remark 2.1.4. If P is linear then ($Pf = f$, $f \in P_r^n$ and there exists $g \in \mathbb{P}_{r+1}^n$ such that $Pg \neq g$) is equivalent to ($Pe_{i_1,...,i_n} = e_{i_1,...,i_n}$, for all $i_1 + ... + i_n \leq r$ and there exists $(\mu_1, ..., \mu_n) \in \mathbb{N}^n$ with $\mu_1 + ... + \mu_n = r + 1$ such that $Pe_{\mu_1,...,\mu_n} \neq e_{\mu_1,...,\mu_n}$), where $e_{\nu_1,...,\nu_n}(x_{1,...,}x_n) = x_1^{\nu_1}..x_n^{\nu_n}$.

Suppose that $\mathcal{B}$ is a set of real-valued functions defined on $\Omega \subset \mathbb{R}^n$, $\mathcal{A} \subset \mathcal{B}$ and let

$$\Lambda := \{\lambda_i \mid \lambda_i : \mathcal{B} \to \mathbb{R}, \quad i = 1, ..., N\}$$

be a set of linear functionals. The information of $f \in \mathcal{B}$ corresponding to the set Λ is denoted by

$$\Lambda(f) := \{\lambda_i(f) \mid \lambda_i \in \Lambda, \quad i = 1, ..., N\}.$$

Definition 2.1.5. *The operator* $P : \mathcal{B} \to \mathcal{A}$ *for which*

$$\lambda_i(Pf) = \lambda_i(f), \qquad i = 1, ..., N, \text{ for } f \in \mathcal{B},$$

is called an interpolation operator *that interpolates the set* Λ. *Formula*

$$f = Pf + Rf$$

is the interpolation formula *generated by* P, *and* R *is* the remainder operator.

Definition 2.1.6. *If for a given* Λ*, the operator* P *exists, then* Λ *is called* an admissible set of functionals, *respectively* $\Lambda(f)$ *is* an admissible set of information.

Definition 2.1.7. *If* p_i, $i = 1, ..., N$ *are fundamental (cardinal) interpolation elements suitable to the interpolation operator* P*, i.e.,*

$$\lambda_k p_i = \delta_{ki}, \qquad i, k = 1, ..., N$$

then

$$Pf = \sum_{i=1}^{N} p_i \lambda_i(f).$$

Now, let $\Lambda_i^{\nu_i} \subset \Lambda$ be a subset of ν_i ($\nu_i < N$) functionals of Λ, associated to λ_i, $i = 1, ..., N$, respectively. We have

$$\bigcup_{i=1}^{N} \Lambda_i^{\nu_i} = \Lambda.$$

Let $Q_i^{\nu_i}$ be the operator that interpolates the set $\Lambda_i^{\nu_i}$.

Definition 2.1.8. *The operator* P_Q *defined by*

$$P_Q = \sum_{i=1}^{N} p_i Q_i^{\nu_i}$$

is called the combined operator *of* P *and* $Q_i^{\nu_i}$, $i = 1, ..., N$.

Remark 2.1.9. If P and $Q_i^{\nu_i}$, $i = 1, ..., N$ are linear operators then the combined operator P_Q is also linear.

Definition 2.1.10. *A sequence of subsets* $\Lambda_i^{\nu_i} \subset \Lambda$, $i = 1, ..., N$, *for which the suitable interpolation operators* $Q_i^{\nu_i}$, $i = 1, ..., N$ *exist for all* $i = 1, ..., N$, *is called* an admissible sequence.

Theorem 2.1.11. *Let* P *and* $Q_i^{\nu_i}, i = 1, ..., N$*, be the operators that interpolate the sets* Λ *and* $\Lambda_i^{\nu_i}$, $i = 1, ..., N$*, respectively. If* Λ *is an admissible set and* $\Lambda_1^{\nu_1}, ..., \Lambda_N^{\nu_N}$ *is an admissible sequence then the combined operator* P_Q *exists.*

Proof. As Λ is an admissible set and $\Lambda_1^{\nu_1}, ..., \Lambda_N^{\nu_N}$ is an admissible sequence, we have that all the operators P and $Q_i^{\nu_i}$, $i = 1, ..., N$ exist, so the proof follows. ■

Theorem 2.1.12. *Let P and $Q_i^{\nu_i}$, $i = 1, ..., N$, be some linear operators. If* $\text{dex}\ (P) \geq 0\ \left(\sum_{i=1}^{N} p_i = 1\right)$ *and* $\text{dex}(Q_i^{\nu_i}) = r_i$, $i = 1, ..., N$, *then*

$$\text{dex}(P_Q) = r_m := \min\{r_1, ..., r_N\}.$$

Proof. As, $\text{dex}(Q_i^{\nu_i}) = r_i$ and $Q_i^{\nu_i}$ is linear, we have

$$Q_i^{\nu_i} e_{i_1,...,i_n} = e_{i_1,...,i_n},$$

for all $(i_1, ..., i_n) \in \mathbb{N}^n$, with $i_1 + ... + i_n \leq r_i$ and there exists $(\mu_1, ..., \mu_n) \in \mathbb{N}^n$, with $\mu_1 + ... + \mu_n = r_i + 1$, such that

$$Q_i^{\nu_i} e_{\mu_1,...,\mu_n} \neq e_{\mu_1,...,\mu_n},$$

for $i = 1, ..., N$, respectively. It follows that

$$Q_i^{\nu_i} e_{i_1,...,i_n} = e_{i_1,...,i_n},$$

with $i_1 + ... + i_n \leq r_m$, $(r_m = \min\{r_1, ..., r_N\})$, for all $i = 1, ..., N$ and there exists an index j, $1 \leq j \leq N$, for which

$$Q_j^{\nu_j} e_{\mu_1,...,\mu_n} \neq e_{\mu_1,...,\mu_n},$$

with $\mu_1 + ... + \mu_n = r_m + 1$. We have

$$P_Q e_{i_1,...,i_n} = \sum_{k=1}^{N} p_k Q_k^{\nu_k} e_{i_1,...,i_n} = e_{i_1,...,i_n} \sum_{k=1}^{N} p_k = e_{i_1,...,i_n},$$

for all $i_1 + ... + i_n \leq r_m$. From

$$Q_j^{\nu_j} e_{\mu_1,...,\mu_n} \neq e_{\mu_1,...,\mu_n}, \quad \text{when } \mu_1 + ... + \mu_n = r_m + 1,$$

it follows that

$$P_Q e_{\mu_1,...,\mu_n} \neq e_{\mu_1,...,\mu_n}.$$

Finally, the linearity of P_Q implies that

$$\text{dex}(P_Q) = r_m.$$

■

2.2 Univariate interpolation operators

Suppose that $\mathcal{B}$ is a linear space of univariate real-valued functions defined on the interval $[a, b] \subset \mathbb{R}$, $\mathcal{A} \subset \mathcal{B}$,

$$\Lambda = \{\lambda_i \mid \lambda_i : \mathcal{B} \to \mathbb{R},\ i = 1, ..., N\}$$

is a set of linear functionals and $P : \mathcal{B} \to \mathcal{A}$ is the interpolation operator suitable to Λ, i.e.,

$$\lambda_i(Pf) = \lambda_i(f), \qquad i = 1, ..., N, \text{ for } f \in \mathcal{B}.$$

The basic elements in the definition of on interpolation operator are the sets $\mathcal{A}$ and Λ. For a given pair $(\mathcal{A}, \Lambda)$, it is obtained a corresponding interpolation operator.

For $\mathcal{A}$ will be considered the sets $\mathbb{P}$, $\mathcal{S}$ and $\mathcal{R}$ of polynomial functions, natural spline functions, respectively, rational functions.

Supposing that $x_i \in [a, b],\ i = 0, ..., m$ are the interpolation nodes, for Λ will be considered the functionals of the following types:

- Lagrange type (Λ_L):

$$\Lambda_L = \{\lambda_i \mid \lambda_i(f) = f(x_i), \quad i = 0, ..., m\};$$

- Hermite type (Λ_H):

$$\Lambda_H = \{\lambda_{ij} \mid \lambda_{ij}(f) = f^{(j)}(x_i),\ j = 0, ..., r_i;\ i = 0, ..., m\},$$

  with $r_i \in \mathbb{N}$, $i = 0, ..., m$;

- Birkhoff type (Λ_B):

$$\Lambda_B = \{\lambda_{ij} \mid \lambda_{ij}(f) = f^{(j)}(x_i),\ j \in I_i,\ i = 0, ..., m\},$$

  with

$$I_i \subset \{0, 1, ..., r_i\},\ \text{for } r_i \in \mathbb{N},\ i = 0, ..., m.$$

Hence, for example, the pair $(\mathbb{P}, \Lambda_L)$ gives rise to the polynomial interpolation operators of Lagrange type, the pair $(\mathcal{S}, \Lambda_B)$ generates spline interpolation operators of Birkhoff type and so on.

2.2.1 Polynomial interpolation operators

Consider the polynomial interpolation operators of Lagrange $(\mathbb{P}, \Lambda_L)$, Hermite $(\mathbb{P}, \Lambda_H)$ and Birkhoff $(\mathbb{P}, \Lambda_B)$ type, briefly called Lagrange (L_m), Hermite (H_n), respectively Birkhoff (B_k) interpolation operators, where $m = |\Lambda_L| - 1$, $n = |\Lambda_H| - 1$ and $k = |\Lambda_B| - 1$, i.e., the degree of the corresponding polynomials.

Next we give the interpolation formulas generated by these operators.

★ *Lagrange interpolation formula:*

$$f = L_m f + R_m f, \tag{2.2.1}$$

with Lagrange interpolation polynomial given by:

$$(L_m f)(x) = \sum_{i=0}^{m} \frac{u_m(x)}{(x-x_i)u'_m(x_i)} f(x_i), \tag{2.2.2}$$

with

$$u_m(x) = \prod_{i=0}^{m} (x - x_i).$$

For $f \in C^m[a, b]$ and such that there exists $f^{(m+1)}$ on (a, b), we have

$$(R_m f)(x) = \frac{u_m(x)}{(m+1)!} f^{(m+1)}(\xi), \quad a < \xi < b, \tag{2.2.3}$$

respectively, while if $f \in C^{m+1}[a, b]$ we have

$$(R_m f)(x) = \int_a^b \varphi_m(x, t) f^{(m+1)}(t) dt, \tag{2.2.4}$$

where

$$\varphi_m(x, t) = R_m \left[\frac{(\cdot - t)_+^m}{m!} \right]$$

is the Peano's kernel.

Remark 2.2.1. If $f^{(m+1)}$ is also continuous on $[a, b]$ we have

$$\|R_m f\|_\infty \leq \frac{\|u_m\|_\infty}{(m+1)!} \left\| f^{(m+1)} \right\|_\infty .$$

Minimal norm property: $\|R_m f\|_\infty$ takes the minimum value when $\|u_m\|_\infty$ takes the minimum value, i.e., when $u_m \equiv \tilde{T}_{m+1}(\cdot, a, b)$, where $\tilde{T}_{m+1}$

is the Chebyshev polynomial of the first kind. It means that the interpolation nodes are the zeros of T_{m+1} and

$$\|R_{mf}\|_\infty \leq \frac{(b-a)^{m+1}}{(m+1)!2^{2m+1}} \left\|f^{(m+1)}\right\|_\infty .$$

★ *Hermite interpolation formula:*

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

where

$$(H_n f)(x) = \sum_{i=0}^{m} \sum_{j=0}^{r_i} h_{ij}(x) f^{(j)}(x_i),$$

with

$$h_{ij}(x) = u_i(x) \frac{(x-x_i)^j}{j!} \sum_{\nu=0}^{r_i-j} \frac{(x-x_i)^\nu}{\nu!} \left[\frac{1}{u_i(x)}\right]^{(\nu)}_{x=x_i}$$

and

$$u_n(x) = \prod_{i=0}^{m} (x-x_i)^{r_i+1}, \qquad u_i(x) = \frac{u_n(x)}{(x-x_i)^{r_i+1}}.$$

We present two representations of the remainder:

- for $f \in C^n[a,b]$, and such that there exists $f^{(n+1)}$ on (a,b), we have

$$(R_n f)(x) = \frac{u_n(x)}{(n+1)!} f^{(n+1)}(\xi), \quad a < \xi < b;$$

- for $f \in C^{n+1}[a,b]$ we have

$$(R_n f)(x) = \int_a^b \varphi_n(x,t) f^{(n+1)}(t) dt,$$

where φ_n is the corresponding Peano's kernel.

★ *Birkhoff interpolation formula:*

$$f = B_k f + R_k f,$$

with

$$(B_k f)(x) = \sum_{i=0}^{m} \sum_{j \in I_i} b_{ij}(x) f^{(j)}(x_i), \tag{2.2.5}$$

where b_{ij} are obtained from the cardinality conditions:

$$\begin{cases} b_{ij}^{(p)}(x_\nu) = 0, & \nu \neq i,\ p \in I_\nu, \\ b_{ij}^{(p)}(x_i) = \delta_{jp}, & p \in I_i, \end{cases} \tag{2.2.6}$$

for all $j \in I_i$ and $i = 0, ..., m$. For $f \in C^{k+1}[a, b]$ we have:

$$(R_k f)(x) = \int_a^b \varphi_k(n, t) f^{(k+1)}(t) dt,$$

with

$$\varphi_k(a, t) = R\left[\frac{(\cdot - t)_+^k}{k!}\right].$$

Particular cases of Birkhoff interpolation.

• *Abel-Goncharov interpolation formula:*

$$f = P_n f + R_n f,$$

with

$$(P_n f)(x) = \sum_{k=0}^{n} g_k(x) f^{(k)}(x_k),$$

where g_k, $k = 0, ..., n$ are called Goncharov polynomials of degree k, and they have the following expressions:

$$\begin{aligned} g_0(x) &= 1, \\ g_1(x) &= x - x_0, \\ g_k(x) &= \int_{x_0}^{x} dt_1 \int_{x_1}^{t_1} dt_2 \cdots \int_{x_{k-1}}^{t_{k-1}} dt_k \\ &= \tfrac{1}{k!}\left[x^k - \sum_{j=0}^{k-1} g_j(x)\binom{k}{j} x_j^{k-j}\right], \quad k = 2, ..., n. \end{aligned} \tag{2.2.7}$$

For $n \in \mathbb{N}$, $a, b \in \mathbb{R}$, $a < b$ and a function $f : [a, b] \to \mathbb{R}$, having n derivatives $f^{(i)}$, $i = 1, 2, ..., n$, the Abel-Goncharov interpolation problem consists in finding a polynomial $P_n f$ of degree n such that:

$$(P_n f)^{(i)}(x_i) = f^{(i)}(x_i), \quad 0 \leq i \leq n. \tag{2.2.8}$$

If $f \in H^{n+1}[a,b]$ then

$$(R_n f)(x) = \int_a^b \varphi_n(x,s) f^{(n+1)}(s) ds,$$

with

$$\varphi_n(x,s) = \frac{(x-s)_+^n}{n!} - \sum_{k=0}^{n} g_k(x) \frac{(x_k-s)_+^{n-k}}{(n-k)!}. \tag{2.2.9}$$

The determinant of the corresponding interpolation system is always nonzero, so the problem (2.2.8) always has a unique solution.

• *Lidstone interpolation formula:*

$$f = L_m^{\Delta} f + R_m^{\Delta} f, \tag{2.2.10}$$

with

$$(L_m^{\Delta} f)(x) = \sum_{i=0}^{N+1} \sum_{j=0}^{m-1} r_{m,i,j}(x) f^{(2j)}(x_i),$$

where $r_{m,i,j}$, $0 \le i \le N+1$, $0 \le j \le m-1$ are the basic elements of the set $L_m(\Delta) := \{h \in C[a,b] : h$ is a polynomial of degree at most $2m-1$ in each subinterval $[x_i, x_{i+1}]$, $0 \le i \le N\}$, satisfying

$$D^{2\nu} r_{m,i,j}(x_\mu) = \delta_{i\mu}\delta_{2\nu,j}, \quad 0 \le \mu \le N+1, 0 \le \nu \le m-1 \tag{2.2.11}$$

and

$$r_{m,i,j}(x) = \begin{cases} \Lambda_j\left(\frac{x_{i+1}-x}{h}\right) h^{2j}, & x_i \le x \le x_{i+1},\ 0 \le i \le N \\ \Lambda_j\left(\frac{x-x_{i-1}}{h}\right) h^{2j}, & x_{i-1} \le x \le x_i,\ 1 \le i \le N+1 \\ 0, \quad \text{otherwise} \end{cases}$$

with Δ denoting a fixed partition of $[a,b]$.

The Lidstone polynomial Λ_n of degree $2n+1$, $n \in \mathbb{N}$ on the interval $[0,1]$, is defined by:

$$\begin{aligned} &\Lambda_0(x) = x, \\ &\Lambda_n''(x) = \Lambda_{n-1}(x), \\ &\Lambda_n(0) = \Lambda_n(1) = 0, \quad n \ge 1, \end{aligned} \tag{2.2.12}$$

For $f \in C^{2m-2}[a,b]$, the Lidstone polynomial uniquely exists and it satisfies the interpolation conditions:

$$(L_m^{\Delta} f)^{(2k)}(x_i) = f^{(2k)}(x_i), \quad 0 \le k \le m-1,\ 0 \le i \le N+1. \tag{2.2.13}$$

The remainder is given by

$$(R_m^{\Delta} f)(x) = \int_a^b g_m(x,s) f^{(2m)}(s) ds, \tag{2.2.14}$$

where $g_m(x,s)$ is the corresponding Peano's kernel.

Common properties:

1. Evidently, all the operators L_m, H_n, B_k, P_n and L_m^{Δ} are interpolation operators.

2. Each operator L_m, H_n, B_k, P_n and L_m^{Δ} has the degree of exactness equal to the degree of corresponding interpolation polynomial, i.e.,

$$\mathrm{dex}(L_m) = m,\ \mathrm{dex}(H_n) = n,\ \mathrm{dex}(B_k) = k,\ \ \mathrm{dex}(P_n) = n \text{ and } \mathrm{dex}(L_m^{\Delta}) = 2m-1.$$

3. All operators L_m, H_n, B_k, P_n and L_m^{Δ}, and their corresponding remainder operators are projectors.

Existence and uniqueness conditions:

1. If $x_i \neq x_j$, for $i \neq j$; $i, j = 0, ..., m$ the operators L_m and H_n exist and are unique.

2. A necessary condition for the existence of the Birkhoff operator B_k is given by the Pólya condition. Let

$$E = (e_{ij})_{i=0, j\in I_i}^{m}$$

be the $(m+1) \times (k+1)$ matrix associated to Λ_B, in which

$$e_{ij} = \begin{cases} 1, & \text{if } j \in I_i, \\ 0, & \text{if } j \notin I_i, \end{cases}$$

for all $i = 0, ..., m$. One notices that the number of 1's in E is exactly $k+1$.

The matrix E is called *the incidence matrix* corresponding to the set Λ_B.

In particular, for Λ_L the $(m+1) \times (m+1)$ incidence matrix is

$$E_L = \begin{pmatrix} 1 & 0 & ... & 0 \\ 1 & 0 & ... & 0 \\ \vdots & \vdots & & \vdots \\ 1 & 0 & ... & 0 \end{pmatrix}.$$

Considering Hermite interpolation, the i-th row of the $(m+1) \times (n+1)$ incidence matrix E_H is

$$(\underbrace{1, ..., 1}_{r_i+1}, \underbrace{0, ..., 0}_{n-r_i}), \qquad \text{for } i = 0, 1, ..., m.$$

An incidence matrix E satisfies the Pólya condition if

$$\sum_{j=0}^{r}\sum_{i=0}^{m} e_{ij} \geq r+1, \quad \text{for all } r=0,...,k.$$

It can be seen that the incidence matrices E_L and E_H satisfy the Pólya condition.

Next consider an example of Birkhoff interpolation.

Example 2.2.2. For $f \in C^3[a,b]$ one considers the set of information

$$\Lambda_B(f) = \{f'(0), f(\tfrac{h}{2}), f'(h)\}.$$

The incidence matrix is

$$E_B = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

that verifies the Pólya condition. Therefore, we look for the interpolation polynomial, as a polynomial of degree 2 ($k=2$):

$$(B_2 f)(x) = Ax^2 + Bx + C, \tag{2.2.15}$$

that interpolates the information $\Lambda_B(f)$. One obtains

$$\begin{cases} (B_2f)'(0) := B = f'(0) \\ (B_2f)(\frac{h}{2}) := \frac{h^2}{4}A + \frac{h}{2}B + C = f(\frac{h}{2}) \\ (B_2f)'(h) := 2hA + B = f'(h). \end{cases} \tag{2.2.16}$$

The determinant of this system is $D = 2h \neq 0$, hence $B_2 f$ exists and is unique. There are two ways to find it.

1) Solving the system (2.2.16) directly. One obtains

$$\begin{aligned} A &= \tfrac{f'(h)-f'(0)}{2h}, \\ B &= f'(0), \\ C &= f\left(\tfrac{h}{2}\right) - \tfrac{h}{8}f'(h) - \tfrac{3h}{8}f'(0), \end{aligned}$$

so

$$(B_2 f)(x) = \tfrac{f'(h)-f'(0)}{2h}x^2 + f'(0)x + f\left(\tfrac{h}{2}\right) - \tfrac{h}{8}f'(h) - \tfrac{3h}{8}f'(0),$$

or

$$(B_2f)(x) = \tfrac{(2x-h)(3h-2x)}{8h} f'(0) + f\left(\tfrac{h}{2}\right) + \tfrac{4x^2-h^2}{8h} f'(h).$$

2) Using the general form (2.2.5):

$$(B_2f)(x) = b_{01}(x)f'(0) + b_{10}(x)f\left(\tfrac{h}{2}\right) + b_{21}(x)f'(h)$$

and finding the cardinal polynomials b_{01}, b_{10} and b_{21}. Each of these polynomials is of degree 2 (as in (2.2.15)) and they must satisfy the cardinality conditions (2.2.6), i.e.:

$$\begin{cases} b'_{01}(0) = 1 \\ b_{01}\left(\frac{h}{2}\right) = 0 \\ b'_{01}(h) = 0 \end{cases} \quad \begin{cases} b'_{10}(0) = 0 \\ b_{10}(\frac{h}{2}) = 1 \\ b'_{10}(h) = 0 \end{cases} \quad \begin{cases} b'_{21}(0) = 0 \\ b_{21}(\frac{h}{2}) = 0 \\ b'_{21}(h) = 1. \end{cases}$$

Solving these systems, we get

$$\begin{aligned} b_{01}(x) &= \tfrac{(2x-h)(3h-2x)}{8h}, \\ b_{10}(x) &= 1, \\ b_{21}(x) &= \tfrac{4x^2-h^2}{8h}. \end{aligned}$$

Now, regarding the remainder term of the interpolation formula

$$f = B_2f + R_2f$$

we have:

$$(R_2f)(x) = \int_0^h \varphi_2(x,t) f^{(3)}(t)dt,$$

where

$$\varphi_2(x,t) = \tfrac{1}{2}[(x-t)_+^2 - (\tfrac{h}{2}-t)_+^2 - \tfrac{4x^2-h^2}{4h}(h-t)].$$

By a straightforward computation one obtains that

$$\varphi_2(x,t) \geq 0, \quad x \in [0, \tfrac{h}{2}]$$

and

$$\varphi_2(x,t) \leq 0, \quad x \in [\tfrac{h}{2}, h],$$

for $t \in [0,h]$. As, for a given $x \in [0,h]$, $\varphi_2(x,t)$ do not change the sign, for $t \in [0,h]$, and $f^{(3)}$ is continuous on $[0,h]$, using the mean value theorem, one obtains:

$$(R_2f)(x) = f^{(3)}(\xi) \int_0^h \varphi_2(x,t)dt, \qquad 0 \leq \xi \leq h.$$

Finally, we have

$$(R_2 f)(x) = \frac{(2x-h)(2x^2-2hx-h^2)}{24} f^{(3)}(\xi), \qquad 0 \leq \xi \leq h,$$

respectively,

$$\|R_2 f\|_\infty \leq \frac{h^3}{24} \left\|f^{(3)}\right\|_\infty .$$

2.2.2 Polynomial spline interpolation operators

Let $\Lambda = \{\lambda_i \mid \lambda_i : H^{m,2}[a,b] \to \mathbb{R},\ i = 1, ..., n\}$ be a set of linear functionals, $y \in \mathbb{R}^n$ and

$$U(y) = \left\{ f \in H^{m,2}[a,b] \mid \lambda_i(f) = y_i, \quad i = 1, ..., n \right\}. \tag{2.2.17}$$

Definition 2.2.3. *The problem of finding the elements $s \in U(y)$ with the property that*

$$\left\|s^{(m)}\right\|_2 = \inf_{u \in U(y)} \left\|u^{(m)}\right\|_2$$

is called a Polynomial Spline Interpolation Problem (*denoted* PSIP).

Remark 2.2.4. A solution s of a (PSIP) is the smoothest function ($\left\|s^{(m)}\right\|_2 \to$ min), or in other words, the "closest" function to a polynomial $P \in \mathbb{P}_{m-1}$ ($\left\|P^{(m)}\right\|_2 = 0$), that also satisfies the conditions $\lambda_i(P) = y_i$, $i = 1, ..., n$.

A solution of a (PSIP) is called *a polynomial spline function* that interpolates y with respect to Λ, or *a natural spline* that interpolates y.

Next, we are looking for some existence and uniqueness conditions, basic properties and a structural characterization for a solution of a (PSIP). For these, we need some results given by the following lemmas.

Lemma 2.2.5. *Let e_i, $i = 0, ..., m-1$ be a basis of $\mathbb{P}_{m-1}$ and let $k \in \mathbb{N}$, $0 \leq k \leq m-1$. There exists a linear functional λ_k, defined on $H^{m,2}[a,b]$, such that*

$$\lambda_k(e_i) = \delta_{ki}, \qquad i = 0, ..., m-1.$$

Proof. Considering $f \in H^{m,2}[a,b]$, it is known that it can be represented uniquely in the form

$$f(x) = p_f(x) + \int_a^x \frac{(x-t)^{m-1}}{(m-1)!} f^{(m)}(t)dt,$$

with $p_f \in \mathbb{P}_{m-1}$. Denote

$$p_f = \sum_{i=0}^{m-1} \alpha_i(f) e_i$$

the representation of p_f in terms of e_i, $i = 0, ..., m-1$. Thus, the functional $\lambda_k = \alpha_k(f)$ has the desired property. ■

Lemma 2.2.6. *Let $f \in L^p[c,d]$, $1 < p < \infty$. If*

$$\int_c^d f(x)\varphi(x)dx = 0, \qquad \varphi \in C^\infty(c,d),$$

then f vanishes almost everywhere on (c,d), which is denoted by $f \overset{a.e.}{=} 0$, on (c,d).

Proof. Assume $f > 0$ on a set $I \subset (c,d)$, with positive measure. Denoting by C_I the characteristic function of the set I, we have $f^{p-1}C_I \in L^{p'}(c,d)$, where $\frac{1}{p} + \frac{1}{p'} = 1$. Indeed,

$$\int_c^d \left[(f(x))^{p-1} C_I(x)\right]^{p'} dx = \int_I \left[(f(x))^{p-1}\right]^{p'} dx = \int_I (f(x))^p dx.$$

Note that the latter integral is well defined, since $f \in C^p[c,d]$. Taking into account that $C^\infty[c,d]$ is dense in $L^{p'}[c,d]$, i.e., for every $g \in L^{p'}[c,d]$ there exists a sequence $(\varphi_k)_{k\in\mathbb{N}} \subset C^\infty[c,d]$, such that $g = \lim_{k\to\infty} \varphi_k$, one obtains

$$\int_c^d (f(x))^p dx = \int_c^d f(x)(f(x))^{p-1} C_I(x)dx = \lim_{k\to\infty} \int_c^d f(x)\varphi_k(x)dx = 0,$$

that contradicts the assumption $f(x) > 0$, $x \in I$. ■

Lemma 2.2.7. *Let $f \in C^p[c,d]$, $1 < p < \infty$. If*

$$\int_c^d f(x)\varphi^{(m)}(x)dx = 0, \qquad \varphi \in C^\infty_{m-1}[c,d], \tag{2.2.18}$$

then there exists a polynomial $p_f \in \mathbb{P}_{m-1}$ such that $f \overset{a.e.}{=} p_f$, on (c,d), where

$$C^\infty_{m-1}[c,d] = \left\{ f \in C^\infty[c,d] \mid f^{(j)}(c) = f^{(j)}(d) = 0, \quad j = 0, ..., m-1 \right\}.$$

Proof. We construct $p_f \in \mathbb{P}_{m-1}$ such that

$$\langle f - p_f, e_i \rangle_2 = 0, \qquad i = 0, ..., m-1.$$

Taking into account Lemma 2.2.6, it is sufficient to prove that $g = f - p_f$ satisfies the condition

$$\langle g, \psi \rangle_2 = 0, \qquad \psi \in C^\infty[c,d].$$

Let $\psi \in C^\infty[c,d]$ and construct the polynomial $p_\psi \in \mathbb{P}_{m-1}$, such that

$$\langle \psi - p_\psi, e_i \rangle_2 = 0, \qquad i = 0, ..., m-1.$$

Let

$$\varphi(x) = \int_c^x \tfrac{(x-t)^{m-1}}{(m-1)!}(\psi - p_\psi)(t)dt.$$

We have

$$\varphi^{(j)}(c) = 0, \qquad j = 0, ..., m-1$$

and from the orthogonality condition $\psi - p_\psi \perp \mathbb{P}_{m-1}$, we also get

$$\varphi^{(j)}(d) = 0, \qquad j = 0, ..., m-1.$$

Thus, it follows $\varphi \in C^\infty_{m-1}[c,d]$. Using $f - p_f \perp \mathbb{P}_{m-1}$ and $\psi - p_\psi \perp \mathbb{P}_{m-1}$, one obtains

$$\langle f - p_f, \psi \rangle_2 = \langle f - p_f, \psi - p_\psi \rangle_2 = \langle f, \psi - p_\psi \rangle_2 = \left\langle f, \varphi^{(m)} \right\rangle_2 = 0,$$

according to (2.2.18). Hence,

$$f - p_f \overset{a.e.}{=} 0, \text{ on } (c,d),$$

which implies

$$f \overset{a.e.}{=} p_f, \text{ on } (c,d).$$

■

Remark 2.2.8. The problem

$$(PSIP) \text{ find } s \in U(y) \text{ such that } \left\|s^{(m)}\right\|_2 = \inf_{u \in U(y)} \left\|u^{(m)}\right\|_2$$

is equivalent to *the best approximation problem* in $L^2[a,b]$, i.e.,

$$(BAP) \text{ find } \sigma \in U^{(m)}(y) \text{ such that } \|\sigma\|_2 = \inf_{v \in U^{(m)}(y)} \|v\|_2,$$

where

$$U^{(m)}(y) = \left\{ v \mid v = u^{(m)}, \ \ u \in U(y) \right\}.$$

Theorem 2.2.9. (Existence theorem). *If the functionals $\lambda_i \in \Lambda$, $i = 1, ..., n$ are bounded and $U(y)$ given in (2.2.17) is a non-empty set then (PSIP) has a solution.*

Proof. Using the equivalence from Remark 2.2.8 and taking into account that the (BAP) has a unique solution if $U^{(m)}(y) \subset L^2[a, b]$ is: 1) non-empty, 2) convex and 3) closed, we have to check that $U^{(m)}(y)$ verifies these three conditions.

1) Since $U(y)$ was assumed to be non-empty, $U^{(m)}(y)$ is also non-empty.

2) The functionals λ_i, $i = 1, ..., n$ are linear. Thus, for $u_1, u_2 \in U(y)$ and $\alpha \in [0, 1]$, we have:

$$\begin{aligned}\lambda_i(\alpha u_1 + (1-\alpha)u_2) &= \alpha\lambda_i(u_1) + (1-\alpha)\lambda_i(u_2) \\ &= \alpha y_i + (1-\alpha)y_i = y_i.\end{aligned}$$

So, $\alpha u_1 + (1-\alpha)u_2 \in U(y)$ which implies that $U(y)$ is convex. Since, $U(y)$ is convex and D^m is linear it follows that $U^{(m)}(y)$ is also a convex set.

3) Let $(g_k)_{k\in\mathbb{N}}$ be a sequence in $U^{(m)}(y)$ that converges to a function $g \in L^2[a, b]$. We have to prove that $g \in U^{(m)}(y)$. We show that there exists a polynomial $p \in \mathbb{P}_{m-1}$ such that the function

$$f = p + h,$$

with

$$h(x) = \int_a^x \frac{(x-t)^{m-1}}{(m-1)!} g(t)dt,$$

belongs to $U(y)$. In this case

$$f^{(m)} = g \in U^{(m)}(y).$$

Since λ_i, $i = 1, ..., n$ are bounded, $U(y)$ is a closed set. Thus, it is sufficient to show that f is the limit of a sequence of functions from $U(y)$, in the norm $\|\cdot\|_{m,2}$ of $H^{m,2}[a, b]$. Assume that $f_k \in U(y)$ are such that

$$g_k = f_k^{(m)}, \qquad k \in \mathbb{N}.$$

It follows that

$$f_k = p_k + h_k,$$

with $p_k \in \mathbb{P}_{m-1}$ and

$$h_k(x) = \int_a^x \frac{(x-t)^{m-1}}{(m-1)!} g_k(t)dt.$$

The sequence $(h_k)_{k\in\mathbb{N}}$ converges to h. In order to find the limit of the sequence $(p_k)_{k\in\mathbb{N}}$, we assume first that there exist m linear independent functionals $\tilde{\lambda}_i \in \Lambda$, $i = 1, ..., m$, on $\mathbb{P}_{m-1}$. In particular, we assume that $\tilde{\lambda}_i$ are such that the matrix

$$A = (\lambda_i(v_j))_{i,j=1,...,m}, \text{ with } v_j(x) = (x-a)^{j-1},$$

is non-singular. Then,

$$\begin{bmatrix} p_k(a) \\ \vdots \\ p_k^{(m-1)}(a) \end{bmatrix} = A^{-1} \begin{bmatrix} \tilde{\lambda}_1 p_k \\ \vdots \\ \tilde{\lambda}_m p_k \end{bmatrix}.$$

Since $\lambda_i(f_k) = y_i$ are bounded and $(\lambda_i(h_k))_{k\in\mathbb{N}}$ converges to $\lambda_i(h)$, $i = 1, ..., n$ it follows that $\lambda_i(p_k)$, $i = 1, ..., m$ are uniformly bounded. Hence, each of the sequences $(p_k^{(j)}(a))_{k\in\mathbb{N}}$, $j = 0, ..., m-1$, is uniformly bounded, which implies that each of the m sequences contains a convergent subsequence $(p_{k_\nu}^{(j)}(a))_{k_\nu\in\mathbb{N}}$. Let

$$p^{(j)}(a) = \lim_{k_\nu\to\infty} p_{k_\nu}^{(j)}(a), \qquad j = 0, ..., m-1.$$

These limits define a polynomial $p \in \mathbb{P}_{m-1}$. The sequence $(f_k)_{k\in\mathbb{N}}$ with $f_k = p_k + h_k$, converges to $f = p + h$, which completes the proof in this case.

Assume now that there exist only $d < m$ linear independent functionals $\lambda_i \in \Lambda$, $i = 1, ..., d$, on $\mathbb{P}_{m-1}$, i.e., the rank of the matrix

$$(\lambda_i(v_j))_{i=1,...,n;j=1,...,m}$$

is equal to d. Let $\tilde{\lambda}_i$, $i = 1, ..., d$ and v_j, $j = 1, ..., m$ be a renumbering of v_j, $j = 1, ..., m$ such that the rank of the matrix $(\lambda_i(v_j))_{i,j=1,...,d}$ is d. According to Lemma 2.2.5, one can construct the functionals $\tilde{\lambda}_i$, $i = d+1, ..., m$, on $H^{m,2}[a, b]$, such that

$$\tilde{\lambda}_i(v_j) = \delta_{ij}, \qquad i = d+1, ..., m;\ j = 1, ..., m.$$

It follows that the functionals $\tilde{\lambda}_i$, $i = 1, ..., m$ are linear independent on $\mathbb{P}_{m-1}$ and the proof is reduced to the previous case. Note that in the representation of f_k, the polynomials p_k have to be chosen such that

$$\lambda_i(p_k) = 0, \qquad i = d+1, ..., m.$$

Thus, the (BAP) has an unique solution. This implies that the equivalent (PSIP) has a solution. ■

Theorem 2.2.10. *If s_1, s_2 are two solutions of (PSIP) then*

$$s_1 - s_2 \in \mathbb{P}_{m-1}.$$

Proof. As the equivalent (BAP) has a unique solution σ, it follows that

$$s_1^{(m)} = s_2^{(m)} = \sigma,$$

or

$$(s_1 - s_2)^{(m)} = 0,$$

which is equivalent to

$$s_1 - s_2 \in \mathbb{P}_{m-1}.$$

■

Theorem 2.2.11. *A (PISP) has an unique solution if and only if*

$$\mathbb{P}_{m-1} \cap U(0) = \{0\}. \tag{2.2.19}$$

Proof. Let s_1 be a solution of (PSIP). Suppose that there exists

$$v \in \mathbb{P}_{m-1} \cap U(0), \qquad v \neq 0.$$

Then

$$s_2 = s_1 + v$$

is also a solution of (PSIP), since

$$\lambda_i(s_2) = \lambda_i(s_1) + \lambda_i(v) = \lambda_i(s_1) = y_i$$

and

$$\left\| s_2^{(m)} \right\|_2 = \left\| s_1^{(m)} \right\|_2.$$

As $s_2 \neq s_1$, it means that (PSIP) has at least two solutions and the necessity is proved. For the sufficiency, we assume that s_1 and s_2 are solutions of (PSIP). Then, by Theorem 2.2.10 it follows that $s_1 - s_2 \in \mathbb{P}_{m-1}$ and by

$$\lambda_i(s_1 - s_2) = \lambda_i(s_1) - \lambda_i(s_2) = y_i - y_i = 0$$

we get that $s_1 - s_2 \in U(0)$. Thus,

$$s_1 - s_2 \in \mathbb{P}_{m-1} \cap U(0)$$

and from (2.2.19) it follows that

$$s_1 = s_2.$$

■

By Theorem 2.2.9 we have that the existence of a solution of (PSIP) is characterized by the set of functionals Λ. The question is whether the set Λ can also characterize the uniqueness of the solution. The answer is given by the following result.

Theorem 2.2.12. *If Λ contains at least m functionals of Hermite type, then*

$$\mathbb{P}_{m-1} \cap U(0) = \{0\}.$$

Proof. Consider $p \in \mathbb{P}_{m-1} \cap U(0)$ and let $M = \{\lambda_1, ..., \lambda_r\} \subset \Lambda$, with $r \geq m$, be a set of Hermite functionals. Since $p \in U(0)$, we have

$$\lambda_i(p) = 0, \qquad i = 1, ..., r.$$

Thus, p has at least r zeros ($r \geq m$). Taking into account that $p \in \mathbb{P}_{m-1}$, it follows that $p = 0$. ■

The next theorem states one of the most important properties of the spline interpolant, namely the orthogonality property.

Theorem 2.2.13. *The function $s \in U(y)$ is a solution of (PSIP) if and only if*

$$\left\langle s^{(m)}, g^{(m)} \right\rangle_2 = 0, \qquad g \in U(0). \tag{2.2.20}$$

Proof. It is used again the equivalence between (PSIP) and (BAP) (Remark 2.2.8). Denote by $\sigma := s^{(m)}$. Since $U^{(m)}(0)$ is a linear subspace of $L^2[a, b]$ and

$U^{(m)}(y)$ is a translation of $U^{(m)}(0)$, it follows that $\sigma \in U^{(m)}(y)$ is a solution of the (BAP) if and only if $\sigma \perp U^{(m)}(0)$, or

$$\langle \sigma, g \rangle_2 = 0, \qquad g \in U^{(m)}(0),$$

i.e.,

$$\left\langle s^{(m)}, g^{(m)} \right\rangle_2 = 0, \qquad g \in U(0).$$

■

According to Theorem 2.2.13, the orthogonality property (2.2.20) completely characterizes the solution of (PSIP), so it is natural to investigate the set

$$\mathcal{S}(\Lambda) = \left\{ f \in H^{m,2}[a,b] \mid \left\langle f^{(m)}, g^{(m)} \right\rangle_2 = 0 \quad g \in U(0) \right\}.$$

Theorem 2.2.14. *The set $\mathcal{S}(\Lambda)$ is a closed linear subspace of $H^{m,2}[a,b]$.*

Proof. Let $f_1, f_2 \in \mathcal{S}(\Lambda)$, i.e.,

$$\left\langle f_i^{(m)}, g^{(m)} \right\rangle_2 = 0, \quad g \in U(0), \qquad \text{for } i = 1, 2.$$

Taking $f = \alpha_1 f_1 + \alpha_2 f_2$, we have

$$\left\langle f^{(m)}, g^{(m)} \right\rangle_2 = \alpha_1 \left\langle f_1^{(m)}, g^{(m)} \right\rangle_2 + \alpha_2 \left\langle f_2^{(m)}, g^{(m)} \right\rangle_2 = 0, \quad \text{for } g \in U(0),$$

hence $f \in \mathcal{S}(\Lambda)$. It follows that $\mathcal{S}(\Lambda)$ is a linear subspace of $H^{m,2}[a,b]$.

In order to show that $\mathcal{S}(\Lambda)$ is closed, we consider a sequence $(f_k)_{k \in \mathbb{N}} \subset \mathcal{S}(\Lambda)$ that converges to $f \in H^{m,2}[a,b]$. We have to prove that $f \in \mathcal{S}(\Lambda)$. Since $f_k \in \mathcal{S}(\Lambda)$, we have

$$\left\langle f_k^{(m)}, g^{(m)} \right\rangle_2 = 0, \qquad g \in U(0),$$

and from

$$\lim_{k \to \infty} f_k = f, \quad \text{in } H^{m,2}[a,b],$$

one obtains

$$\lim_{k \to \infty} \langle f_k - f, g \rangle_{m,2} = 0,$$

which implies

$$\lim_{k \to \infty} \left\langle (f_k - f)^{(m)}, g^{(m)} \right\rangle_2 = 0.$$

Consequently,

$$\left\langle f^{(m)}, g^{(m)} \right\rangle_2 = \lim_{k \to \infty} \left[\left\langle f^{(m)} - f_k^{(m)}, g^{(m)} \right\rangle_2 + \left\langle f_k^{(m)}, g^{(m)} \right\rangle_2 \right] = 0, \quad \text{for } g \in U(0),$$

hence, $f \in \mathcal{S}(\Lambda)$. ■

Remark 2.2.15. The orthogonality property (2.2.20) implies that

$$\left\langle p^{(m)}, g^{(m)} \right\rangle_2 = 0, \qquad \text{for } p \in \mathbb{P}_{m-1},$$

i.e.,

$$\mathbb{P}_{m-1} \subset \mathcal{S}(\Lambda).$$

Definition 2.2.16. *The space $\mathcal{S}(\Lambda)$ is called the space of spline functions interpolating $U(y)$.*

Theorem 2.2.17. *Let $(v_i)_{i=1,...,d}$ be a basis of the space $\mathbb{P}_{m-1} \cap U(0)$, and s_i a solution of (PSIP) that interpolates the set*

$$U_i = \left\{ f \in H^{m,2}[a,b] \mid \lambda_i(f) = \delta_{ij}, j = 1, ..., n \right\},$$

for $i = 1, ..., n$. Then, the set $(s_i)_{i=1,...,n} \cup (v_j)_{j=1,...,d}$ is a basis for $S(\Lambda)$.

Proof. Let $s \in \mathcal{S}(\Lambda)$ and

$$h = s - \sum_{i=1}^{n} s_i \lambda_i(s). \tag{2.2.21}$$

Then $h \in \mathcal{S}(\Lambda)$ and $h \in U(0)$. From (2.2.20) it follows that

$$\left\langle h^{(m)}, h^{(m)} \right\rangle_2 = 0.$$

Thus, $h^{(m)} = 0$, implying $h \in \mathbb{P}_{m-1}$. Hence, we have $h \in \mathbb{P}_{m-1} \cap U(0)$. Since $(v_j)_{j=1,...,d}$ is a basis of $\mathbb{P}_{m-1} \cap U(0)$, we can write

$$h = \sum_{j=1}^{d} c_j v_j$$

and using (2.2.21), we obtain

$$s = \sum_{i=1}^{n} s_i \lambda_i(s) + \sum_{j=1}^{d} c_j v_j.$$

In order to show that the elements of the set $(s_i)_{i=1,...,n} \cup (v_j)_{j=1,...,d}$ are linearly independent, we start with

$$\sum_{i=1}^{n} a_i s_i + \sum_{j=1}^{d} v_j b_j = 0. \tag{2.2.22}$$

We have

$$\lambda_k\left(\sum_{i=1}^{n} a_i s_i + \sum_{j=1}^{d} v_j b_j\right) = a_k = 0, \quad k = 1, ..., n,$$

since $\lambda_k(s_i) = \delta_{ki}$ and $\lambda_k(v_j) = 0$, $j = 1, ..., d \quad (v_j \in U(0))$. Substituting $a_k = 0$ in (2.2.22), it yields

$$\sum_{j=1}^{d} v_j b_j = 0.$$

Because v_j, $j = 1, ..., d$ are linearly independent, we obtain $b_j = 0$, $j = 1, ..., d$. Hence, (2.2.22) holds if and only if $a_i = 0$, $i = 1, ..., n$ and $b_j = 0$, $j = 1, ..., d$. This implies that $s_1, ..., s_n, v_1, ..., v_d$ are linearly independent. ■

Remark 2.2.18. If

$$\mathbb{P}_{m-1} \cap U(0) = \{0\},$$

(i.e., (PSIP) has a unique solution), then

$$\dim \mathcal{S}(\Lambda) = n$$

and $s_1, ..., s_n$ is a basis of $\mathcal{S}(\Lambda)$.

The functions s_i, $i = 1, ..., n$ are called *fundamental interpolation splines* or *cardinal splines*, due to the property that

$$\lambda_k(s_i) = \delta_{ki}, \qquad k, i = 1, ..., n.$$

Also a very important property for the solutions of a (PSIP) is their structural characterization.

Obviously, the structure of a solution of (PSIP) essentially depends on the set Λ that defines the interpolatory set $U(y)$.

In the sequel we suppose that Λ is a set of Birkhoff type functionals,

$$\Lambda = \left\{\lambda_{ij} \mid \lambda_{ij} f = f^{(j)}(x_i), \;\; i = 1, ..., k, \;\; j \subset I_i\right\}, \tag{2.2.23}$$

where $a \leq x_1 < ... < x_k \leq b$ is a partition of the interval $[a, b]$, $r_1, ..., r_k \in \mathbb{N}$, with $r_i \leq m - 1$ and $I_i \subseteq \{0, 1, ..., r_i\}$, $i = 1, ..., k$.

Theorem 2.2.19. (Structural Characterization) *Let Λ be a set of Birkhoff type functionals given by (2.2.23), $y \in \mathbb{R}^n$ and let $U(y)$ be the corresponding interpolatory set. The function $s \in U(y)$ is a solution of (PSIP) if and only if:*

1) $s^{(2m)}(x) = 0, \quad x \in [x_1, x_k] \setminus \{x_1, ..., x_k\}$,
2) $s^{(m)}(x) = 0, \quad x \in (a, x_1) \cup (x_k, b)$,
3) $s^{(2m-1-\mu)}(x_i - 0) = s^{(2m-1-\mu)}(x_i + 0), \quad \mu \in \{0, 1, ..., m-1\} \setminus I_i$, *for* $i = 1, ..., k$.

Proof. Let $s \in U(y)$ be a solution of (*PSIP*).

1) Let I be one of the intervals (x_i, x_{i+1}), $i = 0, 1, ..., k$ ($x_0 = a$ and $x_{k+1} = b$). Considering $\varphi \in C^{\infty}_{m-1}(I)$, we define the function

$$g(x) = \begin{cases} \varphi(x), & x \in I \\ 0, & x \in [a, b] \setminus I. \end{cases}$$

Since $g \in U(0)$ and s is a solution of (*PSIP*), we have (the orthogonality property):

$$\int_a^b s^{(m)}(x) g^{(m)}(x)\, dx = 0.$$

As $g^{(m)} = \varphi^{(m)}$ on I, we obtain

$$\int_a^b s^{(m)}(x)\, g^{(m)}(x)\, dx = \int_I s^{(m)}(x)\, \varphi^{(m)}(x)\, dx = 0.$$

Using Lemma 2.2.7, it follows

$$s^{(m)} \overset{a.e.}{=} p \in \mathbb{P}_{m-1}, \text{ on } I.$$

But $s \in H^{m,2}[a, b]$, hence

$$s(x) = p_s(x) + \int_a^x \frac{(x-t)^{m-1}}{(m-1)!} s^{(m)}(t)\, dt, \quad p_s \in \mathbb{P}_{m-1}.$$

Consequently, $s \in \mathbb{P}_{2m-1}$ on the interval I. Thus $s^{(2m)}(x) = 0$, $x \in I$, which implies 1).

2) Now, we denote $I := [a, x_1) \neq \emptyset$. Since $s \in \mathbb{P}_{2m-1}$ on I, we have $s^{(m)} \in C(I)$. Consider the function

$$\tilde{s}(x) = \begin{cases} p(x), & x \in [a, x_1], \\ s(x), & x \in (x_1, b], \end{cases}$$

where $p \in \mathbb{P}_{m-1}$ is such that

$$p^{(j)}(x_1) = s^{(j)}(x_1), \qquad j = 0, 1, ..., m-1.$$

Notice first that $\tilde{s} \in U(y)$. Indeed,

$$\lambda_i(\tilde{s}) = \lambda_i(s), \qquad i = 1, ..., n.$$

Assume that there exists $\xi \in I$, such as $s^{(m)}(\xi) \neq 0$. Since $s^{(m)}$ is continuous on I, there exists a neighborhood $V(\xi)$ of ξ, such that

$$s^{(m)}(x) \neq 0, \qquad x \in V(\xi).$$

This leads to the inequality

$$\left\|\tilde{s}^{(m)}\right\|_2 < \left\|s^{(m)}\right\|_2,$$

which contradicts the hypothesis that s is a solution of $(PSIP)$. Hence,

$$s^{(m)}(x) = 0, \qquad x \in I.$$

The case $I = (x_k, b)$ can be treated similarly.

3) Let $g \in H^{m,2}[a,b]$. One has

$$\left\langle s^{(m)}, g^{(m)} \right\rangle_2 = \int_a^b s^{(m)}(x)\, g^{(m)}(x)\, dx = \sum_{i=0}^{k} \int_{x_i}^{x_{i+1}} s^{(m)}(x)\, g^{(m)}(x)\, dx.$$

Integrating by parts in the above relation, it yields

$$\begin{aligned} \left\langle s^{(m)}, g^{(m)} \right\rangle_2 &= \int_a^{x_1} s^{(m)}(x)\, g^{(m)}(x)\, dx + \int_{x_k}^{b} s^{(m)}(x)\, g^{(m)}(x)\, dx \\ &+ \sum_{i=1}^{k-1} (-1)^m \int_{x_i}^{x_{i+1}} s^{(2m)}(x)\, g(x)\, dx \\ &+ \sum_{i=1}^{k} \sum_{j=0}^{m-1} (-1)^{m-j} g^{(j)}(x_i) \left[s^{(2m-1-j)}(x_i + 0) - s^{(2m-1-j)}(x_i - 0) \right]. \end{aligned} \tag{2.2.24}$$

If $x_1 = a$, the first integral is missing and in x_1 the saltus is considered to be the limit on the right in x_1. The case $x_k = b$ is analogously treated.

Consider now an $\varepsilon > 0$ such that $(x_i - \varepsilon, x_i + \varepsilon) \cap \{x_i\}_{i=1,...,k} = \{x_i\}$, and $\mu \in \{0, 1, ..., m-1\} \setminus I_i$. We construct $h \in \mathbb{P}_{3m-1}$ that satisfies conditions

$$\begin{aligned} &h^{(j)}(x_i - \varepsilon) = h^{(j)}(x_i + \varepsilon) = 0, \\ &h^{(j)}(x_i) = \delta_{j\mu}, \end{aligned}$$

for $j = 0, 1, ..., m-1$. We define the function g by

$$g(x) = \begin{cases} h(x), & x \in J = [x_i - \varepsilon, x_i + \varepsilon] \cap [a, b] \\ 0, & \text{otherwise.} \end{cases}$$

We have $g \in U(0)$, hence

$$\int_a^b s^{(m)}(x)\, g^{(m)}(x)\, dx = 0.$$

Taking into account the latter equality and 1), 2), (2.2.24), we obtain

$$\sum_{j=0}^{m-1} (-1)^{m-j} g^{(j)}(x_i) \left[s^{(2m-1-j)}(x_i + 0) - s^{(2m-1-j)}(x_i - 0) \right] = 0, \quad i = 1, ..., k.$$

Since,

$$g^{(j)}(x_i) = \delta_{j\mu},$$

we get

$$s^{(2m-1-\mu)}(x_i + 0) - s^{(2m-1-\mu)}(x_i - 0) = 0, \quad i = 1, ..., k.$$

Conversely, if statements 1)–3) are satisfied, then, from (2.2.24), it follows

$$\left\langle s^{(m)}, g^{(m)} \right\rangle_2 = 0, \qquad g \in U(0),$$

thus s is solution of $(PSIP)$. ■

Remark 2.2.20. The Structural Characterization Theorem states that the solution s of $(PSIP)$ is a polynomial of $2m-1$ degree on each interior interval (x_i, x_{i+1}) and it is a polynomial of $m-1$ degree on the intervals $[a, x_1)$ and $(x_k, b]$. Furthermore, the derivative of order $2m-1-\mu$ is continuous in x_i if the value of the μ-th order derivative of f, in x_i, does not belong to Λ.

A solution of a $(PSIP)$ is also called *(natural) spline* of order $2m-1$.

In conclusion, according to Theorems 2.2.13 and 2.2.19 we can state the following result.

Theorem 2.2.21. *If Λ is a set of Birkhoff type linear functionals, $U(y)$, $y \in \mathbb{R}^n$ is the corresponding interpolation set and $\mathcal{S}(\Lambda)$ is the set of splines that interpolate $U(y)$, then the following three statements are equivalent:*

$$(A) \qquad s \in \mathcal{S}(\Lambda) \Longleftrightarrow \left\|s^{(m)}\right\|_2 = \inf_{u \in U(y)} \left\|u^{(m)}\right\|_2$$

$$(B) \qquad s \in \mathcal{S}(\Lambda) \Longleftrightarrow \left\langle s^{(m)}, g^{(m)} \right\rangle_2 = 0,\ g \in U(0)$$

$$(C) \qquad s \in \mathcal{S}(\Lambda) \Longleftrightarrow \begin{cases} 1)\ s^{(2m)}(x) = 0, \quad x \in [x_1, x_k] \setminus \{x_i\}_{i=1,...,k} \\ 2)\ s^{(m)}(x) = 0, \quad x \in [a, x_1) \cup (x_k, b] \\ 3)\ s^{(2m-1-\mu)}(x_i + 0) - s^{(2m-1-\mu)}(x_i - 0) = 0, \\ \qquad \text{with } \mu \in \{0, 1, ..., m-1\} \setminus I_i, i = 1, ..., k. \end{cases}$$

Consequently, each of the three statements above can be used as definition of the solution s of (PSIP) and the other two can be proved as theorems. In the present case, (A) was taken as definition (Definition 2.2.3), while (B) was proved as The Orthogonality Theorem (Theorem 2.2.13), respectively, (C) as The Structural Characterization Theorem (Theorem 2.2.19).

Suppose now that (PSIP) has an unique solution.

Theorem 2.2.22. *Consider $y = (y_1, ..., y_n) \in \mathbb{R}^n$ and let s_i, $i = 1, ..., n$ be the fundamental interpolation splines. Then the function*

$$s_y = \sum_{i=1}^{n} s_i y_i$$

is the solution of (PSIP) corresponding to the set $U(y)$.

Proof. One has

$$\lambda_k(s_y) = y_k, \qquad k = 1, ..., n,$$

thus $s_y \in U(y)$. Since s_i are the solutions of $(PSIP)$ that interpolate the sets U_i, $i = 1, ..., n$, respectively, by Theorem 2.2.13 it follows that

$$\left\langle s_i^{(m)}, g^{(m)} \right\rangle_2 = 0, \qquad g \in U(0),$$

for all $i = 1, ..., n$. Consequently,

$$\left\langle s_y^{(m)}, g^{(m)} \right\rangle_2 = \sum_{i=1}^{n} y_i \left\langle s_i^{(m)}, g^{(m)} \right\rangle_2 = 0, \quad g \in U(0),$$

hence,

$$s_y \in \mathcal{S}(\Lambda).$$

■

Remark 2.2.23. Theorem 2.2.22 can be interpreted as follows: For a given $f \in H^{m,2}[a,b]$ and a given set of functionals Λ, the function

$$Sf = \sum_{i=1}^{n} s_i \lambda_i (f) \tag{2.2.25}$$

is a spline that interpolates f with respect to Λ, i.e.,

$$\lambda_i (Sf) = \lambda_i (f), \quad i = 1, ..., n.$$

Furthermore, Sf is the smoothest function in $H^{m,2}[a,b]$ that interpolates f, in the sense that

$$\left\| (Sf)^{(m)} \right\|_2 \to \min .$$

Remark 2.2.24. From (2.2.25) one can notice that $S : H^{m,2}[a,b] \to \mathcal{S}(\Lambda)$ is a linear and idempotent operator, hence it is a projector.

Definition 2.2.25. *The operator $S : H^{m,2}[a,b] \to \mathcal{S}(\Lambda)$ is called* polynomial spline interpolation operator *suitable to Λ.*

Next, there will be considered the spline operator S for particular sets of functionals Λ.

2.2.3 Spline operators of Lagrange type

For a given function $f : [a,b] \to \mathbb{R}$ and $x_i \in [a,b]$, $i = 1, ..., n$, we consider

$$\Lambda := \Lambda_L = \{\lambda_i \mid \lambda_i(f) = f(x_i), \quad i = 1, ..., n\}.$$

By Theorem 2.2.12 it follows that if $n \geq m$ then for every $f \in H^{m,2}[a,b]$ the interpolation spline function $S_L f$ exists and is unique.

Definition 2.2.26. *For $\Lambda := \Lambda_L$, the corresponding spline operator S_L is called* the spline operator of Lagrange type.

For determining S_L one can use Theorem 2.2.19, but with the remark that the third condition from this theorem becomes $3')$ $S_L f \in C^{2m-2}[a,b]$. Writing the function Sf as

$$(S_L f)(x) = \sum_{i=0}^{m-1} a_i x^i + \sum_{j=1}^{n} b_j (x - x_j)_+^{2m-1}, \tag{2.2.26}$$

it follows that the conditions 1), 2) for $x \in [a, x_1)$ and 3′) are verified. Indeed,

$$(S_L f)(x) = \sum_{i=0}^{m-1} a_i x^i + \sum_{j=1}^{k} b_j (x - x_j)^{2m-1},$$

for $x \in (x_k, x_{k+1})$, $k = 1, ..., n-1$,

$$(S_L f)(x) = \sum_{i=0}^{m-1} a_i x^i, \quad x \in [a, x_1),$$

and

$$(S_L f)^{(\nu)}(x_j - 0) = (S_L f)^{(\nu)}(x_j + 0), \ \ \nu = 0, ..., 2m-2, \ j = 1, ..., n.$$

So, $S_L f \in \mathbb{P}_{2m-1}$ on each interval (x_j, x_{j+1}), $j = 1, ..., m-1$, $S_L f \in \mathbb{P}_{m-1}$ on $[a, x_1)$ and $S_L f \in C^{2m-2}[a, b]$. On the interval $(x_n, b]$ we have

$$(S_L f)(x) = \sum_{i=0}^{m-1} a_i x^i + \sum_{j=1}^{n} b_j (x - x_j)^{2m-1},$$

so $S_L f \in \mathbb{P}_{2m-1}$. Hence, the condition 2) is not verified on this interval.

Writing the polynomial $S_L f$ in the Taylor's form, i.e.,

$$(S_L f)(x) = \sum_{k=0}^{2m-1} \frac{(x-\alpha)^k}{k!} (S_L f)^{(k)}(\alpha), \ \ \alpha > x_n,$$

it follows that it reduces to a polynomial of $m-1$ degree on $(x_n, b]$, if we suppose that

$$(S_L f)^{(p)}(\alpha) = 0, \ p = m, ..., 2m-1. \tag{2.2.27}$$

Consequently, if the conditions (2.2.27) are verified then all the three properties of a spline function are fulfilled.

Regarding the expression from (2.2.26), we notice that the function $S_L f$ depends on $m + n$ parameters, a_i, $b_j \in \mathbb{R}$, $i = 0, ..., m-1$; $j = 1, ..., n$. We also notice that (2.2.27) and the interpolation conditions

$$(S_L f)(x_i) = f(x_i), \ i = 1, ..., n$$

provide a $(m+n)\times(m+n)$ linear algebraic system:

$$\begin{cases} (S_Lf)^{(p)}(\alpha)=0, & p=m,...,2m-1 \\ (S_Lf)(x_i)=f(x_i), & i=1,...,n. \end{cases} \tag{2.2.28}$$

For $n\geq m$, the function S_Lf may also be written in the form

$$S_Lf=\sum_{k=1}^{n}s_kf(x_k),$$

where s_k, $k=1,...,n$ are the fundamental interpolation spline functions. So, the determination of S_Lf reduces to the problem of finding the functions s_k, $k=1,...,n$. For doing this we write these functions in some forms corresponding to (2.2.26), i.e.,

$$s_k(x)=\sum_{i=0}^{m-1}a_i^kx^i+\sum_{j=1}^{n}b_j^k(x-x_j)_+^{2m-1},\ k=1,...,n,$$

with a_i^k, $i=0,...,m-1$ and b_j^k, $j=1,...,n$ obtained as the solutions of the linear algebraic system:

$$\begin{cases} s_k^{(p)}(\alpha)=0, & p=m,...,2m-1 \text{ and } \alpha>x_n \\ s_k(x_\nu)=\delta_{k\nu}, & \nu=1,...,n \end{cases} \tag{2.2.29}$$

for $k=1,...,n$. We remark that the matrices of all these systems coincide and only the free terms are different.

Definition 2.2.27. *The formula*

$$f=S_Lf+R_Lf$$

is called the spline interpolation formula of Lagrange type*, where* R_L *is* the remainder operator.

We have $\mathrm{dex}(S_L)=m-1$ and applying the Peano's theorem, we obtain

$$(R_Lf)(x)=\int_a^b\varphi_L(x,t)f^{(m)}(t)dt,$$

where

$$\varphi_L(x,t)=\frac{(x-t)_+^{m-1}}{(m-1)!}-\sum_{i=1}^{n}s_i(x)\frac{(x_i-t)_+^{m-1}}{(m-1)!}.$$

If $f^{(m)}$ is continuous on $[a,b]$, then

$$|(R_L f)(x)| \leq \left\|f^{(m)}\right\|_\infty \int_a^b |\varphi_L(x,t)|dt.$$

Example 2.2.28. Let $f \in C[0,1]$ and

$$\Lambda_L(f) = \left\{ f\left(\tfrac{i}{n}\right) \middle| \, i = 0, ..., n \right\}.$$

We are looking for a linear spline interpolation formula.

We have

$$f = S_1 f + R_1 f,$$

where

$$(S_1 f)(x) = \sum_{i=0}^{n} s_i(x) f\left(\tfrac{i}{n}\right)$$

and

$$(R_1 f)(x) = \int_0^1 \varphi_1(x,t) f'(t) dt,$$

with

$$\varphi_1(x,t) = (x-t)_+^0 - \sum_{i=0}^{n} s_i(x) \left(\tfrac{i}{n} - t\right)_+^0.$$

We have to find the functions s_i, $i = 0, ..., n$. We notice that

$$s_i(x) = a_0^i + \sum_{j=0}^{n} b_j^i \left(x - \tfrac{j}{n}\right)_+, \; i = 0, ..., n,$$

and the coefficients $a_0^i, b_0^i, ..., b_n^i$, $i = 0, ..., n$ are provided by the conditions:

$$\begin{cases} s_i\left(\frac{k}{n}\right) = \delta_{ik}, & k = 0, ..., n \\ s_i'(\alpha) = 0, & \alpha > 1 \text{ (we take } \alpha = 2). \end{cases}$$

We obtain

$$\begin{aligned}
s_0(x) &= 1 - nx + n\left(x - \tfrac{1}{n}\right)_+ \\
s_1(x) &= nx - 2n\left(x - \tfrac{1}{n}\right)_+ + n\left(x - \tfrac{2}{n}\right)_+ \\
s_2(x) &= n\left(x - \tfrac{1}{n}\right)_+ - 2n\left(x - \tfrac{2}{n}\right)_+ + n\left(x - \tfrac{3}{n}\right)_+ \\
&... \\
s_{n-1}(x) &= n\left(x - \tfrac{n-2}{n}\right)_+ - 2n\left(x - \tfrac{n-1}{n}\right)_+ + n\left(x-1\right)_+ \\
s_n(x) &= n\left(x - \tfrac{n-1}{n}\right)_+ - n\left(x-1\right)_+.
\end{aligned}$$

2.2.4 Spline operators of Hermite type

For a function $f \in H^{m,2}[a,b]$ we have the set of Hermite type functionals

$$\Lambda_H = \{\lambda_{kj} \mid \lambda_{kj} f = f^{(j)}(x_k),\ k = 1, ..., n,\ j = 0, ..., r_k\},$$

where $r_k \in \mathbb{N}$, $r_k < m$.

Definition 2.2.29. *The spline operator S corresponding to the set of Hermite type functionals, Λ_H, is called* the spline operator of Hermite type.

The spline function of Hermite type can be written as

$$(S_H f)(x) = \sum_{i=0}^{m-1} a_i x^i + \sum_{k=1}^{n} \sum_{j=0}^{r_k} b_{kj} (x - x_k)_+^{2m-j-1}, \tag{2.2.30}$$

with the parameters a_i, b_{kj} given by conditions 1)–3) from Theorem 2.2.19 and by the interpolation conditions:

$$(S_H f)^{(j)}(x_k) = f^{(j)}(x_k), \quad k = 1, ..., n;\ j = 0, ..., r_k.$$

It is easy to check that $S_H f$, written as in (2.2.30), verifies the conditions 1), 2) on $[a, x_1)$ and 3), from Theorem 2.2.19. For the condition 2) to take place also on $(x_n, b]$, it is sufficient to consider

$$(S_H f)^{(p)}(\alpha) = 0, \quad p = m, ..., 2m-1,$$

that is analogous with (2.2.27). Therefore, the parameters a_i, $i = 0, ..., m-1$ and b_{kj}, $k = 1, ..., n$, $j = 0, ..., r_k$, (their number is $N := n + m + \sum_{k=1}^{n} r_k$), are determined as the solution of the $N \times N$ linear algebraic system:

$$\begin{cases} (S_H f)^{(j)}(x_k) = f^{(j)}(x_k), & k = 1, ..., n;\ j = 0, ..., r_k \\ (S_H f)^{(p)}(\alpha) = 0, & p = m, ..., 2m-1. \end{cases}$$

If $|\Lambda_H| \geq m$, by Theorem 2.2.12, it follows that the spline function $S_H f$ exists and is unique, so the system is compatible.

Also, the function $S_H f$ can be represented using the fundamental spline functions, i.e.,

$$(S_H f)(x) = \sum_{k=1}^{n} \sum_{j=0}^{r_k} s_{kj}(x) f^{(j)}(x_k), \tag{2.2.31}$$

where

$$s_{kj}(x) = \sum_{\mu=0}^{m-1} a_\mu^{kj} x^\mu + \sum_{\mu=1}^{n} \sum_{\nu=0}^{r_k} b_{\mu\nu}^{kj} (x - x_\mu)_+^{2m-\nu-1},$$

for $k = 1, ..., n$ and $j = 0, ..., r_k$. Each fundamental spline function s_{kj} is obtained from the corresponding system of the form:

$$\begin{cases} s_{kj}^{(q)}(x_\nu) = 0, & \nu = 1, ..., n,\ \nu \neq k,\ q = 0, ..., r_\nu, \\ s_{kj}^{(q)}(x_k) = \delta_{jq}, & q = 0, ..., r_k, \\ s_{kj}^{(p)}(\alpha) = 0, & p = m, ..., 2m-1 \text{ and } \alpha > x_n, \end{cases} \tag{2.2.32}$$

for $k = 1, ..., n$ and $j = 0, ..., r_k$. As in case of the Lagrange interpolation, all the systems have the same matrices.

Definition 2.2.30. *Let $\Lambda := \Lambda_H$ and S_H be the corresponding spline operator. The formula*

$$f = S_H f + R_H f,$$

is called the spline interpolation formula of Hermite type*, where R_H is* the remainder operator*.*

Applying the Peano's theorem, we obtain

$$(R_H f)(x) = \int_a^b \varphi_H(x,t) f^{(m)}(t) dt,$$

where

$$\varphi_H(x,t) = \frac{(x-t)_+^{m-1}}{(m-1)!} - \sum_{k=1}^{n} \sum_{j=0}^{r_k} s_{kj}(x) \frac{(x_k - t)_+^{m-j-1}}{(m-j-1)!}.$$

If $f^{(m)}$ is continuous on $[a, b]$ then

$$|(R_H f)(x)| \leq \left\| f^{(m)} \right\|_\infty \int_a^b |\varphi_H(x,t)| dt.$$

Example 2.2.31. Let $f \in C^2[0,1]$ and $\Lambda_H(f) = \left\{ f(0), f'(0), f\left(\frac{1}{2}\right), f(1), f'(1) \right\}$ be given. We have to find the corresponding spline interpolation formula.

We consider the cubic spline function ($m = 2$) with its interpolation formula:

$$f = S_3 f + R_3 f.$$

From (2.2.31) we have

$$(S_3f)(x) = s_{00}(x)f(0) + s_{01}(x)f'(0) + s_{10}(x)f\left(\tfrac{1}{2}\right) + s_{20}(x)f(1) + s_{21}(x)f'(1),$$

with the fundamental interpolation spline functions having the following expressions:

$$s_{kj}(x) = a_0^{kj} + a_1^{kj}x + b_{00}^{kj}x^3 + b_{01}^{kj}x^2 + b_{10}^{kj}\left(x - \tfrac{1}{2}\right)_+^3 + b_{20}^{kj}(x-1)_+^3 + b_{21}^{kj}(x-1)_+^2, \tag{2.2.33}$$

for $(k, j) \in \{(0,0), (0,1), (1,0), (2,0), (2,1)\}$.

In this case the system from (2.2.32), for $\alpha = 2$, becomes:

$$\begin{aligned}
&s_{00}(0) := a_0^{00} = 1 \\
&s'_{00}(0) := a_1^{00} = 0 \\
&s_{00}\left(\tfrac{1}{2}\right) := a_0^{00} + \tfrac{1}{2}a_1^{00} + \tfrac{1}{8}b_{00}^{00} + \tfrac{1}{4}b_{01}^{00} = 0 \\
&s_{00}(1) := a_0^{00} + a_1^{00} + b_{00}^{00} + b_{01}^{00} + \tfrac{1}{8}b_{10}^{00} = 0 \\
&s'_{00}(1) := a_1^{00} + 3b_{00}^{00} + 2b_{01}^{00} + \tfrac{3}{4}b_{10}^{00} = 0 \\
&s''_{00}(2) := 12b_{00}^{00} + 2b_{01}^{00} + 9b_{10}^{00} + 6b_{20}^{00} + 2b_{21}^{00} = 0 \\
&s'''_{00}(2) := b_{00}^{00} + b_{10}^{00} + b_{20}^{00} = 0.
\end{aligned} \tag{2.2.34}$$

By solving this system we get

$$s_{00}(x) = 1 + 10x^3 - 9x^2 - 16\left(x - \tfrac{1}{2}\right)_+^3.$$

The systems that provide the coefficients of the other fundamental spline functions have the same matrix, but the free terms are changing successively in $(0,1,0,0,0,0,0)$, $(0,0,1,0,0,0,0)$, $(0,0,0,1,0,0,0)$, $(0,0,0,0,1,0,0)$. This way, we obtain:

$$\begin{aligned}
s_{01}(x) &= x + 3x^3 - \tfrac{7}{2}x - 4\left(x - \tfrac{1}{2}\right)_+^3, \\
s_{10}(x) &= -16x^3 + 12x^2 + 32\left(x - \tfrac{1}{2}\right)_+^3, \\
s_{20}(x) &= 6x^3 - 3x^2 - 16\left(x - \tfrac{1}{2}\right)_+^3, \\
s_{21}(x) &= -x^3 + \tfrac{1}{2}x^2 + 4\left(x - \tfrac{1}{2}\right)_+^3.
\end{aligned}$$

For the remainder we have:

$$(R_3f)(x) = \int_0^1 \varphi_2(x,t)f''(t)dt,$$

where

$$\varphi_2(x,t) = (x-t)_+ - s_{10}(x)\left(\tfrac{1}{2} - t\right)_+ - s_{20}(x)(1-t) - s_{21}(x).$$

2.2.5 Spline operators of Birkhoff type

For a function $f \in H^{m,2}[a,b]$ the set of Birkhoff type functionals is given by:

$$\Lambda_B = \{\lambda_{kj} \mid \lambda_{kj} f = f^{(j)}(x_k),\ k = 1, ..., n,\ j \in I_k\},$$

for $I_k \subseteq \{0, ..., r_k\}$, $r_k \in \mathbb{N}$, $r_k < m$.

Definition 2.2.32. *The spline operator S_B corresponding to the set of Birkhoff type functionals, Λ_B, is called* the spline operator of Birkhoff type.

If the spline function of Birkhoff type exists, it is of the following form:

$$(S_B f)(x) = \sum_{i=0}^{m-1} a_i x^i + \sum_{k=1}^{n} \sum_{j \in I_k} b_{kj} (x - x_k)_+^{2m-j-1}$$

or

$$(S_B f)(x) = \sum_{k=1}^{n} \sum_{j \in I_k} s_{kj}(x) f^{(j)}(x_k),$$

where s_{kj} are the fundamental interpolation spline functions. They can be obtained by conditions as (2.2.32).

We show this procedure in the following example.

Example 2.2.33. Let $f \in C^2[0,1]$ and $\Lambda_B(f) = \left\{f'(0), f\left(\frac{1}{2}\right), f'(1)\right\}$ be given. We are looking for the corresponding interpolation formula.

We consider the cubic spline function:

$$(S_3 f)(x) = s_{01}(x) f'(0) + s_{10}(x) f\left(\tfrac{1}{2}\right) + s_{21}(x) f'(1).$$

We determine the fundamental interpolation spline functions s_{01}, s_{10} and s_{21}. They have the same expressions as (2.2.33), their coefficients being the solutions of some system of the form (2.2.34). So, for s_{01} we have:

$$\begin{cases} s'_{01}(0) := a_1^{01} = 1 \\ s_{01}\left(\frac{1}{2}\right) := a_0^{01} + \frac{1}{2} a_1^{01} + \frac{1}{4} b_{01}^{01} = 0 \\ s'_{01}\left(1\right) := a_1^{01} + 2 b_{01}^{01} + \frac{3}{4} b_{10}^{01} = 0 \\ s''_{01}\left(2\right) := 2 b_{01}^{01} + 9 b_{10}^{01} + 2 b_{21}^{01} = 0 \\ s'''_{01}\left(2\right) := 6 b_{10}^{01} = 0. \end{cases}$$

It follows

$$s_{01}(x) = -\tfrac{3}{8} + x - \tfrac{1}{2} x^2 + \tfrac{1}{2} (x-1)_+^2 .$$

In the same way we obtain

$$s_{10}(x) = 1$$
$$s_{21}(x) = -\tfrac{1}{8} + \tfrac{1}{2}x^2 - \tfrac{1}{2}(x-1)_+^2 .$$

For the remainder term we have:

$$(R_3 f)(x) = \int_0^1 \varphi_2(x,t) f''(t) dt,$$

with

$$\varphi_2(x,t) = (x-t)_+ - \left(\tfrac{1}{2} - t\right)_+ - s_{21}(x).$$

2.2.6 Rational interpolation operators

2.2.6.1 An iterative method to construct a rational interpolation operator

Let $F = P_r/P_s$, where $P_k \in \mathbb{P}_k$. Because F is not changing if we replace P_r and P_s with CP_r and CP_s, respectively, where C is a nonzero constant, it follows that F depends of $(r+s+1)$ parameters, for their determination being necessary the same number of conditions. In this section we will use interpolation conditions to determine function F.

Denoting by $m = r + s$, we consider $x_i \in [a,b]$, $i = 0, 1, ..., m$, with $x_i \neq x_j$, if $i \neq j$.

Definition 2.2.34. *The problem of determining the function F in conditions*

$$F(x_i) = f(x_i), \quad i = 0, 1, .., m, \tag{2.2.35}$$

is named the rational interpolation problem.

We shall present an iterative method for construction the function F, based on its representation under continuous limited fraction form.

First, we consider the sequence v_k, $k = 0, 1, ...$, defined by recurrence relation:

$$v_k(x) = v_k(x_k) + (x - x_k)/v_{k+1}(x). \tag{2.2.36}$$

Denoting by $v_0 = f$ and applying successively the relation (2.2.36), we obtain:

$$f(x) = f(x_0) + \cfrac{x - x_0}{v_1(x_1) + \cfrac{x - x_1}{v_2(x_2) + \ddots + \cfrac{x - x_{k-1}}{v_k(x_k) + \cfrac{x - x_k}{v_{k+1}(x)}}}}. \tag{2.2.37}$$

The problem is to define the numbers $v_i(x_i)$, such that the function F_m given by

$$F_m(x) = f(x_0) + \cfrac{x - x_0}{v_1(x_1) + \cfrac{x - x_1}{v_2(x_2) + \ddots + \cfrac{x - x_{m-1}}{v_m(x_m)}}},$$

satisfies the interpolation conditions

$$F_m(x_k) = f(x_k), \quad k = 0, 1, .., m.$$

Definition 2.2.35. *Let us consider* $M = \{x_i \mid x_i \in \mathbb{R},\ i = 0, ..., m\}$ *and* $f : M \to \mathbb{R}$. *The value*

$$\{x_0, x_1, ..., x_{k-1}, x_k; f\} = \frac{x_k - x_{k-1}}{\{x_0, ..., x_{k-2}, x_k; f\} - \{x_0, ..., x_{k-1}; f\}},$$

with $\{x_0, x_1; f\} = [x_0, x_1; f]^{-1}$, *is called* the inverse divided difference of k order.

Remark 2.2.36. The inverse divided difference of $k > 1$ order differs from the inverse of divided difference. Also, unlike of divided difference, inverse divided difference depends on the order of nodes.

Theorem 2.2.37. *Let* $f : M \to \mathbb{R}$. *If* $v_k(x_k) = \{x_0, x_1, ..., x_k; f\}$, $k = 1, ..., m$, *then*

$$F_m(x_k) = f(x_k), \quad k = 0, 1, .., m.$$

Proof. We reduce it to a direct verification. Indeed, for each $k = 0, 1, ..., m$, we have

$$F_m(x_k) = f(x_0) + \cfrac{x_k - x_0}{\{x_0, x_1; f\} + \ddots + \cfrac{x_k - x_{k-2}}{\{x_0, ..., x_{k-1}, x_k; f\} + \cfrac{x_k - x_{k-1}}{\{x_0, ..., x_k; f\}}}}$$

Replacing the inverse divided difference of maximum order $\{x_0, ..., x_k; f\}$, by the expression obtained from definition,

$$\{x_0, x_1, ..., x_k; f\} = \frac{x_k - x_{k-1}}{\{x_0, ..., x_{k-2}, x_k; f\} - \{x_0, ..., x_{k-1}; f\}},$$

we obtain

$$F_m(x_k) = f(x_0) + \cfrac{\frac{x_k - x_0}{}}{\{x_0, x_1; f\} + \cfrac{}{\ddots + \frac{x_k - x_{k-2}}{\{x_0, ..., x_{k-2}, x_k; f\}}}}$$

Using this procedure, after $k-1$ substitutions, it follows that

$$F_m(x_k) = f(x_0) + \frac{x_k - x_0}{\{x_0, x_k; f\}}.$$

Since

$$\{x_0, x_k; f\} = \frac{x_k - x_0}{f(x_k) - f(x_0)},$$

it follows that

$$F_m(x_k) = f(x_k).$$

■

Remark 2.2.38. The function F_m is a limited continuous fraction.

We introduce an iterative procedure which puts function F_m under rational function form. For this, we write the relation

$$f(x) = f(x_0) + \cfrac{\frac{x - x_0}{}}{v_1(x_1) + \cfrac{}{\ddots + \frac{x - x_{k-1}}{v_k(x)}}}$$

under equivalent form

$$f(x) = \frac{v_k(x) G_k(x) + (x - x_{k-1}) P_k(x)}{v_k(x) H_k(x) + (x - x_{k-1}) Q_k(x)}. \tag{2.2.38}$$

In order to determine functions G_k, H_k, Q_k and P_k, we first observe that from relation (2.2.37) we also have

$$f(x) = \frac{v_{k+1}(x) G_{k+1}(x) + (x - x_k) P_{k+1}(x)}{v_{k+1}(x) H_{k+1}(x) + (x - x_k) Q_{k+1}(x)}. \tag{2.2.39}$$

On the other hand, by replacing in (2.2.38) the expression (2.2.36) of $v_k(x)$, we obtain:

$$f(x) = \frac{v_{k+1}(x)[v_k(v_k)G_k(x)+(x-x_{k-1})P_k(x)]+(x-x_k)G_k(x)}{v_{k+1}(x)[v_k(v_k)H_k(x)+(x-x_{k-1})Q_k(x)]+(x-x_k)H_k(x)}, \tag{2.2.40}$$

that is identical with (2.2.39).

From (2.2.39) and (2.2.40), it follows that

$$P_{k+1}(x) = G_k(x),$$
$$Q_{k+1}(x) = H_k(x),$$

and

$$G_{k+1}(x) = v_k(x_k)\,G_k(x) + (x-x_{k-1})\,P_k(x),$$
$$H_{k+1}(x) = v_k(x_k)\,H_k(x) + (x-x_{k-1})\,Q_k(x).$$

So, we have the recurrence relations

$$\begin{aligned} G_{k+1}(x) &= v_k(x_k)\,G_k(x) + (x-x_{k-1})\,G_{k-1}(x), \\ H_{k+1}(x) &= v_k(x_k)\,H_k(x) + (x-x_{k-1})\,H_{k-1}(x). \end{aligned} \tag{2.2.41}$$

Consequently,

$$f(x) = \frac{v_k(x)G_k(x)+(x-x_{k-1})G_{k-1}(x)}{v_k(x)H_k(x)+(x-x_{k-1})H_{k-1}(x)}. \tag{2.2.42}$$

The functions G_k, H_k, $k>1$, are determined from (2.2.41), for given G_0, G_1 and H_0, H_1.

In order to obtain these started functions, we consider the following expression,

$$f(x) = \frac{v_1(x)G_1(x)+(x-x_0)G_0(x)}{v_1(x)H_1(x)+(x-x_0)H_0(x)},$$

obtained from (2.2.42), for $k=1$. But, we also have the equivalent representation:

$$f(x) = f(x_0) + \frac{x-x_0}{v_1(x)} = \frac{v_1(x)f(x_0)+(x-x_0)}{v_1(x)}.$$

Identifying these two last expressions of f, one obtains

$$\begin{aligned} &G_0(x) = 1; \quad G_1(x) = f(x_0) \\ &H_0(x) = 0; \quad H_1(x) = 1. \end{aligned} \tag{2.2.43}$$

From (2.2.41) and (2.2.43) it follows that G_k and H_k are polynomials, so

$$G_k \in \mathbb{P}_{\left[\frac{k}{2}\right]}, \quad H_k \in \mathbb{P}_{\left[\frac{k-1}{2}\right]}, \quad k = 1, 2, \ldots.$$

We have proved the following result.

Theorem 2.2.39. *If $f : M \to \mathbb{R}$, $v_k(x_k) = \{x_0, ..., x_k; f\}$, for $k = 1, ..., m$, and*

$$F_m = \frac{G_{m+1}}{H_{m+1}},$$

with G_{m+1} and H_{m+1} given by (2.2.41), then

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

Hence, F_m is a rational function, $F_m = P_r/Q_s$, where $r = (m+1)/2$ and $s = r - 1$, if m is odd, and $r = s = m/2$, if m is even.

We can consider the following formula of the rational interpolation

$$f = \rho_m f + r_m f,$$

where ρ_m, defined by $\rho_m f = F_m$, is a rational interpolation operator and $r_m f$ is the remainder term.

Theorem 2.2.40. *The remainder term have the following expression:*

$$(r_m f)(x) = (-1)^m \frac{u(x)}{H_{m+1}(x)[v_{m+1}(x)H_{m+1}(x)+(x-x_m)H_m(x)]}, \tag{2.2.44}$$

where $u(x) = (x - x_0)...(x - x_m)$.

Proof. We have

$$(r_m f)(x) = \frac{v_m(x)G_m(x)+(x-x_{m-1})G_{m-1}(x)}{v_m(x)H_m(x)+(x-x_{m-1})H_{m-1}(x)} - \frac{G_{m+1}(x)}{H_{m+1}(x)}.$$

Bringing at the same denominator and reducing alike terms, we obtain

$$(r_m f)(x) = \frac{(x-x_{m-1})[v_m(x)-v_m(x_m)][G_m(x)H_{m-1}(x)-H_m(x)G_{m-1}(x)]}{[v_m(x)H_m(x)+(x-x_{m-1})H_{m-1}(x)]H_{m+1}(x)}.$$

Using (2.2.36), it follows

$$v_m(x) - v_m(x_m) = (x - x_m)/v_{m+1}(x)$$

and

$$v_m(x)H_m(x) + (x - x_{m-1})H_{m-1}(x) = H_{m+1}(x) + \frac{x-x_m}{v_{m+1}(x)}H_m(x).$$

Then,

$$(r_m f)(x) = \frac{(x-x_{m-1})(x-x_m)[G_m(x)H_{m-1}(x)-H_m(x)G_{m-1}(x)]}{[v_{m+1}(x)H_{m+1}(x)+(x-x_m)H_m(x)]H_{m+1}(x)}.$$

Taking into account that

$$G_2(x)H_1(x) - H_2(x)G_1(x) = x - x_0$$

and

$$\begin{aligned} &G_m(x)H_{m-1}(x) - H_m(x)G_{m-1}(x) = \\ &= -(x - x_{m-2})[G_{m-1}(x)H_{m-2}(x) - H_{m-1}(x)G_{m-2}(x)], \end{aligned}$$

it follows that

$$G_m(x)H_{m-1}(x) - H_m(x)G_{m-1}(x) = (-1)^{m-2}(x - x_0)...(x - x_{m-2}),$$

so the relation (2.2.44) is proved. ■

Remark 2.2.41. For the absolute approximation error we have

$$|(r_m f)(x)| = \frac{|u(x)|}{|H_{m+1}(x)||v_{m+1}(x)H_{m+1}(x)+(x-x_m)H_m(x)|},$$

for given $x \in [a, b]$.

To evaluate the value of the approximation $\rho_m f$ at a point $\alpha \in \mathbb{R}$, taking into account that

$$(\rho_m f)(\alpha) = \frac{G_{m+1}(\alpha)}{H_{m+1}(\alpha)},$$

we can use the recurrence relations (2.2.41), with starting values (2.2.43).

Using the following scheme we can evaluate the inverse divided differences which intercede in the expressions of the polynomials G_k and H_k:

$$\begin{array}{|c|c|c|c|c|c|c|} x_0 & v_{00} & & & & & \\ x_1 & v_{10} & v_{11} & & & & \\ x_2 & v_{20} & v_{21} & v_{22} & & & \\ ... & & & & & & \\ x_i & v_{i0} & v_{i1} & v_{i2} & ... & v_{ii} & \\ ... & & & & & & \\ x_m & v_{m0} & v_{m1} & v_{m2} & ... & v_{mi} & ... \quad v_{mm}, \end{array}$$

where

$$\begin{aligned} &v_{i0} = f(x_i), \quad i = 0, ..., m \\ &v_{ik} = \frac{x_i - x_{k-1}}{v_{i,k-1} - v_{k-1,k-1}}, \quad k = 1, ..., i; \quad i = 1, ..., m. \end{aligned} \tag{2.2.45}$$

We observe that the inverse divided differences v_{ii}, from the expressions of the polynomials G_k and H_k, are situated on the hypotenuses of the triangular scheme and their evaluation, according to (2.2.45), can be realized after lines, each line giving the inverse divided difference of the following superior order.

Thus, using numbers v_{ii}, $i = 0, 1, ...$, there can be generated successively approximations:

$$(\rho_1 f)(\alpha), (\rho_2 f)(\alpha), ..., (\rho_i f)(\alpha), ... \tag{2.2.46}$$

until the distance between two consecutive elements of the sequence (2.2.46) is less or equal to a given $\varepsilon > 0$, i.e.,

$$|(\rho_i f)(\alpha) - (\rho_{i-1} f)(\alpha)| \leq \varepsilon.$$

Therefore, we can say that $(\rho_i f)(\alpha)$ approximates $f(\alpha)$ with a given precision ε.

2.2.6.2 Univariate Shepard interpolation

D. Shepard has introduced in 1968 an interpolation procedure which is very efficient in interpolation of large scattered data sets.

Let f be a real-valued function defined on $I = [a, b] \subset \mathbb{R}$, $x_i \in I$, $i = 1, \ldots, N$, $X \subset I$, X denoting the set of nodes, and consider

$$\Lambda := \Lambda_L = \{\lambda_i \,|\lambda_i(f) = f(x_i),\ i = 1, \ldots, N\} \tag{2.2.47}$$

a Lagrange type set of functionals.

The univariate Shepard operator S_0 is defined by:

$$(S_0 f)(x) = \sum_{i=1}^{N} A_i(x) f(x_i), \tag{2.2.48}$$

where

$$A_i(x) = \frac{\prod\limits_{j=1,\, j\neq i}^{N} |x - x_j|^{\mu}}{\sum\limits_{k=1}^{N} \prod\limits_{j=1,\, j\neq k}^{N} |x - x_j|^{\mu}}, \tag{2.2.49}$$

and $\mu \in \mathbb{R}_+$. The basis functions A_i may be written in barycentric form

$$A_i(x) = \frac{|x - x_i|^{-\mu}}{\sum\limits_{k=1}^{N} |x - x_k|^{-\mu}},\ i = 1, \ldots, N. \tag{2.2.50}$$

and

$$\sum_{i=1}^{N} A_i(x) = 1. \tag{2.2.51}$$

They have the cardinality property

$$A_i(x_\nu) = \delta_{i\nu}, \quad i, \nu = 1, ..., N. \tag{2.2.52}$$

The main properties of the operator S_0 are:

- the interpolation property regarding the set of functionals given by (2.2.47), i.e.,
$$(S_0 f)(x_i) = f(x_i), \quad i = 1, \ldots, N.$$
- the degree of exactness is
$$\operatorname{dex}(S_0) = 0.$$

Shepard's method is a particular example of a convex combination, i.e., the functions A_i, $i = 1, \ldots, N$ are non-negative and their sum is 1. From this, it follows that the scheme has only constant precision.

The Shepard interpolation has two major drawbacks:

• high computational cost;

• low degree of exactness.

These drawbacks can be overcome in two ways:

★ *Modifying basis functions.* Shepard method has the property that there are flat spots at the nodes, the accuracy tends to decrease in the areas where the interpolation nodes are sparse and the evaluation of $(S_0 f)(x)$ requires a considerable amount of work. These disadvantages are avoided by using a local version of the Shepard formula, given by R. Franke and G. Nielson, that consists in replacing the basis functions A_i, from (2.2.50), by

$$w_i(x) = \left(1 - \frac{|x - x_i|}{R}\right)_+^{\nu}, \tag{2.2.53}$$

where R is a radius of influence about the node x_i and it is varying with i.

★ *Increasing the degree of exactness* by combining the Shepard operator with another interpolation operator. This is one of the most efficient ways of generalizing the Shepard interpolation. In this way its degree of exactness is

increased and there are used different sets of functionals. We notice that in definition of the Shepard operator S there are used Lagrange type functionals.

Let

$$\Lambda := \{\lambda_i : i = 1, ..., N\}$$

be a set of functionals and P be the corresponding interpolation operator. We consider that $\Lambda_i \subset \Lambda$ are the subsets associated to the functionals λ_i, $i = 1, ..., N$. We have

$$\bigcup_{i=1}^{N} \Lambda_i = \Lambda \quad \text{and } \Lambda_i \cap \Lambda_j \neq \emptyset,$$

excepting the case $\Lambda_i = \{\lambda_i\}$, $i = 1, ..., N$, when $\Lambda_i \cap \Lambda_j = \emptyset$, for $i \neq j$. We associate the interpolation operator P_i to each subset Λ_i, $i = 1, ..., N$.

The operator S_P defined by

$$(S_P f)(x) = \sum_{i=1}^{N} A_i(x) (P_i f)(x) \tag{2.2.54}$$

is *the combined operator* of S_0 and P.

Remark 2.2.42. If P_i, $i = 1, ..., N$, are linear operators then S_P is a linear operator.

Remark 2.2.43. Let P_i, $i = 1, ..., N$, be some arbitrary linear operators. If

$$\operatorname{dex}(P_i) = r_i, \qquad i = 1, ..., N,$$

then

$$\operatorname{dex}(S_P) = \min\{r_1, ..., r_N\}.$$

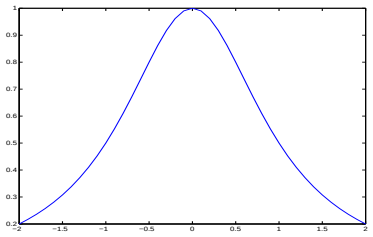
The function $S_0 f$ and its behavior in the neighborhood of the nodes depend on the size of μ.

If $0 < \mu \leq 1$ then the function $S_0 f$ has peaks at the nodes. For $\mu > 1$ the first derivatives vanish at the nodes, i.e., $S_0 f$ has flat spots and if μ is large enough $S_0 f$ becomes a step function. This phenomenon is shown in Figure 2.1.

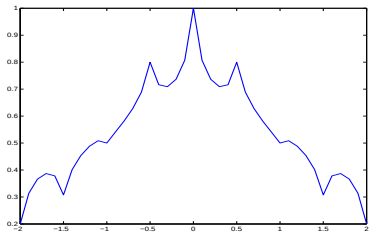
Example 2.2.44. Let $f : [-2, 2] \to \mathbb{R}$,

$$f(x) = \frac{1}{1 + x^2}$$

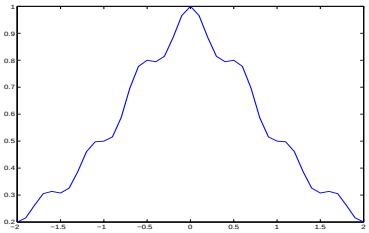
and consider the nodes $x_i = -2 + 0.5i$, $i = 0, ..., 8$. We plot $S_0 f$, for $\mu = 1, 2, 20$.



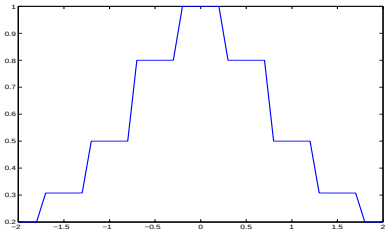
(a) Function f.



(b) Interpolant $S_0 f$, $\mu = 1$.



(c) Interpolant $S_0 f$, $\mu = 2$.



(d) Interpolant $S_0 f$, $\mu = 20$.

Figure 2.1: Univariate Shepard interpolants.

The most frequent choice is $\mu = 2$, in which case the basis functions A_i are rational and infinitely differentiable.

In the sequel we study the remainder of the interpolation formula generated by Shepard operator. The interpolation function $S_0 f$ can be viewed as a projection of f into the finite-dimensional linear space spanned by the functions $A_i,\ i = 1, \ldots, N$. The Shepard interpolation formula is:

$$f = S_0 f + R_0 f,$$

where $S_0 f$ was defined in (2.2.48) and $R_0 f$ is the remainder term.

Theorem 2.2.45. *If f is absolutely continuous on $[a, b]$, then*

$$(R_0 f)(x) = \int_a^b \varphi(x, s) f'(s) ds, \tag{2.2.55}$$

where

$$\varphi(x, s) = (x - s)_+^0 - \sum_{i=1}^{N} A_i(x)(x_i - s)_+^0.$$

Also,

$$|(R_0 f)(x)| \leq K(x) \left\| f' \right\|_\infty,$$

where

$$K(x) = x - \sum_{i=1}^{N} x_i A_i(x) + 2\sum_{i=1}^{N} A_i(x)(x_i - x)_+.$$

Proof. Using the Peano's theorem and taking into account that $R_0 e_0 = e_0$, where $e_k(x) = x^k$, formula (2.2.55) follows.

Next, let us suppose that $x \in (x_{k-1}; x_k)$ and

$$\varphi^j(x;\cdot) = \varphi(x;\cdot)\,\big|_{[x_{j-1},x_j]}\ ,\ \text{for } j = 1,\ldots,N.$$

We have

$$\varphi^j(x;s) = (x-s)_+^0 - \sum_{i=j}^{N} A_i(x).$$

Using $\sum_{i=1}^{N} A_i(x) = 1$ and $A_i(x) \geq 0$, $i = 1,\ldots,N$, one obtains:

$$\varphi^j(x;s) = \begin{cases} \sum\limits_{i=1}^{j-1} A_i(x), & \text{for } \ s \leq x, \\ -\sum\limits_{i=j}^{N} A_i(x), & \text{for } \ s > x. \end{cases}$$

Hence,

$$\varphi^j(x;s) \geq 0,\ \text{for } s \leq x,$$
$$\varphi^j(x;s) \leq 0,\ \text{for } s > x.$$

Since $s, x \in (x_{k-1}, x_k)$, it follows that

$$\varphi^j(x;s) \geq 0,\ \text{for } j = 1,\ldots,k-1,$$
$$\varphi^j(x;s) \leq 0,\ \text{for } j = k+1,\ldots,N,$$
$$\varphi^k(x;s) \geq 0,\ \text{for } s \leq x,$$
$$\varphi^k(x;s) \leq 0,\ \text{for } s > x.$$

Finally, one obtains that

$$\varphi(x;s) \geq 0,\ \text{for } s \leq x$$

and

$$\varphi(x;s) \leq 0,\ \text{for } s > x.$$

From (2.2.55) we have:

$$|(R_0 f)(x)| \leq \|f'\|_\infty \int_a^b |\varphi(x;s)|\, ds,$$

with

$$\int_a^b |\varphi(x;s)|\, ds = \int_a^x \varphi(x;s) ds - \int_x^b \varphi(x;s) ds = K(x)$$

and the theorem is proved. ■

Next, we assume that at each point x_i, $i = 1, \ldots, N$, there also exists the derivative $f'(x_i)$ and we consider the operator

$$(S_1 f)(x) = \sum_{i=1}^{N} A_i(x)[f(x_i) + (x - x_i) f'(x_i)].$$

It is easy to prove that if $\mu > 1$, the operator S_1 interpolates both the function f and its first derivative at the points x_i, $i = 1, \ldots, N$.

Lemma 2.2.46. *The operator S_1 is linear and*

$$S_1 e_k = e_k, \text{ for } k = 0, 1.$$

Proof. The proof follows after a straightforward computation. ■

This operator generates the following interpolation formula:

$$f = S_1 f + R_1 f.$$

Regarding its remainder term we have the following result.

Theorem 2.2.47. *If $f \in H^2[a, b]$ then*

$$(R_1 f)(x) = \int_a^b \varphi_1(x, s) f''(s) ds, \tag{2.2.56}$$

where

$$\varphi_1(x, s) = (x - s)_+ - \sum_{i=1}^{N} A_i(x)[(x_i - s)_+ + (x - x_i)(x_i - s)_+^0].$$

Moreover, if $f''$is continuous on the interval $[a, b]$, then

$$(R_1 f)(x) = \left[-\frac{x^2}{2} + \frac{1}{2} \sum_{i=1}^{N} A_i(x) x_i^2 \right] f''(\xi), \text{ with } \xi \in (a, b). \tag{2.2.57}$$

Proof. Lemma 2.2.46 and the Peano's theorem imply (2.2.56). The function $\varphi_1^k(x;\cdot) = \varphi_1(x;\cdot)\big|_{[x_{k-1},x_k]}$ has the form

$$\varphi_1^k(x,s) = (x-s)_+ - (x-s)\sum_{i=k}^{N} A_i(x),$$

so

$$\varphi_1^k(x;s) = \begin{cases} (x-s)\sum_{i=1}^{k-1} A_i(x), & \text{for } s \le x \\ -(x-s)\sum_{i=k}^{N} A_i(x), & \text{for } s > x, \end{cases}$$

i.e.,

$$\varphi_1^k(x;s) \ge 0, \quad \text{for all } k.$$

It follows that $\varphi_1(x;s) \ge 0$, for $x, s \in [a,b]$, and by the mean theorem, one obtains the representation (2.2.57). ■

2.2.6.2.1 The univariate Shepard-Lagrange operator Let f be a real-valued function defined on $X \subset \mathbb{R}$, $x_i \in X$, $i = 1,\dots,N$ some distinct nodes. Consider the set of Lagrange functionals

$$\Lambda := \Lambda_L(f) = \{\lambda_i \,|\lambda_i(f) = f(x_i),\ i = 1,\dots,N\}$$

and

$$\Lambda_i(f) = \{f(x_{i+\nu}) \mid \nu = 0,1,\dots,m,\ i = 1,\dots,N\},$$

with $x_{N+\nu} = x_{N-\nu}$, $\nu = 1,\dots,m$. The operator

$$(L_m^i f)(x) = \sum_{\nu=0}^{m} \frac{u(x)}{(x-x_{i+\nu})u'(x_{i+\nu})} f(x_{i+\nu})$$

is the Lagrange operator suitable to $\Lambda_i(f)$.

Definition 2.2.48. *The operator S_{L_m} given by*

$$(S_{L_m} f)(x) = \sum_{i=1}^{N} A_i(x)(L_m^i f)(x)$$

is called the univariate Shepard-Lagrange operator.

Theorem 2.2.49. *For $\mu > m$, we have the following interpolation properties:*

$$(S_{L_m} f)(x_i) = f(x_i), \quad i = 1, \ldots, N, \tag{2.2.58}$$

and the degree of exactness is:

$$\operatorname{dex}(S_{L_m}) = m. \tag{2.2.59}$$

Proof. The interpolation properties (2.4.8) follows taking into account relations (2.2.52) and the interpolatory properties of $L_m^i f$, $i = 1, \ldots, N$.

Relation (2.2.59) follows directly by Remark 2.2.43. ■

The Shepard-Lagrange univariate interpolation formula is

$$f = S_{L_m} f - R_{L_m} f,$$

where $R_{L_m} f$ denotes the remainder term. We have the following error estimation.

Theorem 2.2.50. *If $f \in C^{m+1}(I)$, then*

$$\|R_{L_m} f\|_\infty \leq CM \left\|f^{(m+1)}\right\|_\infty \varepsilon_\mu^m(r),$$

with

$$\varepsilon_\mu^m(r) = \begin{cases} |\log r|^{-1}, & \mu = 1, \\ r^{\mu-1}, & 1 < \mu < m+2, \\ r^{\mu-1} |\log r|, & \mu = m+2, \\ r^{m+1}, & \mu > m+2, \end{cases}$$

where C is a positive constant independent of x and X.

Proof. If $f \in C^{m+1}(I)$, then the following bound for the interpolation error is known:

$$|(R_m f)(x; x_i)| = |(L_m f)(x; x_i) - f(x)| \leq \tfrac{|x-x_i|\ldots|x-x_{i+m}|}{(m+1)!} \left\|f^{(m+1)}\right\|_\infty.$$

We have

$$|(S_{L_m} f)(x) - f(x)| \leq \sum_{i=1}^{N} |(R_m f)(x; x_i)| \, A_i(x) \leq C_m \left\|f^{(m+1)}\right\|_\infty s_\mu^m(x),$$

where

$$s_\mu^m(x) = \frac{\sum\limits_{i=1}^{N} |x-x_i|\ldots|x-x_{i+m}||x-x_i|^{-\mu}}{\sum\limits_{k=1}^{N} |x-x_i|^{-\mu}}.$$

We show that

$$s_\mu^m(x) \leq CM\varepsilon_\mu^m(r).$$

We introduce the following notations

$$\begin{aligned} &B_\rho(x) = [x-\rho, x+\rho], \\ &r = \inf\{\rho > 0 : \forall x \in I \;\; \exists\, u \in X \;\; u \in B_\rho(x)\} \end{aligned}$$

and

$$K = \sup_y \operatorname{card}(B_r(y) \cap X).$$

Let

$$n = \left[\tfrac{b-a}{2r}\right] + 1,$$

$Q_r(u)$ be the interval $(u-r, u+r]$ and $T_j = Q_r(x+2rj) \cup Q_r(x-2rj)$. The set

$$\bigcup_{j=-n}^{n} Q_r(x+2rj)$$

is a covering of I with half open intervals. If $x_i \in I \cap T_j$ we have

$$(2j-1)r \leq |x-x_i| \leq (2j+1)r, \quad j = 1, \ldots, N \tag{2.2.60}$$

and

$$1 \leq \operatorname{card}(X \cap T_j) \leq M.$$

Also, $N = \operatorname{card}(x) = O(r^{-1})$ and $[2(j-l)-1]r \leq |x-x_{i+l}| \leq [2(j+l)+1]r$. If $x_i \in T_j$ we have

$$|x-x_i| \ldots |x-x_{i+m}| \leq r^{m+1} \prod_{l=0}^{m} [2(j+l)+1] \tag{2.2.61}$$

Let x_d be the closest point to x. Since

$$\sum_{i+1}^{N} |x-x_i|^{-\mu} \geq |x-x_d|^{-\mu},$$

applying (2.2.60) and (2.2.61), we get:

$$\begin{aligned} s_\mu^m(x) \le & |x-x_d|^{-\mu} \left(\sum_{x_i \in T_0} |x-x_d|^{-\mu} |x-x_i| \dots |x-x_{i+m}| + \right. \\ & \left. + \sum_{j=1}^{n} \sum_{x_i \in T_j} |x-x_d|^{-\mu} |x-x_i| \dots |x-x_{i+m}| \right) \\ \le & r^\mu m M r^{m+1-\mu} \left\{ \prod_{l=0}^{m} (2l+1) + \sum_{j=1}^{n} \left(\prod_{l=0}^{m} [2(j+l)+1](2j+1)^{-\mu} \right) \right\} \\ \le & m M r^{m+1} \prod_{l=0}^{m} (2l+1) \Big(1 + C \sum_{j=1}^{n} j^{m+1-\mu} \Big). \end{aligned}$$

We have two cases. $\mu > 1$:

- if $1 < \mu < m+2$ then

$$r^{m+1} \left(1 + C \sum_{j=1}^{n} j^{m+1-\mu} \right) = O(r^{\mu-1}).$$

- if $\mu = m+2$ then

$$\sum_{j=1}^{n} j^{m+1-\mu} = \log N = |\log r| \,.$$

- if $\mu > m+2$ then $\sum_{j=1}^{n} j^{m+1-\mu}$ is bounded.

$\mu = 1$: We have

$$s_\mu^m(x) = s_1^m(x) = \frac{\sum\limits_{i=1}^{N} |x-x_i|^{-1} |x-x_i| \dots |x-x_{i+m}|}{\sum\limits_{i=1}^{N} |x-x_i|^{-1}}.$$

Applying the inequality

$$\frac{\sum a_i}{\sum b_i} \le \sum \frac{a_i}{b_i},$$

we get

$$\begin{aligned}
&s_1^m(x) = \sum_{x_i \in T_0} |x - x_i| \dots |x - x_{i+m}| \\
&\cdot C_1 \tfrac{r}{|\log r|} \sum_{j=1}^{n} \sum_{x_k \in T_j} |x - x_i|^{-1} |x - x_i| \dots |x - x_{i+m}| \\
&\leq mMr^{m+1} \prod_{l=0}^{m} (2l+1) \left(1 + \tfrac{C_2}{|\log r|} \sum_{j=1}^{n} j^q \right) \\
&\leq C_1 M r^{m+1} \left(1 + \tfrac{C_2}{|\log r|} O(r^{-m-1}) \right) = O(|\log r|^{-1}).
\end{aligned}$$

■

Particular cases. 1) Consider $m = 1$. The combined operator S_{L_1} is given by:

$$(S_{L_1} f)(x) = \sum_{i=1}^{N} A_i(x)(L_1^i f)(x),$$

where

$$(L_1^i f)(x) = \frac{x - x_{i+1}}{x_i - x_{i+1}} f(x_i) + \frac{x - x_i}{x_{i+1} - x_i} f(x_{i+1}), \text{ for } i = 1, ..., N,$$

with $x_{N+1} = x_1$.

2) Consider $m = 2$. The combined operator S_{L_2} is given by:

$$(S_{L_2} f)(x) = \sum_{i=1}^{N} A_i(x)(L_2^i f)(x),$$

where

$$(L_2^i f)(x) = l_i(x) f(x_i) + l_{i+1}(x) f(x_{i+1}) + l_{i+2}(x) f(x_{i+2}),$$

with

$$\begin{aligned}
l_i(x) &= \frac{(x - x_{i+1})(x - x_{i+2})}{(x_i - x_{i+1})(x_i - x_{i+2})}, \\
l_{i+1}(x) &= \frac{(x - x_i)(x - x_{i+2})}{(x_{i+1} - x_i)(x_{i+1} - x_{i+2})}, \\
l_{i+2}(x) &= \frac{(x - x_i)(x - x_{i+1})}{(x_{i+1} - x_i)(x_{i+2} - x_{i+1})},
\end{aligned}$$

for all $i = 1, ..., N$ and considering $x_{N+1} = x_1$ and $x_{N+2} = x_2$.

2.2.6.2.2 The univariate Shepard-Hermite operator Let f be a real-valued function defined on $X \subset \mathbb{R}$, $x_i \in X$, $i = 1, \ldots, N$ some distinct nodes. Consider the set of Hermite functionals regarding f:

Let

$$\Lambda := \Lambda_H(f) = \left\{ \lambda_{ij} \,\middle|\, \lambda_{ij}(f) = f^{(j)}(x_i)\ , \quad i = 1, \ldots, N,\ \ j = 1, \ldots, r_i,\ \ r_i \in \mathbb{N}^* \right\}$$

and the subset

$$\Lambda_{i,j}(f) = \left\{ f^{(j)}(x_{i+\nu}) \ \middle|\ \nu = 0, 1, \ldots, m,\ j = 1, \ldots, r_\nu \right\},$$

for $i = 1, ..., N$ with $x_{N+\nu} = x_{N-\nu}$, for $\nu = 1, ..., m$, $m \in \mathbb{N}^*$.

Definition 2.2.51. *The operator S_{H_q} given by*

$$(S_{H_q} f)(x) = \sum_{i=1}^{N} A_i(x)(H_q^i f)(x)$$

is called the univariate Shepard-Hermite operator*, where $H_q^i f$ is the q-th Hermite interpolation polynomial, with $q = m + r_0 + ... + r_m$.*

Theorem 2.2.52. *For $\mu > q$, the following interpolation conditions are fulfilled:*

$$(S_{H_q} f)^{(j)}(x_i) = f^{(j)}(x_i), \quad i = 1, \ldots, N;\ j = 0, \ldots, r_i, \tag{2.2.62}$$

and the degree of exactness is:

$$\operatorname{dex}(S_{H_q}) = q. \tag{2.2.63}$$

Proof. The interpolation properties (2.2.62) follow taking into account relations (2.2.52) and the interpolatory properties of $H_q^i f$, $i = 1, \ldots, N$.

Relation (2.2.63) follows directly by Remark 2.2.43. ■

The Shepard-Hermite interpolation formula is

$$f = S_{H_q} f + R_{H_q} f,$$

where $R_{H_q} f$ denotes the remainder term.

Theorem 2.2.53. *If $f \in C^{q+1}(I)$, then*

$$\left\|R_{H_q} f\right\|_I \leq CM \left\|f^{(q+1)}\right\|_I \varepsilon_\mu^q(r),$$

where

$$\varepsilon_\mu^q(r) = \begin{cases} |\log r|^{-1}, & \mu = 1, \\ r^{\mu-1}, & 1 < \mu < q+2, \\ r^{\mu-1} |\log r|, & \mu = q+2, \\ r^{q+1}, & \mu > q+2, \end{cases}$$

and C is a positive constant independent of x and X.

Proof. The proof is analogous to that of Theorem 2.2.50, with remark that $H_q^i f$ has the nodes $x_{i+\nu}$ of multiplicities r_ν, respectively. ■

2.2.6.2.3 The univariate Shepard-Taylor operator Let f be a real-valued function defined on $X \subset \mathbb{R}$ and $x_i \in X$, $i = 1, \ldots, N$ some distinct nodes. Consider the set of Taylor functionals regarding f :

$$\Lambda := \Lambda_T(f) = \{\lambda_{kj} \mid \lambda_{kj}(f) = f^{(j)}(x_k), \quad j = 0, 1, \ldots, m;\ k = 1, \ldots, N\},$$

$m \in \mathbb{N}^*$ and $\Lambda_i(f) = \{\lambda_{ij}(f) \mid j = 0, 1, \ldots, m\}$, where $\Lambda_i \subset \Lambda_T$ is a subset of Λ_T associated to the functional λ_i such that $\lambda_i \in \Lambda_i$, for all $i = 1, \ldots, N$. We shall denote by

$$(T_m^i f)(x) = \sum_{j=0}^{m} \frac{(x - x_i)^j}{j!} f^{(j)}(x_i)$$

the Taylor interpolation operator corresponding to the subset of functionals Λ_i.

Definition 2.2.54. *The operator S_{T_m} given by*

$$(S_{T_m} f)(x) = \sum_{i=1}^{N} A_i(x)(T_m^i f)(x)$$

is called the univariate Shepard-Taylor operator.

Theorem 2.2.55. *For $\mu > m$,*

$$(S_{T_m} f)^{(j)}(x_i) = f^{(j)}(x_i), \quad j = 1, \ldots, m,$$

and

$$\operatorname{dex}(S_{T_m}) = m.$$

Proof. The conclusion follows taking into account relations (2.2.52) and the interpolatory properties of $T_m^i f$, $i = 1, \ldots, N$ and by Remark 2.2.43. ■

A generalization of the operator S_{T_m} is obtained when

$$\Lambda_T = \{\lambda_{kj} \mid \lambda_{kj}(f) = f^{(j)}(x_k), \quad j = 0, 1, \ldots, m_k;\ k = 1, \ldots, N\}$$

and

$$\Lambda_i = \{\lambda_{ij} \mid j = 0, 1, \ldots, m_i\}.$$

We have

$$(S_{T_{m_1,\ldots,m_N}} f)(x) = \sum_{i=1}^{N} A_i(x)(T_{m_i}^i f)(x),$$

where the Taylor operator $T_{m_i}^i$ has the degree m_i, $i = 1, \ldots, N$.

Remark 2.2.56. For $\mu > M := \max\{m_1, \ldots, m_N\}$,

$$(S_{T_{m_1,\ldots,m_N}} f)^{(j)}(x) = f^{(j)}(x_i), \quad j = 0, 1, \ldots, m_i, i = 1, \ldots, N,$$

and

$$\operatorname{dex}(S_{T_{m_1,\ldots,m_N}}) = \min\{m_1, \ldots, m_N\}.$$

Remark 2.2.57. Regarding the approximation error we note that the Shepard-Taylor, the Shepard-Lagrange and the Shepard-Hermite operators have the same rate of convergence.

2.2.6.2.4 The univariate Shepard-Birkhoff operator Let f be a real-valued function defined on $X \subset \mathbb{R}$, $x_i \in X$, $i = 1, \ldots, N$ some distinct nodes. Consider the set of Birkhoff functionals regarding f :

$$\Lambda := \Lambda_B(f) = \{\lambda_{kj} \mid \lambda_{kj}(f) = f^{(j)}(x_k), \quad j = I_k;\ k = 1, \ldots, N\},$$

and

$$\Lambda_{i,j}(f) = \left\{ f^{(j)}(x_{i+\nu}) \,|\nu = 0, 1, \ldots, m,\ j \in I_{i+\nu} \right\},$$

with $x_{N+\nu} = x_{N-\nu}$, for $\nu = 1, ..., m$, $m \in \mathbb{N}^*$.

Definition 2.2.58. *The operator S_{B_m} given by*

$$(S_{B_m} f)(x) = \sum_{i=1}^{N} A_i(x)(B_m^i f)(x)$$

is called the univariate Shepard-Birkhoff operator.

Theorem 2.2.59. *For $\mu > m$, the following interpolation conditions are fulfilled:*

$$(S_{B_m} f)^{(j)}(x_k) = f^{(j)}(x_k), \quad j \in I_k;\ k = 1, \ldots, N, \tag{2.2.64}$$

and

$$\operatorname{dex}(S_{B_m}) = m. \tag{2.2.65}$$

Proof. The interpolation properties (2.2.64) follow taking into account relations (2.2.52) and the interpolatory properties of $H_q^i f$, $i = 1, \ldots, N$.

Relation (2.2.65) follows directly by Remark 2.2.43. ■

Remark 2.2.60. In the same way there are defined the combined Shepard-Abel-Goncharov and Shepard-Lidstone operators, which are particular cases of the combined Shepard-Birkhoff operator (see, e.g., [22] and [23]).

2.2.7 Least squares approximation

In this section, as an extension of the interpolation problem, it is given a short review of the least squares approximation.

Recall that when the values of a function f are known for a given set of points x_i, $i = 0, ..., m$, an interpolation method can be used to determine an approximation F of the function f, such that

$$F(x_i) = f(x_i), \ i = 0, ..., m.$$

If only approximations of the values $f(x_i)$ are available, or if the number of interpolation conditions is too large, instead of requiring that the approximating function reproduces the given function values exactly, we ask only that it fits the data "as closely as posible". The most popular application involves the least squares principle, where the approximation F is determined such that

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

in the discrete case, and

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

in the continuous case, where $w \geq 0$ on $[a, b]$ is a weight function.

Remark 2.2.61. Notice that the interpolation is a particular case of the least squares approximation, namely for

$$f(x_i) - F(x_i) = 0, \quad i = 0, ..., m.$$

In the sequel we briefly describe each of the two cases, starting with the continuous one.

Consider the space $L_w^2[a,b]$, the corresponding inner product,

$$\langle f, g \rangle_{w,2} = \int_a^b w(x) f(x) g(x), \quad f, g \in L_w^2[a,b],$$

the norm endowed by it,

$$\|f\|_{w,2} = \left(\int_a^b w(x) f^2(x) dx \right)^{1/2},$$

and the distance between two functions f and g, given by

$$d_w(f,g) = \|f - g\|_{w,2}.$$

For $\mathcal{A} \subset L_w^2[a,b]$ and $f \in L_w^2[a,b]$, the question is whether there exists an element $g^* \in \mathcal{A}$ such that

$$d_w(f, g^*) = \min_{g \in A} d_w(f,g).$$

It will be shown that such an element exists and it is unique.

If $\mathcal{A}$ is a linear finite dimensional space and g^* exists, then

$$\langle f - g^*, g \rangle_{w,2} = 0, \quad g \in \mathcal{A}. \tag{2.2.66}$$

Let n be the dimension of $\mathcal{A}$ and $g_1, ..., g_n \in \mathcal{A}$ be a basis in $\mathcal{A}$.

The orthogonality condition (2.2.66), with

$$g = \sum_{i=1}^n a_i g_i$$

and

$$g^* = \sum_{i=1}^n a_i^* g_i,$$

is equivalent to the system

$$\langle f - g^*, g_k \rangle_{w,2} = 0, \quad k = 1, ..., n,$$

with n equations and n unknowns a_i, $i = 1, ..., n$. The latter system can be explicitly written as

$$\sum_{i=1}^{n} a_i \langle g_i, g_k \rangle_{w,2} = \langle f, g_k \rangle_{w,2}, \quad k = 1, ..., n, \tag{2.2.67}$$

and its determinant is the Gram determinant $G(g_1, ..., g_n)$. Since the functions $g_1, ..., g_n$ are linearly independent, we have

$$G(g_1, ..., g_n) \neq 0,$$

hence

$$g^* = \sum_{i=1}^{n} a_i^* g_i$$

is uniquely determined.

Next we discuss three particular cases.

Case A. We consider $\mathcal{A} = \mathbb{P}_n$, and $g_k = e_k$, $k = 0, ..., n$.

In this case, g^* is the nth degree polynomial whose coefficients satisfy the condition

$$\int_a^b w(x) \left[f(x) - \sum_{k=0}^{n} a_k^* x^k \right]^2 dx = \min_{a_0, ..., a_n} \int_a^b w(x) \left[f(x) - \sum_{k=0}^{n} a_k x^k \right]^2 dx.$$

According to (2.2.67), the solution of the approximation problem can be obtain solving the following system of equations

$$\sum_{k=0}^{n} a_k \left\langle x^k, x^j \right\rangle_{w,2} = \left\langle f(x), x^j \right\rangle_{w,2}, \quad j = 0, ..., n,$$

where

$$\left\langle x^k, x^j \right\rangle_{w,2} = \int_a^b w(x) x^{k+j} dx,$$
$$\left\langle f(x), x^j \right\rangle_{w,2} = \int_a^b w(x) x^j f(x) dx.$$

Case B. We consider $\mathcal{A} = \mathbb{P}_n$, and $g_k = p_k$, $k = 0, ..., n$, where $\{p_0, ..., p_n\}$ is a set of orthogonal polynomials, in the interval $[a, b]$, relative to the weight function w.

Since $\langle p_i, p_j \rangle_{w,2} = 0$, for $i \neq j$, and

$$\langle p_k, p_k \rangle_{w,2} = \|p_k\|^2_{w,2},$$

from (2.2.67) it follows

$$a_k^* = \frac{1}{\|p_k\|^2_{w,2}} \int_a^b w(x) f(x) p_k(x) dx, \quad k = 0, ..., n, \tag{2.2.68}$$

and

$$g^*(x) = \sum_{k=0}^{n} a_k^* p_k(x).$$

Remark 2.2.62. The numbers a_k^* in (2.2.68) are the coefficients of the expansion of f in terms of the orthogonal polynomials p_k, $k = 0, 1, \ldots$. If these polynomials are orthonormal (i.e., $\|p_k\|_2 = 1$), we have

$$a_k^* = \int_a^b w(x) f(x) p_k(x) dx, \quad k = 0, 1, \ldots.$$

Case C. We consider that $f : \mathbb{R} \to \mathbb{R}$ is a periodic function with period 2π, and $\mathcal{A} = \mathbb{T}_n$, (i.e., the set of trigonometric polynomials of degree n, with basis consisting of the functions 1, $\cos x$, $\sin x$, ..., $\cos nx$, $\sin nx$).

Since this set of functions is orthogonal on $[0, 2\pi]$, with respect to the weight function $w(x) = 1$, we have

$$g^*(x) = a_0^* + \sum_{k=1}^{n} (a_k^* \cos kx + b_k^* \sin kx),$$

where

$$\begin{aligned} a_0^* &= \int_0^{2\pi} f(x) dx, \\ a_k^* &= \tfrac{1}{\pi} \int_0^{2\pi} f(x) \cos kx dx, \\ b_k^* &= \tfrac{1}{\pi} \int_0^{2\pi} f(x) \sin kx dx, \end{aligned}$$

for $k = 1, ..., n$. So a_k^* and b_k^* are the Fourier coefficients of f.

The discrete case can be treated similarly. Let f be a given function with available data $f(x_i)$, $i = 0, ..., m$, and $\mathcal{A}$ be a set of approximations of f. According to the least squares principle, we have to determine the function $g^* \in \mathcal{A}$ that satisfies

$$\sum_{i=0}^{m} w(x_i) [f(x_i) - g^*(x_i)]^2 dx = \min_{g \in \mathcal{A}} \sum_{i=0}^{m} w(x_i) [f(x_i) - g(x_i)]^2.$$

If $\mathcal{A}$ is a n-dimensional linear space and $g_1, ..., g_n$ is a basis of $\mathcal{A}$, then such an element g^* exists and it is given by

$$g^* = \sum_{k=1}^{n} a_k^* g_k,$$

where $(a_1^*, ..., a_n^*)$ is the solution of the following system of equations:

$$\sum_{k=1}^{n} a_k \sum_{i=0}^{m} w(x_i) g_k(x_i) g_j(x_i) = \sum_{i=0}^{m} w(x_i) g_j(x_i) f(x_i), \quad j = 1, ..., n. \tag{2.2.69}$$

Since a_k, $k = 1, ..., n$ are the coordinates of the minimizing point of the function

$$G(a_1, ..., a_n) = \sum_{i=0}^{m} w(x_i) \left[f(x_i) - \sum_{k=1}^{n} a_k g_k \right]^2,$$

the system (2.2.69) is equivalent to

$$\frac{\partial G}{\partial a_j}(a_1, ..., a_n) = 0, \quad j = 1, ..., n.$$

Remark 2.2.63. Often n is taken much less than m. Notice that, for $n = m - 1$ one has

$$\sum_{i=0}^{m} w(x_i) [f(x_i) - g^*(x_i)]^2 = 0,$$

implying

$$g(x_i) = f(x_i), \quad i = 0, ..., m,$$

so g^* is a function that interpolates f at the points x_i.

We also consider here two particular cases.

Case C1. $\mathcal{A} = \mathbb{P}_n$ and $g_k = e_k$, $k = 0, ..., n$.

We have

$$g^*(x) = \sum_{k=0}^{n} a_k^* x^k,$$

where $(a_0^*, ..., a_n^*)$ is the solution of the system of equations:

$$\sum_{k=0}^{n} a_k \sum_{i=0}^{m} w(x_i) x_i^{k+j} = \sum_{i=0}^{m} w(x_i) x_i^j f(x_i), \quad j = 0, ..., n. \tag{2.2.70}$$

Denoting by

$$s_k = \sum_{i=0}^{m} w(x_i) x_i^k, \quad k = 0, ..., 2n,$$

$$t_p = \sum_{i=0}^{m} w(x_i) x_i^p f(x_i), \quad p = 0, ..., n,$$

system (2.2.70) becomes

$$\sum_{k=0}^{n} s_{k+j} a_k = t_j, \quad j = 0, ..., n.$$

Since its matrix is symmetric, special methods can be used to solve the latter system of equations.

Remark 2.2.64. One of the major applications of the least squares approximation is the so-called data smoothing. Whenever it is necessary to approximate a function f on a discrete data set, and no other requirements are given, we can achieve this with a polynomial $p \in C^\infty(\mathbb{R})$, i.e., a very smooth function.

Case C2. $g_0, ..., g_n$ are orthogonal polynomials on the points set $\{x_i, i = 0, ..., m\}$, with respect to the weight functions $w(x_i)$, $i = 0, ..., m$, namely

$$\sum_{i=0}^{m} w(x_i) g_k(x_i) g_j(x_i) = \begin{cases} 0, & k \neq j, \\ c_j > 0, & k = j. \end{cases}$$

In this case, the system of equations (2.2.69) becomes

$$a_j c_j = \sum_{i=0}^{m} w(x_i) g_j(x_i) f(x_i), \quad j = 0, ..., n$$

and its solution is given by

$$a_k^* = \frac{1}{c_k} \sum_{i=0}^{m} w(x_i)\, g_k(x_i)\, f(x_i), \quad k = 0, ..., n.$$

2.3 Some elements of multivariate interpolation operators

The basic goal of this section is to construct multivariate interpolation operators based on univariate ones.

As we have already seen, most of the univariate interpolation operators (polynomial, spline, Shepard, etc.) are commuting projectors. This property is essential in construction of multivariate operators.

2.3.1 Interpolation on a rectangular domain

Let $\Omega^n \subset \mathbb{R}$ be a rectangular domain,

$$\Omega^n = \prod_{i=1}^{n} [a_i, b_i], \qquad a_i, b_i \in \mathbb{R},\ a_i < b_i,\ i = 1, ..., n.$$

Let $\mathcal{B}$ be a set of real-valued functions defined on Ω^n, and $f \in \mathcal{B}$. One considers the univariate projectors $P_i,\ P_i : \mathcal{B} \to G_i,\ i = 1, ..., n$, that interpolate the function f with respect to the variables $x_i,\ i = 1, ..., n$, respectively. Of course, each set $G_i,\ i = 1, ..., n$, is a set of $(n-1)$ free variables functions.

Let $\mathcal{P}_n$ be the set of all commuting projectors generated by the projectors $P_1, ..., P_n$ (i.e., $\bar{\mathcal{L}}$ of (1.1.6) generated by $\mathcal{L}$=$\{P_1, ..., P_n\}$).

For example,

$$\mathcal{P}_2 = \{P_1, P_2, P_1P_2, P_1 \oplus P_2\}$$

and

$$\begin{aligned}\mathcal{P}_3 = \{&P_1, P_2, P_3, P_1P_2, P_1P_3, P_2P_3, P_1 \oplus P_2, P_1 \oplus P_3, P_2 \oplus P_3,\\ &P_1(P_2 \oplus P_3), P_2(P_1 \oplus P_3), P_3(P_1 \oplus P_2), P_1 \oplus P_2P_3, P_2 \oplus P_1P_3,\\ &P_3 \oplus P_1P_2, P_1P_2P_3, P_1 \oplus P_2 \oplus P_3\}.\end{aligned}$$

Let $(\mathcal{P}_n, \leq)$ be the lattice generated by $\{P_1, ..., P_n\}$, where " $\leq$ " is the order relation defined in Proposition 1.1.30. In this lattice $P = P_1...P_n$ is the minimal element and $S = P_1 \oplus ... \oplus P_n$ is the maximal element.

Let $R_i = I - P_i,\ i = 1, ..., n$, be the corresponding remainder operators. The operator P_i being projectors, from Remark 1.1.23 it follows that R_i is also a projector and P_i and R_i commute, i.e., $(P_iR_i = R_iP_i)$. Using the identities

$$I = P + R_P,$$

with

$$P = P_1...P_n,$$
$$R_P = R_1 \oplus ...R_n,$$

respectively,

$$I = S + R_S,$$

with

$$S = P_1 \oplus ... \oplus P_n,$$
$$R_S = R_1...R_n,$$

one obtains *the algebraic minimal interpolation formula:*

$$f = Pf + R_Pf,$$

respectively, *the algebraic maximal interpolation formula:*

$$f = Sf + R_Sf,$$

characteristics given by the extremal properties of P and S in the lattice $(\mathcal{P}_n, \leq)$.

The extremal projectors P and S of $(\mathcal{P}_n, \leq)$ can also be characterized by their approximation order.

From the expression of the corresponding remainder operators,

$$\begin{aligned} R_p &= R_1 \oplus ... \oplus R_n \\ &= R_1 + ... + R_n - R_1R_2 - ... - R_{n-1}R_n + ... + (-1)^{n-1}R_1...R_n \end{aligned}$$

and

$$R_S = R_1...R_n,$$

it follows that

$$\operatorname{ord}(P) = \min\{\operatorname{ord}(P_1), ..., \operatorname{ord}(P_n)\}, \tag{2.3.1}$$

respectively,

$$\text{ord}(S) = \sum_{i=1}^{n} \text{ord}(P_i) \tag{2.3.2}$$

and

$$\text{ord}(P) < \text{ord}(Q) < \text{ord}(S), \quad \forall Q \in \mathcal{P}_n.$$

So, the remarkable property of the boolean sum operator S, in $\mathcal{P}_n$, is that it has the highest approximation order. One the other hand, taking into account that

$$S = P_1 + ... + P_n - P_1P_2 - ... - P_{n-1}P_n + ... + (-1)^{n-1}P_1...P_n,$$

it follows that the interpolation function Sf is a sum of functions of $(n-1)$ free variables $(P_if,\ i = 1, ..., n)$, $(n-2)$ free variables $(P_iP_jf,\ i, j = 1, ..., n;\ i \neq j)$ and so on, only the last one $P_1...P_nf$ being a scalar approximation (it doesn't contain free variable), while the operator P, with the lowest approximation order, gives a scalar approximation Pf.

Remark 2.3.1. From the non-scalar approximation Sf it can be obtained a scalar approximation using several levels of approximation.

We illustrate this for the two dimensional case. Let P_1^1 and P_2^1 be the interpolation operators used in the first approximation level and R_1^1, respectively R_2^1, the corresponding remainder operators $(R_1^1 = I - P_1^1;\ R_2^1 = I - P_2^1)$. We have

$$f = P_1^1 \oplus P_2^1 f + R_1^1 R_2^1 f, \tag{2.3.3}$$

or

$$f = (P_1^1 + P_2^1 - P_1^1 P_2^1)f + R_1^1 R_2^1 f.$$

As $P_1^1 f$ and $P_2^1 f$ contain a free variable, using in a second level of approximation the operators P_1^2, P_2^2, with R_1^2, R_2^2 the corresponding remainders, one obtains:

$$f = [P_1^1(P_2^2 + R_2^2) + P_2^1(P_1^2 + R_1^2) - P_1P_2^1]f + R_1^1 R_2^1 f,$$

respectively,

$$f = (P_1^1 P_2^2 + P_1^2 P_2^1 - P_1^1 P_2^1)f + (P_1^1 R_2^2 + P_2^1 R_1^2 + R_1^1 R_2^1)f, \tag{2.3.4}$$

where

$$Q = P_1^1 P_2^2 + P_1^2 P_2^1 - P_1^1 P_2^1, \tag{2.3.5}$$

is the interpolation operator, while

$$R_Q = P_1^1 R_2^2 + P_2^1 R_1^2 + R_1^1 R_1^2 \tag{2.3.6}$$

is the remainder operator.

This way, one obtains the formula (2.3.4), which is a scalar interpolation formula and Q is a scalar interpolation operator.

As it can be seen, the remainder operator R_Q contains three terms. It follows that

$$\operatorname{ord}(Q) = \min\{\operatorname{ord}(P_1^2), \operatorname{ord}(P_2^2), \operatorname{ord}(P_1^1) + \operatorname{ord}(P_2^1)\}.$$

Remark 2.3.2. 1) If

$$\operatorname{ord}(P_1^2) \geq \operatorname{ord}(P_1^1) + \operatorname{ord}(P_2^1) = \operatorname{ord}(P_1^1 \oplus P_2^1)$$

and

$$\operatorname{ord}(P_2^2) \geq \operatorname{ord}(P_1^1) + \operatorname{ord}(P_2^1) = \operatorname{ord}(P_1^1 \oplus P_2^1),$$

then (2.3.4) is called *a consistent approximation formula*.

2) If

$$\operatorname{ord}(P_1^2) = \operatorname{ord}(P_2^2) = \operatorname{ord}(P_1^1 \oplus P_2^1),$$

then (2.3.4) is called *a homogeneous approximation formula*.

Next, we give some examples of product and boolean sum formulas.

Let us denote by $D_h = [0,h] \times [0,h]$, $h > 0$, the so-called standard rectangle and let consider $f : D_h \to \mathbb{R}$.

Example 2.3.3. Let L_1^x and L_1^x be the univariate Lagrange-type interpolation operators defined by

$$(L_1^x f)(x,y) = \tfrac{h-x}{h} f(0,y) + \tfrac{x}{h} f(h,y), \tag{2.3.7}$$

respectively,

$$(L_1^y f)(x,y) = \tfrac{h-y}{h} f(x,0) + \tfrac{y}{h} f(x,h). \tag{2.3.8}$$

It is easy to check that

$$\begin{aligned} &(L_1^x f)(0,y) = f(0,y),\ (L_1^x f)(h,y) = f(h,y), \quad y \in [0,h] \\ &(L_1^y f)(x,0) = f(x,0),\ (L_1^y f)(x,h) = f(x,h), \quad x \in [0,h]. \end{aligned} \tag{2.3.9}$$

Let $R_1^x = I - L_1^x$ and $R_l^y = I - L_1^y$ be the corresponding remainder operators, i.e.,

$$\begin{aligned}(R_1^x f)(x,y) &= \tfrac{x(x-h)}{2} f^{(2,0)}(\xi, y), \text{ for } f^{(2,0)}(\cdot, y) \in C^2[0,h],\ y \in [0,h] \\ (R_1^y f)(x,y) &= \tfrac{y(y-h)}{2} f^{(0,2)}(x, \eta), \text{ for } f^{(0,2)}(x, \cdot) \in C^2[0,h],\ x \in [0,h]\end{aligned} \tag{2.3.10}$$

with $\xi, \eta \in [0,h]$ and

$$\begin{aligned}|(R_1^x f)(\cdot, y)| &\le \tfrac{h^2}{8} \left\| f^{(2,0)}(\cdot, y \right\|_\infty, \\ |(R_1^y f)(x, \cdot)| &\le \tfrac{h^2}{8} \left\| f^{(0,2)}(x, \cdot) \right\|_\infty.\end{aligned} \tag{2.3.11}$$

Hence,

$$\operatorname{ord}(L_1^x) = \operatorname{ord}(L_1^y) = 2.$$

Because in this case $\mathcal{P}_2$ is

$$\mathcal{P}_2 = \{L_1^x, L_1^y, L_1^x L_1^y, L_1^x \oplus L_1^y\}$$

the bivariate operators are only $L_1^x L_1^y$ and $L_1^x \oplus L_1^y$, i.e., the extremal operators of the lattice $(\mathcal{P}_2, \le)$.

So, we have the algebraic minimal approximation formula:

$$f = L_1^x L_1^y f + R_1^x \oplus R_1^y f, \tag{2.3.12}$$

respectively, the algebraic maximal approximation formula:

$$f = L_1^x \oplus L_1^y f + R_1^x R_1^y f. \tag{2.3.13}$$

From (2.3.7) and (2.3.8) it follows that

$$\begin{aligned}(L_1^x L_1^y f)(x,y) = &\tfrac{(h-x)(h-y)}{h^2} f(0,0) + \tfrac{x(h-y)}{h^2} f(h,0) \\ &+ \tfrac{y(h-x)}{h^2} + \tfrac{xy}{h^2} f(h,h)\end{aligned} \tag{2.3.14}$$

and

$$\begin{aligned}(L_1^x \oplus L_1^y f)(x,y) = &\tfrac{h-x}{h} f(0,y) + \tfrac{x}{h} f(h,y) + \tfrac{h-y}{h} f(x,0) \\ &+ \tfrac{y}{h} f(x,h) - \tfrac{(h-x)(h-y)}{h^2} f(0,0) \\ &- \tfrac{x(h-y)}{h^2} f(h,0) - \tfrac{y(h-x)}{h^2} f(0,h) - \tfrac{xy}{h^2}.\end{aligned} \tag{2.3.15}$$

Also, from (2.3.10), one obtains

$$(R_1^x \oplus R_1^y f)(x,y) = \frac{x(x-h)}{2} f^{(2,0)}(\xi, y) + \frac{y(y-h)}{2} f^{(0,2)}(x, \eta) - \frac{xy(x-h)(y-h)}{4} f^{(2,2)}(\xi_1, \eta_1) \tag{2.3.16}$$

and

$$(R_1^x R_1^y f)(x,y) = \frac{xy(x-h)(y-h)}{4} f^{(2,2)}(\xi, \eta). \tag{2.3.17}$$

Remark 2.3.4. 1) The approximation $L_1^x L_1^y f$ of f uses the information $\Lambda(f) = \{f(0,0),\ f(h,0), f(0,h), f(h,h)\}$ of f, which is a scalar information. So, (2.3.12) is a scalar approximation formula.

The information on f used by the approximation $L_1^x \oplus_1^y f$ is $\Lambda(f) = \{f(0,0), f(h,0), f(0,h), f(h,h), f(0,y), f(h,y), f(x,0), f(x,h)\}$, for $x, y \in [0,h]$, that is not a scalar information, i.e., (2.3.13) is a non-scalar approximation formula.

2) $L_1^x L_1^y f$ interpolates the function f at the vertices V_i, $i = 1, ..., 4$ of D_h:

$$(L_1^x L_1^y f)(V_i) = f(V_i), \qquad i = 1, ..., 4,$$

i.e., the product $L_1^x L_1^y$ is a punctual interpolation operator.

$L_1^x \oplus L_1^y f$ interpolates f at the border ∂D_h of D_h, i.e.:

$$L_1^x \oplus L_1^y f \,|_{\partial D_h} = f|_{\partial D_h}.$$

So, $L_1^x \oplus L_1^y$ is a transfinite interpolation operator, also called *a blending interpolation operator.*

3) Regarding the approximation order, from (2.3.16), (2.3.17) and (2.3.11), it follows

$$\|R_1^x \oplus R_1^y f\|_\infty \leq \frac{h^2}{8} \left(\left\|f^{(2,0)}\right\|_\infty + \left\|f^{(0,2)}\right\|_\infty + \frac{h^2}{8} \left\|f^{(2,2)}\right\|_\infty \right),$$

respectively,

$$\|R_1^x R_1^y f\|_\infty \leq \frac{h^4}{64} \left\|f^{(2,2)}\right\|_\infty,$$

i.e.,

$$\begin{aligned} &\mathrm{ord}(L_1^x L_1^y) = 2, \\ &\mathrm{ord}(L_1^x \oplus L_1^y) = 4, \end{aligned}$$

which is in concordance with (2.3.1) and (2.3.2).

Now, using a second approximation level, from the boolean sum interpolation formula (2.3.13), we obtain a homogenous formula. Taking into account that $\mathrm{ord}(L_1^x \oplus L_1^y) = 4$, in a second level, we must use interpolation operators of order 4. Such operators are, for example, the Hermite operators H_3^x and H_3^y that interpolate the function f at the double nodes 0 and h, i.e.,

$$(H_3^x f)(x,y) = h_{00}(x)f(0,y)+h_{01}(x)f^{(1,0)}(0,y)+h_{10}(x)f(h,y)+h_{11}(x)f^{(1,0)}(h,y),$$

respectively,

$$(H_3^y f)(x,y) = h_{00}(y)f(x,0)+h_{01}(y)f^{(0,1)}(x,0)+h_{10}(y)f(x,h)+h_{11}(y)f^{(0,1)}(x,h),$$

where

$$\begin{array}{ll} h_{00}(x) = \frac{1}{h^3}(x-h)^2(2x+h), & h_{01}(x) = \frac{x(x-h)^2}{h^2}, \\ h_{10}(x) = \frac{1}{h^3}x^2(3h-2x), & h_{11}(x) = \frac{x^2(x-h)}{h^2}. \end{array}$$

For the remainder terms, we have

$$\begin{aligned} (R_3^x f)(x,y) &= \tfrac{x^2(x-h)^2}{24} f^{(4,0)}(\xi,y), \\ (R_3^y f)(x,y) &= \tfrac{y^2(y-h)^2}{24} f^{(0,4)}(x,\eta), \end{aligned}$$

with

$$\mathrm{ord}(H_3^x) = \mathrm{ord}(H_3^y) = 4.$$

So,

$$\mathrm{ord}(H_3^x) = \mathrm{ord}(H_3^y) = \mathrm{ord}(L_1^x \oplus L_1^y) = 4.$$

hence,

$$f = (L_1^x H_3^y + H_3^x L_1^y - L_1^x L_1^y)f + (L_1^x R_3^y + L_1^y R_3^x + R_1^x R_1^y)f$$

is a homogenous interpolation formula of order 4.

Remark 2.3.5. From a boolean sum interpolation formula, it can always be obtained a homogenous formula. The condition is to choose, in the second level of approximation, some interpolation operators of a suitable degree. This is always possible.

2.3.2 Interpolation on a simplex domain

We consider only the bivariate case, with the standard triangle

$$T_h = \{(x, y) \in \mathbb{R}^2 \mid x \geq 0, y \geq 0, x + y \leq h\}, \qquad h > 0.$$

Let us consider the univariate operators P_1, P_2 and P_3 which interpolate a given function $f : T_h \to \mathbb{R}$ on the set $\{(0, y), (h - y, y)\}, \{(x, 0), (x, h - x)\}$ and, respectively, on $\{(x + y, 0), (0, x + y)\}$, i.e.,

$$\begin{aligned}
(P_1 f)(x, y) &= \tfrac{h-x-y}{h-y} f(0, y) + \tfrac{x}{h-y} f(h - y, y), \\
(P_2 f)(x, y) &= \tfrac{h-x-y}{h-y} f(x, 0) + \tfrac{y}{h-x} f(x, h - x), \\
(P_3 f)(x, y) &= \tfrac{x}{x+y} f(x + y, 0) + \tfrac{y}{x+y} f(0, x + y).
\end{aligned} \tag{2.3.18}$$

Using the product and the boolean sum of these operators, we construct bivariate operators with punctual or transfinite interpolation properties.

Example 2.3.6. Let

$$P = P_1 P_2 P_3.$$

We have

$$(Pf)(x, y) = \tfrac{h-x-y}{h} f(0, 0) + \tfrac{x}{h} f(h, 0) + \tfrac{y}{h} f(0, h).$$

So, P is a punctual interpolation operator that interpolates f at the vertices of T_h. Also,

$$\operatorname{dex}(P) = 1,$$

i.e.,

$$Pf = f, \quad \text{for } f \in \mathbb{P}_1^2.$$

Let

$$f = Pf + Rf \tag{2.3.19}$$

be the corresponding interpolation formula, where R denotes the remainder operator.

If $f \in B_{1,1}[0, 0]$ then by Peano's theorem, one obtains

$$\begin{aligned}
(Rf)(x, y) = &\int_0^h \varphi_{20}(x, y; s) f^{(2,0)}(s, 0) dt + \int_0^h \varphi_{02}(x, y; t) f^{(0,2)}(0, t) dt \\
&+ \iint_{T_h} \varphi_{11}(x, y; s, t) f^{(1,1)}(s, t) ds dt,
\end{aligned}$$

where

$$\varphi_{20}(x,y;s) = (x-s)_+ - \tfrac{x}{h}(h-s) \le 0, \quad s \in [0,h]$$
$$\varphi_{02}(x,y;t) = (y-t)_+ - \tfrac{y}{h}(h-t) \le 0, \quad t \in [0,h]$$
$$\varphi_{11}(x,y;s,t) = (x-s)_+^0 (y-t)_+^0 \ge 0, \quad (s,t) \in T_h.$$

As the Peano's kernels φ_{20}, φ_{02} and φ_{11} do not change the sign on $[0,h]$, respectively on T_h, by the mean value theorem, we obtain:

$$(Rf)(x,y) = \tfrac{x(x-h)}{2} f^{(2,0)}(\xi,0) + \tfrac{y(y-h)}{2} f^{(0,2)}(0,\eta) + xy f^{(1,1)}(\xi_1,\eta_1),$$

where $\xi, \eta \in [0,h]$ and $(\xi_1,\eta_1) \in T_h$. It follows that:

$$|(Rf)(x,y)| \le \tfrac{h^2}{4}\left[\tfrac{1}{2}\left\|f^{(2,0)}(\cdot,0)\right\|_{L^\infty[0,h]} + \tfrac{1}{2}\left\|f^{(0,2)}(0,\cdot)\right\|_{L^\infty[0,h]} + \left\|f^{(1,1)}\right\|_{L^\infty(T_h)}\right],$$

which implies that (2.3.19) is a homogenous interpolation formula of order 2 ($\mathrm{ord}(P) = 2$).

Remark 2.3.7. We have

1) $P_i \oplus P_j f\,|_{\partial T_h} = f|_{\partial T_h}, \quad \forall i,j = 1,2,3;\ i \ne j$
2) $\mathrm{dex}(P_i \oplus P_j) = 2.$

These properties are easy to verify.

Example 2.3.8. If $f^{(1,0)}(0,y),\ y \in [0,h]$ exists, one considers the Birkhoff type operator B_1, defined by:

$$(B_1 f)(x,y) = f(h-y,y) + (x+y-h) f^{(1,0)}(0,y).$$

For $y \in [0,h]$, it is easy to check that

$$(B_1 f)(h-y,y) = f(h-y,y),$$
$$(B_1 f^{(1,0)}(0,y) = f^{(1,0)}(0,y).$$

Let $Q := B_1 \oplus P_2$. We have

1)

$$(Qf)(x,0) = f(x,0),$$
$$(Qf)(h-y,y) = f(h-y,y),$$
$$(Qf)^{(1,0)}(0,y) = f^{(1,0)}(0,y), \qquad x,y \in [0,h].$$

2)

$$\operatorname{dex}(Q) = 2.$$

Indeed,

$$\begin{aligned}(Qf)(x,y) =& \tfrac{h-x-y}{h-x} f(x,0) + \tfrac{y}{h-x} f(x,h-x) \\ &+ (x+y-h) f^{(1,0)}(0,y) + (h-x-y)\left\{ \tfrac{y}{h^2}[f(0,h) - f(0,0)] \right. \\ &\left. + \tfrac{h-y}{h} f^{(1,0)}(0,0) + \tfrac{y}{h}[f^{(1,0)}(0,h) - f^{(0,1)}(0,h)] \right\},\end{aligned}$$

and the properties 1) and 2) are verified by a straightforward computation.

One considers the interpolation formula

$$f = Qf + Rf, \tag{2.3.20}$$

with Rf the remainder term. If $f \in B_{12}[0,0]$ then by Peano's theorem, one obtains

$$|(Rf)(x,y)| \leq \tfrac{h^3}{27} \left[\tfrac{2}{3} \left\| f^{(0,3)}(0,\cdot) \right\|_{L^\infty[0,h]} + \tfrac{1}{2} \left\| f^{(1,2)} \right\|_{L^\infty(T_h)} \right].$$

So, (2.3.20) is a homogenous blending interpolation formula of Birkhoff-type and

$$\operatorname{ord}(Q) = 3.$$

2.3.3 Bivariate Shepard interpolation

Let $X \subset \mathbb{R}^2$ be an arbitrary domain, $f : X \to \mathbb{R}$ and $Z = \{z_i \mid z_i = (x_i, y_i) \in X, \quad i = 1, ..., N\}$ be the set of the interpolation nodes. In 1968, Donald Shepard introduced a new kind of interpolation procedure, where the fundamental interpolation functions on a point $z := (x, y)$ are defined using the distances from the point (x, y) to the interpolation nodes (x_i, y_i), $i = 1, ..., N$.

The bivariate Shepard operator, denoted by S_0, is defined by:

$$(S_0 f)(x,y) = \sum_{i=1}^{N} A_i(x,y) f(x_i, y_i), \tag{2.3.21}$$

with

$$A_i(x,y) = \frac{\prod\limits_{\substack{j=1 \\ j \neq i}}^{N} \rho_j^\mu(x,y)}{\sum\limits_{k=1}^{N} \prod\limits_{\substack{j=1 \\ j \neq k}}^{N} \rho_j^\mu(x,y)}, \tag{2.3.22}$$

where $\mu \in \mathbb{R}_+$ and ρ is a metric on $\mathbb{R}^2$. Usually,

$$\rho_j(x,y) = ((x-x_j)^2 + (y-y_j)^2)^{1/2}.$$

The functions A_i, $i = 1, ..., N$, can also be written as

$$A_i(x,y) = \frac{\frac{1}{\rho_i^\mu(x,y)}}{\sum\limits_{j=1}^{N} \frac{1}{\rho_j^\mu(x,y)}}$$

and

$$(S_0 f)(x,y) = \left(\sum_{i=1}^{N} \frac{f(x_i,y_i)}{\rho_i^\mu(x,y)} \right) \Bigg/ \left(\sum_{j=1}^{N} \frac{1}{\rho_i^\mu(x,y)} \right).$$

It is easy to verify that

$$A_i(x_j,y_j) = \delta_{ij} \tag{2.3.23}$$

and

$$\sum_{i=1}^{N} A_i(x,y) = 1. \tag{2.3.24}$$

The main properties of Shepard operator S_0 are:

- Interpolation properties:

$$(S_0 f)(x_i,y_i) = f(x_i,y_i), \quad i = 1, ..., N.$$

- Degree of exactness is:

$$\operatorname{dex}(S_0) = 0.$$

$$\min_{i=1,...,N} f(x_i,y_i) \leq (S_0 f)(x,y) \leq \max_{i=1,...,N} f(x_i,y_i).$$

These properties are implied by the relations (2.3.23) and (2.3.24).

We notice that the drawbacks mentioned for the univariate Shepard method are also maintained in bivariate case, namely:

- high computational cost;
- low degree of exactness.

Two ways to overcome these drawbacks are:

★ *Modifying basis functions,* for example, using local version of Shepard formula, introduced by Franke and Nielson. In this case the Shepard operator is of the form:

$$\left(S^w f\right)(x,y) = \frac{\sum_{i=0}^{N} W_i(x,y)\, f(x_i,y_i)}{\sum_{i=0}^{N} W_i(x,y)}, \tag{2.3.25}$$

with

$$W_i(x,y) = \left[\frac{(R_w - r_i)_+}{R_w r_i}\right]^2, \tag{2.3.26}$$

where R_w is a radius of influence about the node (x_i, y_i) and it is varying with i.

★ *Increasing the degree of exactness* by combining the Shepard operator with another interpolation operator. In this way its degree of exactness is increased and there are used another sets of functionals. The general procedure is presented in Section 2.2.6.2.

We have some similar results with that given for univariate case.

Remark 2.3.9. If P_i, $i = 0, ..., N$, are linear operators then S_P is also a linear operator.

Remark 2.3.10. Let P_i, $i = 0, ..., N$, be some linear operators. If

$$\operatorname{dex}(P_i) = \rho_i, \qquad i = 0, ..., N,$$

then

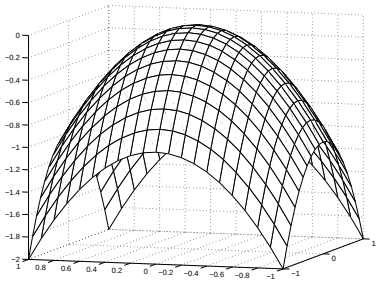
$$\operatorname{dex}(S_P) = \min\{\rho_0, ..., \rho_N\}.$$

Remark 2.3.11. As $A_i(x,y) \geq 0$, for $(x,y) \in X$, it follows that S_0 is a positive operator.

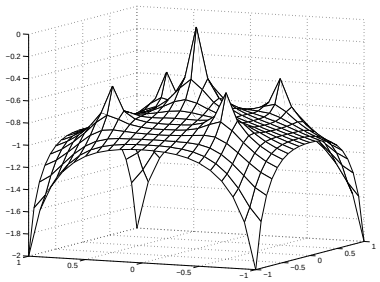
Remark 2.3.12. The exponent $\mu \in \mathbb{R}_+$ can be chosen arbitrary. If $0 \leq \mu \leq 1$ the function $S_0 f$ has peaks at the nodes, for $\mu > 1$ it is level at nodes and for μ large enough $S_0 f$ becomes a step function.

We illustrate this phenomenon by some examples.

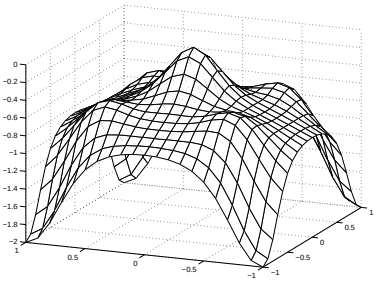
Example 2.3.13. Let $f : [-1,1] \times [-1,1] \to \mathbb{R}$, $f(x,y) = -(x^2 + y^2)$ and consider the nodes $z_1 = (-1,-1)$, $z_2 = (-1,1)$, $z_3 = (1,-1)$, $z_4 = (1,1)$, $z_5 = (-0.5,-0.5)$, $z_6 = (-0.5,0.5)$, $z_7 = (0.5,-0.5)$, $z_8 = (0.5,0.5)$, $z_9 = (0,0)$. In Figure 2.2 we plot $S_0 f$, for $\mu = 1, 2, 20$.



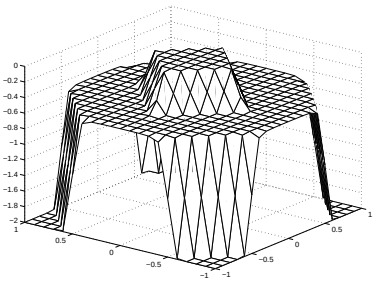
(a) Function f.



(b) Interpolant $S_0 f$, $\mu = 1$.



(c) Interpolant $S_0 f$, $\mu = 2$.



(d) Interpolant $S_0 f$, $\mu = 20$.

Figure 2.2: Bivariate Shepard interpolants.

2.3.3.1 The bivariate Shepard-Lagrange operator

Let $f : X \to \mathbb{R}$, $\mathbb{X} \subset \mathbb{R}^2$ and $Z = \{z_i \mid z_i = (x_i, y_i) \in X, \;\; i = 1, ..., N\}$ be the set of the interpolation nodes. Consider the set of Lagrange type functionals

$$\Lambda_L(f) = \{\lambda_i \mid \lambda_i(f) = f(x_i, y_i),\, i = 1, \ldots, N\}\,.$$

Taking into account that $\mathrm{dex}(S_0) = 0$, our goal is to construct Shepard-type operators of higher degree of exactness. To do this, it will be used a combination of the Shepard operator S_0 with some Lagrange operator for bivariate functions.

Let $L_i f$ be the bivariate n degree Lagrange polynomial that interpolates the function f, respectively, at the sets of points

$$Z_{m,i} := \{z_i, z_{i+1}, \ldots, z_{i+m-1}\}\,,\; i = 1, \ldots, N,\; m < N \tag{2.3.27}$$

with $z_{N+i} = z_i$, $i = 1, \ldots, m-1$ and $m := (n+1)(n+2)/2$ being the number

of the coefficients of a bivariate polynomial of the degree n,

$$P_n(x,y) = \sum_{i+j\le n} a_{ij}x^i y^j.$$

Remark 2.3.14. For given N, it can be considered only operators L_i^n with n such that $(n+1)(n+2)/2 < N$, i.e., for $n \in \{1,\dots,\nu\}$, where $\nu = \text{integer}[(\sqrt{8N+1}-3)/2]$. The existence and the uniqueness of the operators L_i^n are assured by the following theorem.

Theorem 2.3.15. *Let $z_i := (x_i, y_i)$, $i = 1,\dots,(n+1)(n+2)/2$ be different points in plane that do not lie on the same algebraic curve of n-th degree ($\sum_{i+j\le n} a_{ij}x^i y^j = 0$). Then, for every function f defined at the points z_i, $i = 1,\dots,(n+1)(n+2)/2$ there exists a unique polynomial Q_n of n-th degree that interpolates f at z_i, i.e.,*

$$Q_n(x_i,y_i) = f(x_i,y_i), \quad i = 1,\dots,(n+1)(n+2)/2.$$

Hence, if the points z_k, $k = i,\dots,i+m-1$; $i = 1,\dots,N$, of the set (2.3.27), do not lie on an algebraic curve of n-th degree, then L_i^n exists and it is unique for all $i = 1,\dots,N$.

Suppose that the existence and uniqueness conditions of the operators L_i^n, $i = 1,\dots,N$, are satisfied.

We have

$$(L_i^n f)(x,y) = \sum_{k=i}^{i+m-1} l_k(x,y) f(x_k,y_k), \quad i = 1,\dots,N,$$

where l_k are the corresponding cardinal polynomials

$$l_k(x_j,y_j) = \delta_{kj}, \quad k,j = i,\dots,i+m-1.$$

The main properties of the Lagrange interpolation operators L_i^n are:

$$(L_i^n f)(x_k,y_k) = f(x_k,y_k), \quad k = i,\dots,i+m-1 \tag{2.3.28}$$

and

$$\text{dex}(L_i^n) = n, \quad i = 1,\dots,N. \tag{2.3.29}$$

Definition 2.3.16. *The operator* S_n^L *given by*

$$(S_n^L f)(x,y) = \sum_{i=1}^{N} A_i(x,y)(L_i^n f)(x,y) \tag{2.3.30}$$

is called the bivariate Shepard-Lagrange operator.

Remark 2.3.17. As the Lagrange operators L_i^n, $i = 1, \ldots, N$ and the Shepard operator S_0 are linear operators, it follows that the combined operator S_n^L is also linear.

Theorem 2.3.18. *Let* $f : D \to \mathbb{R}$ *be a given function,* $z_i := (x_i, y_i) \in D$, $i = 1, \ldots, N$ *and* $m := (n+1)(n+2)/2$. *If the points* $z_i, \ldots, z_{i+m-1}$ *do not lie on an algebraic curve of* n*-th degree, for all* $i = 1, \ldots, N$ ($z_{N+k} := z_k$, $k = 1, \ldots, m-1$), *then the combined operator* S_n^L *exists and it have the following properties:*

$$(S_n^L f)(x_j, y_j) = f(x_j, y_j), \qquad j = 1, \ldots, N \tag{2.3.31}$$

and

$$\operatorname{dex}(S_n^L) = n. \tag{2.3.32}$$

Proof. From (2.3.28) it follows

$$(S_n^L f)(x_j, y_j) = \sum_{i=1}^{N} A_i(x_j, y_j)(L_i^n f)(x_j, y_j) = \sum_{i=1}^{N} A_i(x_j, y_j) f(x_j, y_j)$$

and (2.3.23) implies the interpolation property (2.3.31).

Relation (2.3.32) follows by Remark 2.3.10, taking into account that $\operatorname{dex}(L_i^n) = n$. ■

Next, one considers two particular cases: $n = 1$ and $n = 2$.

1) Case $n = 1$. We have

$$(S_1^L f)(x,y) = \sum_{i=1}^{N} A_i(x,y)(L_i^1 f)(x,y), \tag{2.3.33}$$

where

$$(L_i^1 f)(x,y) = l_i(x,y) f(x_i, y_i) + l_{i+1}(x,y) f(x_{i+1}, y_{i+1}) + l_{i+2}(x,y) f(x_{i+2}, y_{i+2}),$$

for $i = 1, \ldots, N$ ($z_{N+1} := z_1, z_{N+2} := z_2$). One obtains

$$l_i(x,y) = \frac{(y_{i+1}-y_{i+2})x+(x_{i+2}-x_{i+1})y+x_{i+1}y_{i+2}-x_{i+2}y_{i+1}}{(x_i-x_{i+1})(y_{i+1}-y_{i+2})-(x_{i+1}-x_{i+2})(y_i-y_{i+1})},$$
$$l_{i+1}(x,y) = \frac{(y_{i+2}-y_i)x+(x_i-x_{i+2})y+x_{i+2}y_i-x_iy_{i+2}}{(x_{i+1}-x_{i+2})(y_{i+2}-y_i)-(x_{i+2}-x_i)(y_{i+1}-y_{i+2})},$$
$$l_{i+2}(x,y) = \frac{(y_i-y_{i+1})x+(x_{i+1}-x_i)y+x_iy_{i+1}-x_{i+1}y_i}{(x_{i+2}-x_i)(y_i-y_{i+1})-(x_i-x_{i+1})(y_{i+2}-y_i)}.$$

Remark 2.3.19. The existence and uniqueness condition of L_i^1 is that the points $z_i,\ z_{i+1},\ z_{i+2}$ do not lie on a line $Ax + By + C = 0$, or have to be the vertices of a non-degenerate triangle Δ_i, for all $i = 1, \ldots, N$.

Remark 2.3.20. The functions $S_1^L f$ and $S_0 f$ use the same information about f ($f(x_i, y_i),\ i = 1, \ldots, N$), but $\operatorname{dex}(S_1^L) = 1$, while $\operatorname{dex}(S_0) = 0$.

2) Case $n = 2$. From (2.3.30), we obtain

$$(S_2^L f)(x,y) = \sum_{i=1}^{N} A_i(x,y)(L_i^2 f)(x,y), \tag{2.3.34}$$

where $L_i^2 f$ are two-degree polynomials that interpolate f at $z_i, \ldots, z_{i+5}$, respectively.

In accordance with Theorem 2.3.18, the interpolation nodes $z_i, \ldots, z_{i+5}$ must not lie, respectively, on an algebraic curve of second degree g_i, $i = 1, \ldots, N$. If, for some j, $1 \le j \le N$ this condition is not satisfied, the index order of z_i, $i = 1, \ldots, N$ can be changed.

Remark 2.3.21. A Shepard-Lagrange operator of two-degree of exactness can be obtained using more information about f. Roughly speaking, the only problem is to exist some two-degree Lagrange type polynomials which interpolate f at z_k, $k = 1, \ldots, N$.

A sufficient condition for the existence and uniqueness of such polynomials, say $\widetilde{L}_i^2 f$, is that the six interpolation nodes to be the vertices z_i, z_{i+1}, z_{i+2} of the triangle Δ_i and the midpoints, say ξ_i, ξ_{i+1}, ξ_{i+2}, of the sides of Δ_i, for $i = 1, \ldots, N$.

Using the polynomials $\widetilde{L}_i^2 f,\ i = 1, \ldots, N$, one can define the Shepard operator $\widetilde{S}_2^L$:

$$(\widetilde{S}_2^L f)(x,y) = \sum_{i=1}^{N} A_i(x,y)(\widetilde{L}_i^2 f)(x,y),$$

which interpolates f at z_i, $i = 1, \ldots, N$ and $\operatorname{dex}(\widetilde{S}_2^L) = 2$.

Of course, $\widetilde{S}_2^L f$ also uses the values of the function f at the midpoints $\xi_i,\ i = 1, \ldots, N+2$. But, the advantage of $\widetilde{S}_2^L f$ is that the existence and uniqueness conditions of the polynomials $\widetilde{L}_i^2 f$ are more simple to verify.

An extension of the Shepard-Lagrange operator type S_n^L, is obtained if the Lagrange polynomials $L_i f$, used in combination with the Shepard operator S_0, are of different degree, say n_i, $i = 1, \ldots, N$, such that $(n_i+1)(n_i+2)/2 < N$.

So, let

$$Z_{m_i,i} := \{z_i, z_{i+1}, \ldots, z_{i+m_i-1}\},\ i = 1, \ldots, N,$$

be some sets of m_i interpolation nodes, respectively, with $m_i = (n_i+1)(n_i+2)/2$. Using the given information on f at the nodes $z_i \in Z_{m_i,i}$, there are constructed the corresponding Lagrange interpolation operators $L_i^{n_i}$, $i = 1, \ldots, N$.

Definition 2.3.22. *The operator* $S^L_{n_1,\ldots,n_N}$ *defined by*

$$(S^L_{n_1,\ldots,n_N} f)(x,y) = \sum_{i=1}^{N} A_i(x,y)(L_i^{n_i} f)(x,y) \tag{2.3.35}$$

is called a general Shepard-Lagrange operator.

Remark 2.3.23. For $n_1 = \cdots = n_N = n$ the operator $S^L_{n_1,\ldots,n_N}$ becomes S_n^L.

Theorem 2.3.24. *Suppose that* $S^L_{n_1,\ldots,n_N}$ *exists. Then*

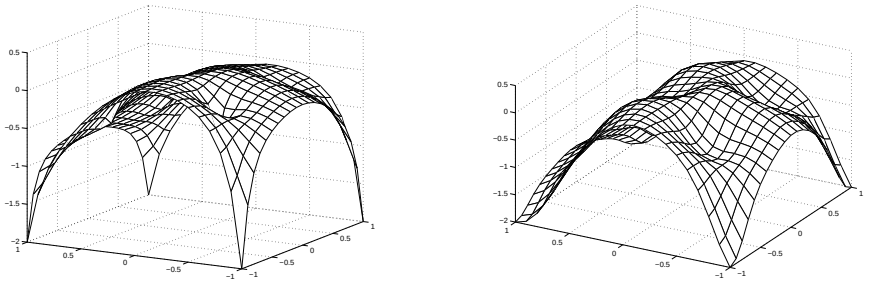
$$(S^L_{n_1,\ldots,n_N} f)(x_i,y_i) = f(x_i,y_i),\quad i = 1, \ldots, N$$

and

$$\operatorname{dex}(S^L_{n_1,\ldots,n_N}) = \min\{n_1, \ldots, n_N\}.$$

Proof. The proof follows by (2.3.23) and Remark 2.2.43. ■

Example 2.3.25. For f the same as in Example 2.3.13 and the same nodes, in Figure 2.3 we plot $S_1^L f$, for $\mu = 1; 2$.



(a) Interpolant $S_1^L f$, $\mu = 1$. (b) Interpolant $S_1^L f$, $\mu = 2$.

Figure 2.3: Bivariate Shepard-Lagrange interpolants.

2.3.3.2 The bivariate Shepard-Hermite operator

A new extension of Shepard operator S_0 can be obtained using not only Lagrange type information on f, but also some of its partial derivatives.

Let $X \subset \mathbb{R}^2$ be an arbitrary domain, $f : X \to \mathbb{R}$ and $Z = \{z_i \mid z_i = (x_i, y_i) \in X, \quad i = 1, ..., N\}$ be the set of the interpolation nodes. Consider the set of Hermite type functionals:

$$\Lambda_H(f) = \left\{\lambda_i^{p,q} \mid \lambda_i^{p,q} f = f^{(p,q)}(z_i),\ (p,q) \in \mathbb{N}^2,\ p+q \le m_i,\ i = 1, \ldots, N\right\},$$

with $m_i \in \mathbb{N}^*$, $i = 1, ..., N$.

First, denote by $\Lambda_{H,k} \subset \Lambda_H$ the functionals corresponding to the point z_k, i.e.,

$$\Lambda_{H,k}(f) := \left\{\lambda_k^{p,q} \mid \lambda_k^{p,q} f = f^{(p,q)}(z_k),\ (p,q) \in \mathbb{N}^2,\ p+q \le m_k\right\}.$$

Let $T_k^{m_k}$ be the bivariate Taylor interpolation operator corresponding to the set $\Lambda_{H,k}$:

$$(T_k^{m_k} f)(x,y) = \sum_{i+j \le m_k} \frac{(x-x_k)^i (y-y_k)^j}{i!j!} f^{(i,j)}(x_k, y_k).$$

As $\Lambda_{H,k}(f)$, $k = 1, \ldots, N$, is an admissible sequence of functionals, the operator $T_k^{m_k}$ exists, for all $k = 1, \ldots, N$.

Definition 2.3.26. *The operator* $S^T_{m_1,\ldots,m_N}$ *given by*

$$(S^T_{m_1,\ldots,m_N} f)(x,y) = \sum_{i=1}^{N} A_i(x,y)(T_i^{m_i} f)(x,y) \tag{2.3.36}$$

is called the bivariate Shepard-Taylor operator.

Remark 2.3.27. The operator S_1^T, given by

$$(S_1^T f)(x,y) = \sum_{i=1}^{N} A_i(x,y)(T_i^1 f)(x,y)$$

or

$$(S_1^T f)(x,y) = \sum_{i=1}^{n} A_i(x,y)[f(x_i,y_i)+(x-x_i)f^{(1,0)}(x_i,y_i)+(y-y_i)f^{(0,1)}(x_i,y_i)]$$

was defined and studied by Shepard himself.

For $\mu > 1$, it is easy to verify that

$$(S_1^T f)^{(p,q)}(x_i,y_i) = f^{(p,q)}(x_i,y_i), \quad p+q \le 1, \quad i = 1,\dots,N$$

and

$$\operatorname{dex}(S_1^T) = 1.$$

Theorem 2.3.28. *For $\mu > \max\{m_1,\dots,m_N\}$ we have*

$$(S^T_{m_1,\dots,m_N} f)^{(p,q)}(x_i,y_i) = f^{(p,q)}(x_i,y_i), \quad p+q \le m_i, \quad i = 1,\dots,N, \tag{2.3.37}$$

respectively,

$$\operatorname{dex}(S^T_{m_1,\dots,m_N}) = \min\{m_1,\dots,m_N\}. \tag{2.3.38}$$

Proof. First, we see that

$$\begin{cases} A_i^{(p,q)}(x_k,y_k) = 0, & k = 1,\dots,N; \quad 0 \le p+q \le m_k, \quad i \ne k \\ A_i^{(p,q)}(x_i,y_i) = 0, & p+q \ge 1, \end{cases} \tag{2.3.39}$$

for all $i = 1,\dots,N$. Indeed, let us consider

$$A_k = \frac{g_k}{h_k}$$

with

$$g_k(x,y) = \prod_{\substack{j=1 \\ j\ne k}}^{N} \rho_j^{\mu}(x,y)$$

$$h_k(x,y) = \sum_{k=1}^{N} \prod_{\substack{j=1 \\ j\ne k}}^{N} \rho_j^{\mu}(x,y).$$

After a straightforward computation, one obtains

$$g_k^{(p,q)}(x_i, y_i) = 0, \quad 1 \leq p + q \leq m_i,$$

for $\mu > \max\{m_1, \ldots, m_N\}$, $i = 1, \ldots, N$, $i \neq k$ and

$$g_k^{(p,q)}(x_k, y_k) = h_k^{(p,q)}(x_k, y_k), \quad 1 \leq p + q \leq m_k.$$

So, (2.3.39) follows. Now, relations (2.3.39) imply the interpolation property (2.3.37). The degree of exactness property (2.3.38) is a consequence of Remark 2.3.10, namely of the fact that $\operatorname{dex}(T_k^{m_k}) = m_k$, $k = 1, ..., N$. ■

Next, for $\nu_k \in \mathbb{N}^*$, $\nu_k < N$, let us consider the set

$$\begin{aligned}\Lambda_{H,\nu_k}(f) :=& \{\lambda_{k+j}^{p,q} \mid \lambda_{k+j}^{p,q} f = f^{(p,q)}(z_{k+j}), (p,q) \in \mathbb{N}^2,\ p + q \leq m_k,\\ & j = 0, 1, \ldots, \nu_k - 1\}\end{aligned}$$

with $z_{N+i} = z_i$, $i \in \mathbb{N}^*$, i.e., let $\Lambda_{H,\nu_k}(f) \subset \Lambda_H(f)$ be the set of functional corresponding to the nodes $z_k, z_{k+1}, \ldots, z_{k+\nu_k-1}$, regarding function f.

If $|\Lambda_{H,\nu_k}(f)| = m_k$ let $H_k^{n_k}$ be the Hermite interpolation operator corresponding to the set $\Lambda_{H,\nu_k}(f)$, $k = 1, \ldots, N$, with n_k such that $m_k = (n_k + 1)(n_k + 2)/2$.

Definition 2.3.29. *Suppose that $H_k^{n_k}$ exists for all $k = 1, \ldots, N$. The operator $S_{n_1,\ldots,n_N}^H$ defined by*

$$(S_{n_1,\ldots,n_N}^H f)(x, y) = \sum_{k=1}^{N} A_k(x, y)(H_k^{n_k} f)(x, y)$$

is called the bivariate Shepard-Hermite operator.

Theorem 2.3.30. *If $\Lambda_{H,\nu_k}(f) \subset \Lambda_H(f)$, $k = 1, \ldots, N$ is an admissible sequence, then $S_{n_1,\ldots,n_N}^H$ exists and it has the following interpolation properties:*

$$(S_{n_1,\ldots,n_N}^H f)^{(p,q)}(x_i, y_i) = f^{(p,q)}(x_i, y_i), \quad p + q \leq m_i, \quad i = 1, \ldots, N, \tag{2.3.40}$$

for $\mu > \max\{m_1, ..., m_N\}$ and

$$\operatorname{dex}(S_{n_1,\ldots,n_N}^H) = \min\{n_1, \ldots, n_N\} \tag{2.3.41}$$

Proof. Relations (2.3.40) follows by (2.3.39), taking into account the interpolation properties of the polynomials $H_k^{n_k} f$, $k = 1, ..., N$.

Relation (2.3.41) follows by Remark 2.2.43. ■

Particular cases

• if $\nu_k = 1$, for all $k = 1, \ldots, N$, then $S^H_{n_1,\ldots,n_N}$ becomes the Shepard-Taylor operator (2.3.36)

• for $n_1 = \cdots = n_N = n$, all polynomials $H_k^{n_k}$, $k = 1, \ldots, N$ are of degree n and one obtains the Shepard-Hermite operator S_n^H:

$$(S_n^H f)(x, y) = \sum_{k=1}^{N} A_k(x, y)(H_k^n f)(x, y). \tag{2.3.42}$$

Example 2.3.31. Let

$$\Lambda_H(f) = \left\{\lambda_i^{p,q} \mid \lambda_i^{p,q} f = f^{(p,q)}(x_i, y_i),\ (p, q) \in \mathbb{N}^2,\ p + q \leq 1,\ i = 1, \ldots, N\right\} \tag{2.3.43}$$

and

$$\Lambda_{H,\nu_k}(f) = \left\{\lambda_k^{00}, \lambda_k^{10}, \lambda_k^{01}, \lambda_{k+1}^{00}, \lambda_{k+2}^{00}, \lambda_{k+3}^{00}\right\}, \tag{2.3.44}$$

with $\nu_k = 4$ and $\lambda_{N+1} = \lambda_1$, $\lambda_{N+2} = \lambda_2$, $\lambda_{N+3} = \lambda_3$. The existence and uniqueness of the Hermite operator H_k^2 are based on the following auxiliary result.

Lemma 2.3.32. *Let $z_{k+i} = (x_{k+i}, y_{k+i})$, $i = 0, 1, 2, 3$ be some given points in plane such that:*

• *$x_{k+i} \neq x_k$, for $i = 1, 2, 3$*

• *if l_{k+i} is the line determined by the points z_k and z_{k+i}, $i = 1, 2, 3$ then $l_{k+i} \neq l_{k+j}$, for $i, j = 1, 2, 3$, $i \neq j$.*

Then for every function f with the given information $\left\{f(z_k), f^{(1,0)}(z_k), f^{(0,1)}(z_k), f(z_{k+1}), f(z_{k+2}), f(z_{k+3})\right\}$ there exists a unique polynomial P_2 of second degree such that

$$\begin{cases} P_2^{(i,j)}(z_k) = f^{(i,j)}(z_k), & i, j = 0, 1 \quad i \neq j \leq 1 \\ P_2(z_{k+i}) = f(z_{k+i}), & i = 1, 2, 3 \end{cases} \tag{2.3.45}$$

Proof. Let P_2 be an arbitrary polynomial of second degree:

$$P_2(x, y) = Ax^2 + Bxy + Cy^2 + Dx + Ey + F. \tag{2.3.46}$$

From (2.3.45) one obtains a 6×6 linear algebraic system. Let M be its matrix. Then, after some permissible transformation, one obtains

$$\det M = -\prod_{i=1}^{3}(x_{k+i} - x_i)^2 \begin{vmatrix} 1 & \alpha_{k+1} & \alpha_{k+1}^2 \\ 1 & \alpha_{k+2} & \alpha_{k+2}^2 \\ 1 & \alpha_{k+3} & \alpha_{k+3}^2 \end{vmatrix},$$

where

$$\alpha_{k+i} = \frac{y_{k+i} - y_k}{x_{k+i} - x_k}, \quad i = 1, 2, 3. \tag{2.3.47}$$

The hypothesis from Lemma (2.3.32) implies that $\det M \neq 0$, i.e., the system has a unique solution. ■

Theorem 2.3.33. *If Λ_{H,ν_k}, $k = 1, \ldots, N$, given by (2.3.44), is an admissible sequence, then there exists the Shepard-Hermite operator S_2^H, defined by*

$$(S_2^H f)(x, y) = \sum_{k=1}^{N} A_k(x, y)(H_k^2 f)(x, y),$$

where H_k^2 is the interpolation operator corresponding to the set Λ_{H,ν_k}.

Proof. The proof follows from Lemma 2.3.32. In order to give effectively the function $S_2^H f$, we have to determine the Hermite polynomials $H_k^2 f$, $k = 1, \ldots, N$. For this, we consider $H_k^2 f$ in the form:

$$\begin{aligned} H_k^2 f =& h_{kk}^{00} f(x_k, y_k) + h_{kk}^{10} f^{(1,0)}(x_k, y_k) + h_{kk}^{01} f^{(0,1)}(x_k, y_k) \\ &+ h_{kk}^{00} f(x_{k+1,k+1}) + h_{kk}^{00} f(x_{k+2,k+2}) + h_{kk}^{00} f(x_{k+3,k+3}), \end{aligned}$$

where $h_{i,j}^{p,q}$ are the corresponding fundamental interpolation polynomials. Hence, each polynomial $h_{i,j}^{p,q}$ is a bivariate polynomial of the second degree, as in (2.3.46), that satisfies the cardinal interpolation conditions. For example,

$$\begin{cases} h_{kk}^{00}(x_k, y_k) = 1 \\ h_{kk}^{10}(x_k, y_k) = 0 \\ h_{kk}^{01}(x_k, y_k) = 0 \\ h_{kk}^{00}(x_{k+1}, y_{k+1}) = 0 \\ h_{kk}^{00}(x_{k+2}, y_{k+2}) = 0 \\ h_{kk}^{00}(x_{k+3}, y_{k+3}) = 0 \end{cases}$$

which is a linear algebraic 6×6 system. ■

Example 2.3.34. Let

$$\Lambda_H(f) = \left\{ \lambda_k^{0,0}, \lambda_k^{1,0}, \lambda_k^{0,1}, \lambda_k^{1,1} \,|k = 1, \ldots, N \right\}. \tag{2.3.48}$$

One considers the sequence of subsets of Λ_H:

$$\Lambda_{H,\nu_k}(f) = \left\{ \lambda_k^{0,0}, \lambda_k^{1,0}, \lambda_k^{0,1}, \lambda_k^{1,1}, \lambda_{k+1}^{0,0}, \lambda_{k+2}^{0,0} \right\}, \quad \nu_k = 3, \quad k = 1, \ldots, N, \tag{2.3.49}$$

with

$$\lambda_{N+1} = \lambda_1, \qquad \lambda_{N+2} = \lambda_2.$$

Lemma 2.3.35. *If*

- $x_{k+1} \neq x_k$ *and* $x_{k+2} \neq x_k$,
- $\alpha_{k+1} \neq \alpha_{k+2}$ *and* $\alpha_{k+1} \neq -\alpha_{k+2}$, *for* α_{k+1}*and* α_{k+2} *given in (2.3.47) then (2.3.49) is an admissible sequence.*

Proof. For P_2 as in (2.3.46) we have to study the linear system:

$$\begin{cases} P_2^{(i,j)}(x_k, y_k) = f^{(i,j)}(x_k, y_k), \\ P_2(x_{k+1}, y_{k+1}) = f(x_{k+1}, y_{k+1}), \\ P_2(x_{k+2}, y_{k+2}) = f(x_{k+2}, y_{k+2}), \end{cases}$$

with $(i, j) \in \{(0, 0), (0, 1), (1, 0), (1, 1)\}$. The determinant of this system is:

$$D = (x_{k+1} - x_k)^2 (x_{k+2} - x_k)^2 (\alpha_{k+2}^2 - \alpha_{k+1}^2)$$

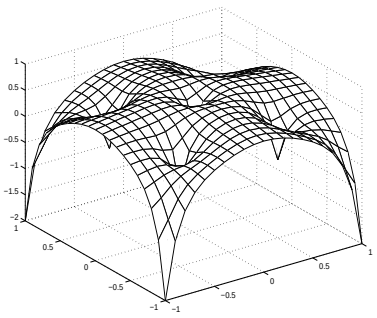
and the proof follows. ■

Remark 2.3.36. From Lemma 2.3.35 it follows that the Shepard-Hermite operator, corresponding to the set of functionals (2.3.48), is given by

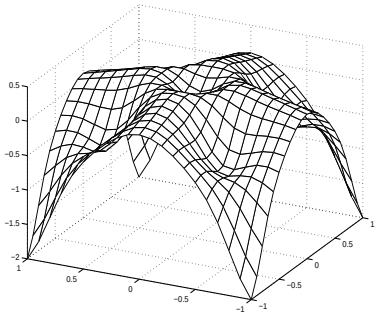
$$S_2^H f = \sum_{i=0}^{n} A_i H_k^2 f,$$

where H_k^2 is the Hermite interpolation operator regarding Λ_{H,ν_k}. This operator exists and is unique.

Example 2.3.37. For f the same as in Example 2.3.13 and the same nodes, in Figure 2.4 we plot $S_1^T f$, for $\mu = 1; 2$.



(a) Interpolant $S_1^T f$, $\mu = 1$.



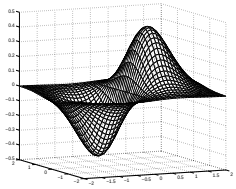
(b) Interpolant $S_1^T f$, $\mu = 2$.

Figure 2.4: Bivariate Shepard-Taylor interpolants.

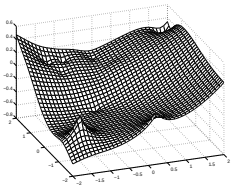
Example 2.3.38. Let $g : [-2, 2] \times [-2, 2] \to \mathbb{R}$,

$$g(x, y) = xe^{-(x^2+y^2)},$$

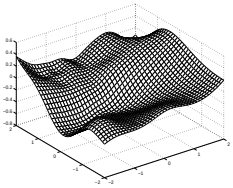
and consider 10 random nodes, uniformly distributed on the domain of g. In Figure 2.5 we plot $S_2^H g$, for $\mu = 1; 2$.



(a) Function g.



(b) $S_2^H g$, $\mu = 1$.



(c) $S_2^H g$, $\mu = 2$.

Figure 2.5: Bivariate Shepard-Hermite interpolants.

2.3.3.3 The bivariate Shepard-Birkhoff operator

Let $X \subset \mathbb{R}^2$ be an arbitrary domain, $f : X \to \mathbb{R}$ and $Z = \{z_i \mid z_i = (x_i, y_i) \in X, \quad i = 1, ..., N\}$ be the set of the interpolation nodes. Consider the set of Birkhoff type functionals:

$$\Lambda_B(f) = \left\{ \lambda_k^{p,q} | \lambda_k^{p,q} f = f^{(p,q)}(z_k), \quad (p, q) \in I_k \subset \mathbb{N}^2, \ k = 1, \ldots, N \right\}.$$

and the sequence Λ_{k,ν_k}, $k = 1, \ldots, N$ of the subsets of Λ_B:

$$\Lambda_{B,\nu_k}(f) = \left\{ \lambda_{k+j}^{p,q} \in \Lambda_B(f) \,|(p, q) \in I_{k+j}, \ j = 0, 1, \ldots, \nu_k - 1 \right\}, \quad (2.3.50)$$

with $\nu_k < N$, $k = 1, \ldots, N$ and $z_{N+i} = z_i$, $i \geq 1$.

Denote by $B_k^{n_k}$ the interpolation operator of the total degree n_k corresponding to the subset Λ_{B,ν_k}, i.e.,

$$\lambda_{k+j}^{p,q}(B_k^{n_k}) = \lambda_{k+j}^{p,q}(f), \quad (p,q) \in I_{k+j},\ j = 0, 1, \ldots, \nu_k - 1. \tag{2.3.51}$$

Taking into account that

$$(B_k^{n_k})(x,y) = \sum_{i+j \leq n_k} a_{ij} x^i y^j,$$

the interpolation condition (2.3.51) gives rise to a linear algebraic system in the unknowns a_{ij}, $i, j \in \mathbb{N}$, $i + j \leq n_k$. Let $M_k := M(\Lambda_{B,\nu_k})$ be the matrix of this system. If $|\Lambda_{B,\nu_k}| = m_k := (n_k+1)(n_k+2)/2$ then M_k is a square matrix. Hence, for $\det M_k \neq 0$ the system has a unique solution. It follows that, if $\det M_k \neq 0$, for all $k = 1, \ldots, N$, then (2.3.50) is an admissible sequence.

Definition 2.3.39. *Suppose that* $\det M_k \neq 0$, $k = 1, \ldots, N$. *The operator* $S^B f$ *defined by*

$$(S^B_{n_1,\ldots,n_N} f)(x,y) = \sum_{k=1}^{N} A_k(x,y)(B_k^{n_k} f)(x,y) \tag{2.3.52}$$

is called the bivariate Shepard-Birkhoff operator.

Theorem 2.3.40. *If* Λ_{B,ν_k}, $k = 1, \ldots, N$ *is an admissible sequence then, for* $\mu > M := \max\limits_{1 \leq k \leq N} \{\mu_k \mid \mu_k = \max\{p+q \mid (p,q) \in I_k\}\}$ *we have*

$$(S^B_{n_1,\ldots,n_N} f)^{(p,q)}(x_k, y_k) = f^{(p,q)}(x_k, y_k),\ (p,q) \in I_k,\ k = 1, \ldots, N \tag{2.3.53}$$

and

$$\operatorname{dex}(S^B) = m := \min\{n_k \mid k = 1, \ldots, N\}. \tag{2.3.54}$$

Proof. For $\mu > M$, the relation (2.3.39) implies:

$$\begin{aligned}(S^B_{n_1,\ldots,n_N} f)^{(p,q)}(x_i, y_i) &= \sum_{k=1}^{N} (A_k B_k^{n_k} f)^{(p,q)}(x_i, y_i) \\ &= \sum_{k=1}^{N} A_k(x_i, y_i)(B_k^{n_k} f)^{(p,q)}(x_i, y_i)\end{aligned}$$

Now, from (2.3.23) and (2.3.51) the interpolation property (2.3.53) follows.

Relation (2.3.54) follows by Remark 2.2.43. ■

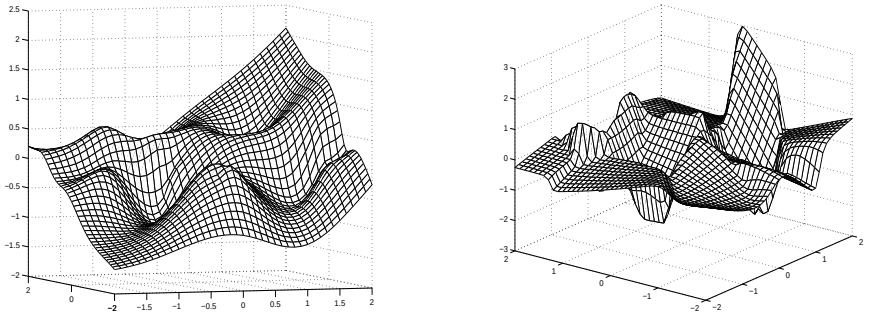
Remark 2.3.41. A particular case of the operator $S^B_{n_1,\dots,n_N}$ is obtained for $n_k = n$, $k = 1, \dots, N$, i.e., when all the polynomials $B^{n_k}_k f$ have the same degree n. One obtains

$$(S^B_n f)(x,y) = \sum_{k=1}^{N} A_k(x,y)(B^n_k f)(x,y).$$

Remark 2.3.42. The Shepard operator of Birkhoff type S^B_n is an extension of all previous Shepard operators of Lagrange, Taylor and Hermite type.

The difficult problem which appears in the Hermite and Birkhoff cases is to prove the admissibility of the considered sequence of the subsets of Λ_B. Theorem 2.3.15 gives sufficient condition for the admissibility of such sequences in the Shepard operators of Lagrange type. Also, the problem of sequence admissibility is completely solved for the Shepard operators of Taylor type.

Example 2.3.43. For g the same as in Example 2.3.38 and 10 random nodes, uniformly distributed on the domain of g, in Figure 2.6 we plot $S^B_2 g$, for $\mu = 2; 20$.



(a) Interpolant $S^B_2 g$, $\mu = 2$. (b) Interpolant $S^B_2 g$, $\mu = 20$.

Figure 2.6: Bivariate Shepard-Birkhoff interpolants.

2.4 Uniform approximation

Let $f : [a,b] \to \mathbb{R}$ be an integrable function, $\lambda_k(f)$, $k = 0, 1, ...m$ some information about f, $w : [a,b] \to \mathbb{R}_+$ a weight function, also integrable on

$[a,b]$, and

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

2.4.1 The Weierstrass theorems

Let $\mathcal{B}$ be a given class of real-valued functions defined on an interval $[a,b]\subset\mathbb{R}$ and $\mathcal{A}\subset\mathcal{B}$. One considers the following problem: for some given $f\in\mathcal{B}$ and $\varepsilon\in\mathbb{R}_+$ find a function $g\in\mathcal{A}$ such that

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

The solution of this problem is based on the Weierstrass theorems, usually called "First Weierstrass Theorem", respectively, "Second Weierstrass Theorem".

Theorem 2.4.1. (First Weierstrass Theorem) *For any $f\in C[a,b]$ and $\varepsilon>0$ there exists an algebraic polynomial P ($P\in\mathbb{P}$) such that*

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

In other words, the set $\mathbb{P}$ ($\mathbb{P}\subset C[a,b]$) is dense in the Banach space $C[a,b]$.

Next we give some other equivalent formulations of this theorem.

Theorem 2.4.2. *Any function $f\in C[a,b]$ is the limit of a sequence of algebraic polynomials, uniform convergent to f, on $[a,b]$.*

Theorem 2.4.3. *For any function $f\in C[a,b]$ a series of algebraic polynomials absolute and uniform convergent to f, on $[a,b]$, can be found.*

Theorem 2.4.4. (Second Weierstrass Theorem) *The set $\mathcal{T}$ of all trigonometric polynomials is dense in $C(T)$, where $C(T)$ denotes the class of continuous functions of period 2π.*

Remark 2.4.5. As Theorem 2.4.4 can be obtained as a consequence of Theorem 2.4.1, we shall focus on the First Weierstrass Theorem and present some of the many existing proofs.

A proof of Weierstrass theorem based on the probability theory. In what follows, we present the well-known proof given by S. Bernstein in 1912, exactly in the way it was given. Because he used probabilistic tools, the interval $I=[0,1]$ is considered. Under these circumstances, we give another enounce of the Weierstrass theorem:

Theorem 2.4.6. *If $F(x)$ is a continuous function everywhere in the interval $[0,1]$, it is always possible to determine a polynomial of degree n,*

$$E_n(x) = a_0x^n + a_1x^{n-1} + \cdots + a_n,$$

with n large enough, such that

$$|F(x) - E_n(x)| < \varepsilon$$

at every point of $[0,1]$, for any ε, no matter how small it is.

Proof. We consider an event A of probability x. Suppose that we make n experiences and we pay to a gambler the amount of money $F\left(\frac{m}{n}\right)$ if the event A appears m times. Under this circumstances, the mean value, E_n, of the gambler will have the following expression:

$$E_n = \sum_{m=0}^{n} F\left(\tfrac{m}{n}\right) C_n^m x^m (1-x)^{n-m}. \tag{2.4.1}$$

From the continuity of function $F(x)$, it is possible to fix a number, δ, such as the inequality

$$|x - x_0| \leq \delta$$

generates

$$|F(x) - F(x_0)| < \tfrac{\varepsilon}{2}.$$

Denoting with $\overline{F}(x)$ the maximum and $\underline{F}(x)$ the minimum of $F(x)$ in $(x-\delta, x+\delta)$, we get

$$\overline{F}(x) - F(x) < \tfrac{\varepsilon}{2}, \quad F(x) - \underline{F}(x) < \tfrac{\varepsilon}{2}. \tag{2.4.2}$$

Let η be the probability of inequality $\left|x - \dfrac{m}{n}\right| > \delta$ and L the maximum of $|F(x)|$ in $[0,1]$. Then, we have:

$$\underline{F}(x)(1-\eta) - L\eta < E_n < \overline{F}(x)(1-\eta) + L\eta. \tag{2.4.3}$$

But, according with Bernoulli's Theorem, we may take n so big in order to get

$$\eta < \tfrac{\varepsilon}{4L}. \tag{2.4.4}$$

Then, the inequality (2.4.3) can be written as follows:

$$F(x)+(\underline{F}(x)-F(x))-\eta(L+\underline{F}(x)) < E_n < F(x)+(\overline{F}(x)-F(x))+\eta(L-\overline{F}(x))$$

and then

$$F(x) - \tfrac{\varepsilon}{2} - \tfrac{2L}{4L}\varepsilon < E_n < F(x) + \tfrac{\varepsilon}{2} + \tfrac{2L}{4L}\varepsilon,$$

and so

$$|F(x) - E_n| < \varepsilon. \tag{2.4.5}$$

Or, E_n is a polynomial of degree n. So, the theorem is proved. ■

Remark on the Bernstein polynomial. The Bernstein polynomials can be written in the more general form

$$b(x) = \sum_{k=0}^{n} \beta_k \tbinom{n}{k} x^k (1-x)^{n-k}, \tag{2.4.6}$$

where $\beta_0, \ldots, \beta_n$ are given coefficients. The case $\beta_k = 1$, $k = \overline{0,n}$ generates the relation

$$\sum_{k=0}^{n} \tbinom{n}{k} x^k (1-x)^{n-k} = (x + (1-x))^n = 1,$$

fact which is already well-known, being the sum of the probabilities in the distribution of the random variable introduced by Bernstein (even not very clearly emphasized!). What is more interesting, is the fact that, if $b = 0$ (identically equal), all coefficients β_n are zero.

By induction, the base case $n = 0$ is obvious. When $n > 0$, we see that the derivative of a Bernstein polynomial can be written as another Bernstein polynomial:

$$b'(t) = n \sum_{k=0}^{n-1} (\beta_{k+1} - \beta_k) \tbinom{n-1}{k} x^k (1-x)^{n-1-k}. \tag{2.4.7}$$

In particular, if $b = 0$, it follows by the induction hypothesis that all β_k are equal, and then they are all zero, by (2.4.2).

In other words, the polynomials $x^k(1-x)^{n-k}$, $k = 0, \ldots, n$ are linearly independent, and hence they span the $n+1$ dimensional space of polynomials of degree ≤ 1. Thus, *all polynomials can be written as Bernstein polynomials.*

A proof of Weierstrass theorem using beta functions. Let us define, for $0 < \alpha < 1$,

$$U_\alpha(x) = x^\alpha (1-x)^{1-\alpha}.$$

These functions are the well known Beta functions. They are increasing on the interval $[0, \alpha]$ and decreasing on the interval $[\alpha, 1]$.

Lemma 2.4.7. *Let $0 \leq a < b \leq \alpha$ or $\alpha \leq b < a \leq 1$. Then*

$$\frac{U_\alpha(a)}{U_\alpha(b)} \leq e^{-2(a-b)^2}.$$

Proof. Let us assume that $0 \leq a < b \leq \alpha$. It will be done if we prove that the function

$$H(x) = \ln(U_\alpha(x)) - \ln(U_\alpha(b)) + 2(x-b)^2$$

is negative for $0 \leq x < b$.

A simple computation of the derivative of H gives us

$$H'(x) = \tfrac{\alpha - x}{x(1-x)} + 4(x-b) \geq 4(\alpha - x) + 4(x-b) \geq 0$$

because $x(1-x) \leq \dfrac{1}{4}$. This ends the proof, since $F(b) = 0$.

The case $\alpha \leq b < a \leq 1$ can be proved in a similar way. ■

Lemma 2.4.8. *Let $\varepsilon > 0$, $\mu > 0$ and $0 \leq p \leq n$. There exists N such that for $n > N$*

$$\sum_{|k-p|>2\mu n} \tbinom{n}{k} x^k (1-x)^{n-k} < \varepsilon,$$

for

$$x \in \left(\tfrac{p}{n} - \mu, \tfrac{p}{n} + \mu\right).$$

Proof. Let A be a set of all $0 \leq k \leq n$ such that $|k-p| \geq 2\mu n$. Let B be a part of A of elements less than p, let C be a part of A of elements greater than p.

Define

$$s = \tfrac{p - \mu n}{n}, \quad r = \tfrac{p - 2\mu n}{n}.$$

Notice that

$$x^k(1-x)^{n-k} = (U_\alpha(x))^n, \text{ where } \alpha = \tfrac{k}{n}.$$

For $k \in B$ and $x \in \left(\dfrac{p}{n} - \mu, \dfrac{p}{n} + \mu\right)$ we have $\alpha \leq r \leq r + \eta \leq s < x$ so that Lemma 2.4.7 can be applied:

$$\begin{aligned} x^k(1-x)^{n-k} = (U_\alpha(x))^n &\leq (U_\alpha(s))^n \leq \left(e^{-2\mu^2} U_\alpha(r)\right)^n \\ &= e^{-2\mu^2 n} r^k (1-r)^{n-k} \leq \tfrac{\varepsilon}{2} r^k (1-r)^{n-k} \end{aligned}$$

if

$$n > \frac{\log\left(\frac{2}{\varepsilon}\right)}{2\mu^2}.$$

Finally,

$$\begin{aligned}\sum_{k\in B}\tbinom{n}{k}x^k(1-x)^{n-k} &\le \sum_{k\in B}\tbinom{n}{k}\tfrac{\varepsilon}{2}r^k(1-r)^{n-k}\\ &\le \tfrac{\varepsilon}{2}\sum_{k=0}^{n}\tbinom{n}{k}r^k(1-r)^{n-k} = \tfrac{\varepsilon}{2}.\end{aligned}$$

A similar inequality can be derived with a set C instead of B. Adding both inequalities, we receive the assertion of Lemma 2.4.8. ■

Remark 2.4.9. If the summation of the statement of Lemma 2.4.8 extends over all $0 \le k \le n$, then the result of this summation is 1.

A proof of Weierstrass theorem using the concept of convolution.

Definition 2.4.10. *If f and g are some suitable functions on $\mathbb{R}$, then* the convolution $f * g$ *is the function defined by*

$$f * g(x) = \int_{-\infty}^{\infty} f(x-t)g(t)dt,$$

where "suitable" means that the integral exists.

Remark 2.4.11. In what follows, we need the Delta-Dirac function, δ, which has the properties:

$$\delta(x) = 0 \text{ if } x \neq 0 \text{ and } \int_{-\infty}^{\infty}\delta(t)dt = 1.$$

We should think of it as "The density function of a unit mass at the origin".

For example,

$$f * \delta(x) = \int_{-\infty}^{\infty} f(x-t)\delta(t)dt = f(x),$$

since $\delta(t) = 0$ except at $t = 0$.

Remark 2.4.12. Having in mind the previous remark, what we have to do now is to find a sequence of functions (K_n) which approximate the δ-function. The sequence $(K_n * f)$ will then approximate $\delta * f = f$.

Definition 2.4.13. *The n-th* ***Landau kernel function*** *is*

$$K_n = c_n(1-x^2)^n \text{ for } x \in [-1,1] \text{ and } 0 \text{ otherwise,}$$

where c_n is chosen so that

$$\int_{-\infty}^{\infty} K_n = 1.$$

Lemma 2.4.14. *If f is a continuous function on the interval $[-1,1]$, then $K_n * f$ is a polynomial.*

Proof. We have

$$K_n * f(x) = \int_{-1}^{1} K_n(x-t)f(t)dt$$

and K_n is a polynomial, and so $K_n(x-t)$ can be expanded as

$$g_0(t) + g_1(t)x + \cdots + g_{2n}(t)x^{2n}$$

and so the integral is a polynomial in x. ■

Lemma 2.4.15. *The sequence*

$$(K_n * f) \to f \text{ for } n \to \infty.$$

Proof. We need to show that K_n has "most of its area" concentrated near $x = 0$.

First we estimate how big c_n is:

$$\int_{-1}^{1}(1-t)^{2n}dt = 2\int_{0}^{1}(1-t)^n(1+t)^n dt \geq 2\int_{-1}^{1}(1-t)^n dt = \tfrac{2}{n+1}.$$

Since $\int_{-\infty}^{\infty} K_n = 1$ we must have $c_n \leq \dfrac{n+1}{2}$.

(In fact, c_n grows like a multiple of $\sqrt{n}$. For large n, c_n is approximately $0,565\sqrt{n}$.)

Concerning the area under K_n which is not near 0 we have

$$\int_\delta^1 K_n(t)dt = \int_\delta^1 c_n(1-t^2)^n dt \leq \tfrac{n+1}{2}\int_\delta^1 (1-\delta^2)^n dt,$$

since K_n is decreasing on $[\delta, 1]$ and this is

$$\tfrac{n+1}{2}(1-\delta^2)^n(1-\delta).$$

If $r = 1-\delta^2$ then $(n+1)r^n \to 0$ as $n \to \infty$.

The function f is continuous and bounded by M.

If $x \in [0,1]$ then given $\varepsilon > 0$ we may find $\delta > 0$ such that if $|t| < \delta$ then $|f(x-t) - f(x)| < \varepsilon$. So now look at the convolution $K_n * f$:

$$|f(x) - K_n * f(x)| = \int_{-1}^1 \underbrace{|f(x) - f(x-t)|K_n(t)dt}_{A} = \int_{-1}^{-\delta} A + \int_{-\delta}^{\delta} A + \int_\delta^1 A.$$

Now, on $[-1, -\delta]$ and on $[\delta, 1]$, we have that $K_n(t)$ is small if we choose δ small. In fact, we can choose δ so that $K_n(t) < \frac{\varepsilon}{M}$ and then the first and the third integral are less than ε.

For the middle integral,

$$\int_{-\delta}^{\delta} |f(x) - f(x-t)|K_n(t)dt \leq \int_{-\delta}^{\delta} \varepsilon K_n(t)dt < \varepsilon,$$

since

$$\int_{-1}^1 K_n(t)dt = 1.$$

Thus, $|f(x) - K_n * f(x)|$ is small when n is large and we have our convergence. This complete the proof of Weierstrass approximation theorem. ■

2.4.2 The Stone-Weierstrass theorem

The generalized approach of the Weierstrass approximation theorem is known as **the Stone-Weierstrass theorem**.

Theorem 2.4.16. *Let X be a compact metric space and let $C^0(X,\mathbb{R})$ be the algebra of continuous real functions defined over X. Let $\mathcal{A}$ be a subalgebra of $C^0(X,\mathbb{R})$ for which the following conditions hold:*

(1) $\forall x, y \in X,\ x \neq y,\ \exists f \in \mathcal{A} : f(x) \neq f(y)$,
(2) $1 \in \mathcal{A}$.
Then $\mathcal{A}$ *is dense in* $C^0(X,\mathbb{R})$.

Proof. Let $\overline{\mathcal{A}}$ denote the closure of $\mathcal{A}$ in $C^0(X,\mathbb{R})$, according with the uniform convergence topology. We want to show that, if conditions 1 and 2 are satisfied, then $\overline{\mathcal{A}} = C^0(X,\mathbb{R})$.

1. Firstly, we show that, if $f \in \overline{\mathcal{A}}$, then $|f| \in \overline{\mathcal{A}}$.

Since f is a continuous function on a compact space, f must be bounded: there exists constants a and b such that $a \leq f \leq b$. By the Weierstrass approximation theorem, for every $\varepsilon > 0$, there exists a polynomial such that $\Big|P(x) - |x|\Big| < \varepsilon$ when $x \in [a, b]$.

Define $g : X \to \mathbb{R}$ by $g(x) = P(f(x))$. Since $\overline{\mathcal{A}}$ is an algebra, $g \in \overline{\mathcal{A}}$. For all $x \in X$, $\Big|g(x) - |f(x)|\Big| < \varepsilon$. Since $\overline{\mathcal{A}}$ is closed under the uniform convergence topology, this implies that $|f| \in \overline{\mathcal{A}}$.

A corollary of the fact just proven is that if $f, g \in \overline{\mathcal{A}}$, then $\max(f, g) \in \overline{\mathcal{A}}$ and $\min(f, g) \in \overline{\mathcal{A}}$, because one can write

$$\max(a, b) = \frac{1}{2}(|a + b| + |a - b|),$$

$$\min(a, b) = \frac{1}{2}(|a + b| - |a - b|).$$

2. Secondly, we shall show that, for every $f \in C^0(X,\mathbb{R})$, $x \in X$ and $\varepsilon > 0$, there exists $g_x \in \overline{\mathcal{A}}$ such that $g_x \leq f + \varepsilon$ and $g(x) > f(x)$. By condition 1, if $y \neq x$, there exists a function $\bar{h}_{xy} \in \mathcal{A}$ such that $\bar{h}_{xy}(x) \neq \bar{h}_{xy}(y)$. Define h_{xy} by

$$h_{xy}(z) = p\bar{h}_{xy}(z) + q,$$

where the constants p and q are chosen so that

$$h_{xy}(x) = f(x) + \varepsilon/2,$$
$$h_{xy}(y) = f(y) - \varepsilon/2.$$

By condition 2, $h_{xy} \in \mathcal{A}$.

For every $y \neq x$, define the set

$$U_{xy} = \{z \in X \mid h_{xy}(z) < f(z) + \varepsilon\}.$$

Since f and h_{xy} are continuous, U_{xy} is an open set. Because $x \in U_{xy}$ and $y \in U_{xy}$, $\{U_{xy} \mid y \in X \setminus \{x\}\}$ is an open cover of X. By the definition of a compact space, there must exist a finite subcover. In other words, there exists a finite subset $\{y_1, y_2, \ldots, y_n\} \subset X$ such that

$$X = \bigcup_{m=0}^{n} U_{xy_m}.$$

Define $g_x = \min\{h_{xy_1}, \ldots, h_{xy_n}\}$. By the corollary of the first part of the proof, $g_x \in \overline{\mathcal{A}}$. By construction,

$$g_x(x) = f(x) + \varepsilon/2 \text{ and } g < f + \varepsilon.$$

3. Thirdly, we shall show that, for every $f \in C^0(X, \mathbb{R})$ and every $\varepsilon > 0$, there exists a function $g \in \overline{\mathcal{A}}$ such that $f \leq g < f + \varepsilon$. This will complete the proof because it implies that $\overline{\mathcal{A}} = C^0(X, \mathbb{R})$. For every $x \in X$, define the set V_x as

$$V_x = \{z \in X \mid g_x(z) > f(x)\},$$

where g_x is defined as before. Since f and g_x are continuous, V_x is an open set. Because $g_x(x) = f(x) + \varepsilon/2 > f(x)$, $x \in V_x$. Hence $\{V_x \mid x \in X\}$ is an open cover of X. By the definition of a compact space, there must exist a finite subcover. In other words, there exists a finite subset $\{x_1, \ldots, x_n\} \subset X$ such that

$$X = \bigcup_{m=0}^{n} V_{x_m}.$$

Define g as

$$g(z) = \max\{g_{x_1}(z), \ldots, g_{x_n}(z)\}.$$

By the corollary of the first part of the proof, $g \in \overline{\mathcal{A}}$. By construction, $g > f$. Since

$$g_x < f + \varepsilon \text{ for every } x \in X, \quad g < f + \varepsilon.$$

So, the Stone-Weierstrass theorem is proved. ■

2.4.3 Positive linear operators

Another method that can be used to prove density is based on what is called the Popoviciu-Bohman-Korovkin Theorem. A primitive form of this theorem

was proved by Bohman, in 1952. His proof, and the main idea of his approach, was a generalization of Bernstein's proof of Weierstrass theorem.

One year later, Korovkin proved the same theorem for integral type operators. Korovkin's original proof is in fact based on positive singular integrals and there are very obvious links to Lebesgue's work on singular operator that, in turn, was motivated by various of the proofs of Weierstrass theorem. In 1950, Popoviciu gave also a proof of Weierstrass' theorem using interpolation polynomials.

2.4.3.1 Popoviciu-Bohman-Korovkin Theorem

Let (L_n) be a sequence of positive linear operators mapping $C[a,b]$ into itself. Assume that

$$\lim_{n\to\infty} L_n(x^i) = x^i, \quad i = 0,1,2$$

and the convergence is uniform on $[a,b]$. Then

$$\lim_{n\to\infty} (L_n f)(x) = f(x),$$

uniformly on $[a,b]$, for every $f \in C[a,b]$.

Remark 2.4.17. A similar result holds in the periodic case $\widetilde{C}[0,2\pi]$, where "test functions" are 1, $\sin x$ and $\cos x$.

How can the Popoviciu-Bohman-Korovkin Theorem be applied to obtain density results? In theory it can be easily applied. If $U_n = span\{u_1,\dots,u_n\}$, $n = 1,2,\dots$ is a nested sequence of finite-dimensional subspaces of $C[a,b]$, and L_n is a positive linear operator mapping $C[a,b]$ into U_n that satisfies the condition of the above theorem, then the $(u_k)_{k=1}^{\infty}$ span a dense subset of $C[a,b]$. In practice it is all too rarely applied in this manner.

Remark 2.4.18. The importance of Popoviciu-Bohman-Korovkin Theorem is primarily in that it presents conditions implying convergence, and also in that it provides calculable error bounds on the rate of approximation.

Remark 2.4.19. One immediate application of the theorem is a proof of the convergence of the Bernstein polynomials $B_n(f)$ to f for each $f \in C[0,1]$. We may consider the (B_m) as a sequence of positive linear operators mapping $C[0,1]$ into $\mathbb{P}_n$, the space of algebraic polynomials of degree at most n. It is

verified that

$$\begin{aligned} B_n(1;x) &= 1, \\ B_n(x;x) &= x, \\ B_n(x^2;x) &= x^2 + \frac{x(1-x)}{n} \text{ for all } n \geq 2. \end{aligned}$$

Thus, by Popoviciu-Bohman-Korovkin Theorem it follows that $B_n f$ converge uniformly to f on $[0,1]$.

2.4.3.2 Modulus of continuity

The modulus of continuity is one of the basic characteristics of continuous functions. For a continuous function f on a closed interval, it is defined as

$$\omega(f,\delta) = \max_{|h|\leq\delta} \max_{x} |f(x+h) - f(x)|, \qquad \text{for } \delta \geq 0.$$

This definition was introduced by N. Lebesgue in 1910, although in essence the concept was known earlier. If the modulus of continuity of a function f satisfies the condition

$$\omega(f,\delta) \leq M\delta^{\alpha},$$

where $0 < \alpha < 1$, then f is said to satisfy a Lipschitz condition of order α.

Remark 2.4.20. The modulus of continuity can be defined also in the following manner:

$$\omega(f,\delta) = \sup\{|f(u) - f(v)|, \qquad u,v \in I,\ |u-v| \leq \delta\},$$

for $f \in C(I)$, when I is an interval.

Remark 2.4.21. In what follows, we show some applications of the modulus of continuity in the study of the rapidity of convergence for a sequence of linear operators, L_n, to a function f, according to the Popoviciu-Bohman-Korovkin Theorem.

2.4.3.3 Popoviciu-Bohman-Korovkin Theorem in a quantitative form

We shall estimate the rapidity of convergence of $L_n(f)$ to f in terms of the rapidity of convergence of $L_n(1)$ to 1, $L_n(x)$ to x and $L_n(x^2)$ to x^2.

Theorem 2.4.22. *Let $-\infty < a < b < \infty$, and let $L_1, L_2, \dots$ be linear positive operators, all having the same domain D which contains the restrictions of $1, t, t^2$ to $[a, b]$. For $n = 1, 2, \dots$ suppose $L_n(1)$ is bounded. Let $f \in D$ be continuous in $[a, b]$, with modulus of continuity ω. Then, for $n = 1, 2, \dots$*

$$\|f - L_n(f)\| \le \|f\| \cdot \|L(1) - 1\| + \|L_n(1) + 1\|\omega(\mu_n), \tag{2.4.8}$$

where

$$\mu_n = \|(L_n([t - x]^2))(x)\|^{\frac{1}{2}}, \tag{2.4.9}$$

and $\|\cdot\|$ stands for the sup norm over $[a, b]$.

In particular, if $L_n(1) = 1$, as is often the case, (2.4.8) reduces to

$$\|f - L_n(f)\| \le 2\omega(\mu_n). \tag{2.4.10}$$

Remark 2.4.23. In forming $L_n([t - x]^2)$, the x is held fixed, while t forms the functions t, t^2 on which L_n operates.

Remark 2.4.24. Observe that (2.4.9) implies, for $n = 1, 2, \dots$

$$\mu_n^2 \le \|L_n(t^2)(x) - x^2\| + 2c\|L_n(t)(x) - x\| + c^2\|L_n(1) - 1\|,$$

where $c = \max(|a|, |b|)$. Hence, if $L_n(t^k)(x)$ converges uniformly to x^k in $[a, b]$, for $k = 0, 1, 2$, then $\mu_n \to 0$ and we have a simple estimate of μ_n in terms of $\|L_n(t^k)(x) - x^k\|$, $k = 0, 1, 2$.

Proof. Let $x \in [a, b]$ and let δ be a positive number. Let $t \in [a, b]$. If $|t - x| > \delta$ then

$$\begin{aligned} |f(t) - f(x)| &\le \omega(|t - x|) = \omega(|t - x|\delta^{-1}\delta) \\ &\le (1 + |t - x|\delta^{-1})\omega(\delta) \le [1 + (t - x)^2\delta^{-2}]\omega(\delta). \end{aligned}$$

The inequality

$$|f(t) - f(x)| \le [1 + (t - x)^2\delta^{-2}]\omega(\delta)$$

holds also if $|t - x| \le \delta$. Let n be a positive integer. Then

$$\begin{aligned} |(L_n f - f(x)L_n(1))(x)| &\le \omega(\delta)[(L_n(1) + \delta^{-2}L_n([t - x]^2))(x)] \\ &\le \omega(\delta)[L_n(1)(x) + (\mu_n/\delta)^2]. \end{aligned}$$

If $\mu_n > 0$, take $\delta = \mu_n$. Then

$$\begin{aligned} &|[L_n(f) - f(x)L_n(1)](x)| \le \omega(\mu_n)\|L_n(1) + 1\|, \\ &|-f(x) + f(x)L_n(1)(x)| \le \|f\| \cdot \|L_n(1) - 1\|. \end{aligned} \tag{2.4.11}$$

Adding, we obtain (2.4.8). If $\mu_n = 0$, we have for every positive δ,

$$|(L_n(f) - f(x)L_n(1))(x)| \le \omega(\delta)L_n(1)(x).$$

Making $\delta \to 0{+}0$, we obtain $(L_n f)(x) = f(x)L_n(1)(x)$. Then, by (2.4.11),

$$|[f - L_n(f)](x)| \le \|f\| \cdot \|L_n(1) - 1\|,$$

which implies (2.4.8). ■

Example 2.4.25. Let D be the set of all real functions with domain $[0, 1]$. For $n = 1, 2, \dots$ let L_n be the linear positive operator defined by

$$(L_n\phi)(x) = \sum_{k=0}^{n} \binom{n}{k} x^k (1-x)^{n-k} \phi\left(\frac{k}{n}\right).$$

Let f be a real function with domain $[0, 1]$, continuous there, with modulus of continuity ω. Let n be a positive integer. Then

$$L_n(1) = 1, \quad [L_n(t)](x) = x, \quad [L_n(t^2)](x) = \frac{(n-1)x^2}{n} - \frac{x}{n},$$

$$(L_n([t-x]^2))(x) = \frac{x - x^2}{n}$$

and by Theorem 2.4.22,

$$\max_{0 \le x \le 1} |f(x) - (L_n f)(x)| \le 2\omega\left(\frac{1}{2\sqrt{n}}\right) \le \frac{2\omega}{\sqrt{n}}. \tag{2.4.12}$$

We have thus obtained the known result of Bernstein polynomials. For some universal constant C and for $n = 1, 2, \dots$, the left-hand-side of (2.4.12) is bounded above by $\dfrac{C\omega}{\sqrt{n}}$.

Remark 2.4.26. For more details about this constant C, see the bibliography.

Theorem 2.4.27. *Let $-\infty < a < b < \infty$ and let $L_1, L_2, \dots$ be linear positive operators, all having the same domain D which contains the bounded functions f_0, f_1, f_2 with domain $[a,b]$. We assume that the restriction of 1 to $[a,b]$ belongs to D, and that, for $n = 1, 2, \dots$, $L_n(1)$ is bounded. Let $a_0(x), a_1(x), a_2(x)$ be real functions, defined and bounded in $[a,b]$, and assume that for every $t, x \in [a,b]$,*

$$F(t,x) = \sum_{k=0}^{2} a_k(x) f_k(t) \geq K(t-x)^2, \quad F(x,x) = 0, \tag{2.4.13}$$

where K is a positive constant depending on neither x nor t. Let $f \in D$ be continuous in $[a,b]$, with modulus of continuity ω. Then, for $n = 1, 2, \dots$ we have again

$$\|f - L_n f\| \leq \|f\| \cdot \|L_n(1) - 1\| + \|L_n(1) + 1\| \omega(\mu_n), \tag{2.4.14}$$

where

$$\mu_n = (\|L_n F(t,x)(x)\|/K)^{\frac{1}{2}},$$

and $\|\cdot\|$ stands for the sup norm over $[a,b]$. In particular, if $L_n(1) = 1$, as is often the case, (2.4.14) reduces to

$$\|f - L_n f\| \leq 2\omega(\mu_n).$$

Remark 2.4.28. Since

$$\|(L_n F(t,x))(x)\| \leq \sum_{k=0}^{2} \|a_k(x)\| \cdot \|f_k - L_n f_k\| \quad (n = 1, 2, \dots),$$

if $L_n f_k$ converges uniformly to f_k in $[a,b]$, for $k = 0, 1, 2$, then $\mu_n \to 0$, and one can estimate the rapidity of convergence of μ_n in terms of those of $L_n f_0$, $L_n f_1$ and $L_n f_2$.

Proof. Let $x \in [a,b]$ and let $\delta > 0$. Let $t \in [a,b]$. If $|t-x| > \delta$, then

$$\begin{aligned} |f(t) - f(x)| &\leq \omega(|t-x|) \\ &= \omega(|t-x|\delta^{-1}\delta) \\ &\leq (1 + |t-x|\delta^{-1})\omega(\delta) \\ &\leq [1 + (t-x)^2\delta^{-2}]\omega(\delta) \\ &\leq [1 + K^{-1}F(t,x)\delta^{-2}]\omega(\delta). \end{aligned}$$

The inequality

$$|f(t) - f(x)| \leq [1 + K^{-1}F(t,x)\delta^{-2}]\omega(\delta)$$

holds, obviously, also if $|t - x| \leq \delta$. Let n be a positive integer. Then

$$\begin{aligned} |(L_n f - f(x)L_n(1))(x)| &\leq \omega(\delta)[(L_n(1) + \delta^{-2}K^{-1}L_n F(t,x))(x)] \\ &\leq \omega(\delta)[(L_n(1))(x) + (\mu_n/\delta)^2]. \end{aligned}$$

We proceed, then, as in the proof of Theorem 2.4.22. ■

Remark 2.4.29. For $f_0(t) = 1$, $f_1(t) = t$ and $f_2(t) = t^2$ we observe that Theorem 2.4.22 is a special case of Theorem 2.4.27.

Chapter 3

Numerical integration

Definition 3.0.30. *Formula*

$$I(f) = Q(f) + R(f), \tag{3.0.1}$$

where

$$Q(f) = \sum_{k=0}^{m} A_k \lambda_k(f),$$

is called a numerical integration formula *(of function f) or* a quadrature formula*, A_k, $k = 0, 1, ..., m$ are called* the coefficients *of the quadrature formula and $R(f)$ is* the remainder term.

Definition 3.0.31. *The natural number r such that $R(f) = 0$, $f \in \mathbb{P}_r$ and for which there exists $g \in \mathbb{P}_{r+1}$ such that $R(g) \neq 0$, is called* the degree of exactness *of the quadrature Q, i.e.,*

$$\text{dex}(Q) = r.$$

Remark 3.0.32. The linearity of the remainder operator R implies that $\text{dex}(Q) = r$ if and only if

$$R(e_i) = 0, \qquad i = 0, 1, ..., r$$

and

$$R(e_{r+1}) \neq 0,$$

with $e_i(x) = x^i$.

Remark 3.0.33. Usually, the information $\lambda_k(f)$, $k = 0, 1, ...m$, are the values of f or of certain of its derivatives at some points $x_i \in [a, b]$, $i = 0, 1, ...n$, called *the quadrature nodes.*

So, let $f \in H^{r,2}[a, b]$, and

$$\Lambda_B = \left\{ \lambda_{kj} : H^{r,2}[a,b] \to \mathbb{R} \,\middle|\, \lambda_{kj}(f) = f^{(j)}(x_k), \qquad k = 0, 1, ..., m;\ j \in I_k \right\},$$

with $I_k \subseteq \{0, 1, ..., r_k\}$, $r_k \in \mathbb{N}$, $r_k < r$, $k = 0, 1, ..., m$, be the set of Birkhoff-type functionals (which is the most general punctual evaluation of f and some of its derivatives), i.e.,

$$\Lambda_B(f) = \{\lambda_{kj}(f) \ \mid k = 0, 1, ..., m;\ j \in I_k\}.$$

In this case the quadrature formula is:

$$I(f) = Q_n(f) + R_n(f), \tag{3.0.2}$$

where

$$Q_n(f) = \sum_{k=0}^{m} \sum_{j \in I_k} A_{kj} f^{(j)}(x_k),$$

with $n = |I_0| + ... + |I_m| - 1$. The problem which appears is to find the coefficients A_{kj} and the nodes x_k, $k = 0, 1, ..., m$; $j \in I_k$, and to study the remainder term (the approximation error) for the obtained values of coefficients and nodes. Relatively to the conditions used to find the quadrature parameters A_{kj} and x_k, $k = 0, 1, ..., m$; $j \in I_k$, the quadrature formulas can be classified as follows:

- quadrature formulas of interpolatory type, that are obtained integrating an interpolation formula;
- quadrature of Gauss type or with the maximum degree of exactness;
- quadrature of Chebyshev type or with equal coefficients and of the maximum degree of exactness;
- optimal quadrature formulas.

Next, we are dealing with the last class.

3.1 Optimal quadrature formulas

Definition 3.1.1. *For a given $f \in H^{r,2}[a, b]$, the quadrature formula (3.0.2) for which the error $|R_n(f)|$ takes the minimum value with regard to the parameters A_{kj} and x_k, $k = 0, 1, ..., m$; $j \in I_k$, is called* the local optimal quadrature formula *with regard to the error.*

Definition 3.1.2. *A quadrature formula for which*

$$\sup_{f\in H^{r,2}[a,b]} |R_n(f)|$$

takes the minimum value with regard to its parameters A_{kj}, x_k; $k = 0, 1, ..., m$; $j \in I_k$, *is called* a global optimal formula with regard to the error.

Remark 3.1.3. The parameters A^*_{kj} and x^*_k, $k = 0, 1, ..., m$ of an optimal quadrature formula are called *the (local or global) optimal coefficients,* respectively, *the optimal nodes.*

Remark 3.1.4. Some of the parameters of a quadrature formula can be fixed for the beginning. Such formulas are the quadrature formulas with equal coefficients or with fixed nodes, (usually equally spaced).

Very often, there are studied the quadrature formulas with a given degree of exactness, say r. In such a case the parameters of a quadrature formulas must satisfy the following relations:

$$\sum_{k=0}^{m}\sum_{j\in I_k} A_{kj}x_k^i = \int_a^b w(x)x^i dx, \qquad i = 0, 1, ..., r. \tag{3.1.1}$$

If $r = n + 1$ then (3.1.1) becomes an algebraic $(n + 1) \times (n + 1)$ system. If this system has a solution then all the parameters are determinated, (not free parameters remained), and the optimality problem is superfluous. Such examples are the quadrature formulas of Gauss type or of Chebyshev type.

3.1.1 Optimality in the sense of Sard

This problem will be treated first for some classes of linear functionals, including numerical integration of functions.

Let $\lambda : H^{m,2}[a, b] \to \mathbb{R}$ be a linear functional that commutes with the defined integral, i.e.,

$$\lambda \int_a^b = \int_a^b \lambda,$$

and

$$\Lambda = \left\{\lambda_i \mid \lambda_i : H^{m,2}[a, b] \to \mathbb{R},\ i = 1, ..., n\right\}$$

be a set of given linear functionals such that $\lambda, \lambda_1, ..., \lambda_n$ are linearly independent. For $f \in H^{m,2}[a, b]$, one considers the approximation formula:

$$\lambda(f) = \sum_{i=1}^{n} A_i \lambda_i(f) + R(f). \tag{3.1.2}$$

Definition 3.1.5. *Formula (3.1.2) is called* optimal in the sense of Sard *if:*

$$\begin{array}{l} 1.\ R(e_j) = 0, \quad j = 0, 1, ..., m-1, \\ 2.\ \int_a^b K^2(t)\, dt \ \textit{is minimum}, \end{array} \tag{3.1.3}$$

where $e_j(x) = x^j$ *and* K *is the Peano's kernel of the remainder functional* R_n, *i.e.,*

$$K(t) = R\left[\frac{(\cdot - t)_+^{m-1}}{(m-1)!}\right].$$

The coefficients of the optimal formula, denoted A_i^*, $i = 1, ..., n$, *are called optimal coefficients.*

Remark 3.1.6. From 1. of (3.1.3) it follows that formula (3.1.2) has the degree of exactness at least $m - 1$.

Remark 3.1.7. If $n = m$ $(n \geq m)$ the parameters A_i, $i = 1, ..., n$ can be determined from the conditions 1. of (3.1.3), which become a $m \times m$ linear algebraic system. Conditions 2. are superfluous.

First, one supposes that Λ is a set of Lagrange-type functionals:

$$\Lambda := \Lambda_L = \{\lambda_i \mid \lambda_i(f) = f(x_i),\ i = 1, ..., n\},$$

with $x_i \in [a, b]$, $\quad x_i \neq x_j$ for $i \neq j$, $i, j = 1, ..., n$.

Formula (3.1.2) becomes

$$\lambda(f) = \sum_{i=1}^{n} A_i f(x_i) + R(f), \tag{3.1.4}$$

and we have

$$K(t) = \lambda^x\left[\frac{(x-t)_+^{m-1}}{(m-1)!}\right] - \sum_{i=1}^{n} A_i \frac{(x_i - t)_+^{m-1}}{(m-1)!}. \tag{3.1.5}$$

Theorem 3.1.8. *If* $n > m$ *then the formula of the form (3.1.4), that is optimal in the sense of Sard, exists and is unique.*

Proof. Using the Lagrange's multiplier method, we must find the minimum of the functional

$$F(A,\gamma)=\tfrac{1}{2}\int_a^b K^2(t)\,dt+\sum_{j=0}^{m-1}\gamma_j R(e_j),$$

where $A=(A_1,...,A_n)\in\mathbb{R}^n$ and $\gamma=(\gamma_0,...,\gamma_{m-1})\in\mathbb{R}^m$. So, we have to study the linear algebraic system:

$$\begin{cases}\frac{\partial F(A,\gamma)}{\partial A_p}=0, & p=1,...,n,\\ \frac{\partial F(A,\gamma)}{\partial \gamma_q}=0, & q=0,1,...,m-1,\end{cases}$$

or

$$\begin{cases}\int_a^b\left\{-\lambda^x\left[\frac{(x-t)_+^{m-1}}{(m-1)!}\right]+\sum_{i=1}^n A_i\frac{(x_i-t)_+^{m-1}}{(m-1)!}\right\}\frac{(x_p-t)_+^{m-1}}{(m-1)!}dt+\sum_{j=0}^{m-1}\gamma_j x_p^j=0,\\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad p=1,...,n\\ R(e_q)=0, \quad q=0,1,...,m-1.\end{cases}$$

Using the identities:

$$\begin{aligned}\frac{(x-t)_+^{m-1}}{(m-1)!}&=\frac{(x-t)^{m-1}}{(m-1)!}+(-1)^m\frac{(t-x)_+^{m-1}}{(m-1)!},\\ \frac{(x_i-t)_+^{m-1}}{(m-1)!}&=\frac{(x_i-t)^{m-1}}{(m-1)!}+(-1)^m\frac{(t-x_i)_+^{m-1}}{(m-1)!},\end{aligned}$$

and the relation

$$-\lambda^x\left[\frac{(x-t)^{m-1}}{(m-1)!}\right]+\sum_{i=1}^n A_i\frac{(x_i-t)^{m-1}}{(m-1)!}=0,$$

based on the Remark 3.1.6, one obtains

$$\begin{cases}(-1)^m\int_a^b\left\{-\lambda^x\left[\frac{(t-x)_+^{m-1}}{(m-1)!}\right]+\sum_{i=1}^n A_i\frac{(t-x_i)_+^{m-1}}{(m-1)!}\right\}\frac{(x_p-t)_+^{m-1}}{(m-1)!}dt+\sum_{j=0}^{m-1}\gamma_j x_p^j=0,\\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad p=1,...,n\\ \sum_{i=1}^n A_i x_i^q=\lambda(e_q), \quad q=0,1,...,m-1.\end{cases}$$

Taking into account the commuting property of λ and $\int_a^b$, the above system can be written in the form:

$$\begin{cases} \sum\limits_{j=0}^{m-1} B_j x_p^j + \sum\limits_{i=1}^{n} A_i \frac{(x_p - x_i)_+^{2m-1}}{(2m-1)!} = \lambda^x \left[\frac{(x_p - x)_+^{2m-1}}{(2m-1)!} \right], \ p = 1, ..., n \\ \sum\limits_{i=1}^{n} A_i x_i^q = \lambda (e_q), \quad q = 0, 1, ..., m-1, \end{cases} \tag{3.1.6}$$

where

$$B_j = (-1)^{m+1} \gamma_j, \ j = 0, 1, ..., m-1$$

and

$$\frac{(x_p - x_i)_+^{2m-1}}{(2m-1)!} = \int_a^b \frac{(t-x_i)_+^{m-1}}{(m-1)!} \frac{(x_p - t)_+^{m-1}}{(m-1)!} dt,$$
$$\frac{(x_p - x)_+^{2m-1}}{(2m-1)!} = \int_a^b \frac{(t-x)_+^{m-1}}{(m-1)!} \frac{(x_p - t)_+^{m-1}}{(m-1)!} dt.$$

On the other hand, for $f \in H^{m,2}[a,b]$ the coefficients of the interpolation spline function of Lagrange type,

$$(Sf)(x) = \sum_{j=0}^{m-1} a_j x^j + \sum_{i=1}^{n} b_i \frac{(x - x_i)_+^{2m-1}}{(2m-1)!},$$

are obtained as a solution of the system

$$\begin{cases} (Sf)(x_p) := \sum\limits_{j=0}^{m-1} a_j x_p^j + \sum\limits_{i=1}^{n} b_i \frac{(x_p - x_i)_+^{2m-1}}{(2m-1)!} = f(x_p), \ p = 1, ..., n \\ (Sf)^{(q)}(\alpha) := \sum\limits_{i=1}^{n} b_i \frac{(\alpha - x_i)_+^{2m-q-1}}{(2m-q-1)!} = 0, \ q = m, ..., 2m-1; \ \alpha > x_n. \end{cases} \tag{3.1.7}$$

As the last m equations from (3.1.7) can be written in the equivalent form

$$\sum_{i=1}^{n} b_i x_i^q = 0, \quad q = 0, 1, ..., m-1,$$

the two systems, (3.1.6) and (3.1.7), are equivalent. Hence, the uniqueness of the spline function Sf, implies that the system (3.1.6) has a unique solution, say (A^*, γ^*), i.e., $A^* = (A_1^*, ..., A_n^*)$ is the unique minimum point of the integral from (3.1.3). ■

Theorem 3.1.9. *For $f \in H^{m,2}[a,b]$, if*

$$f = Sf + Rf$$

is the spline interpolation formula corresponding to $\Lambda := \Lambda_L$, with $n > m$, then the approximation formula of the form (3.1.4), that is optimal in sense of Sard, is

$$\lambda(f) = \lambda(Sf) + \lambda(Rf). \tag{3.1.8}$$

Proof. As $n := |\Lambda_L| > m$ formula (3.1.8) exists, is unique and we have

$$f(x) = \sum_{i=1}^{n} s_i(x) f(x_i) + \int_a^b \varphi(x,t) f^{(m)}(t)\, dt,$$

where

$$\varphi(x,t) = \frac{(x-t)_+^{m-1}}{(m-1)!} - \sum_{i=1}^{n} s_i(x) \frac{(x_i-t)_+^{m-1}}{(m-1)!}.$$

It follows that

$$\lambda(f) = \sum_{i=1}^{n} A_i^* f(x_i) + R^*(f), \tag{3.1.9}$$

with

$$A_i^* = \lambda(s_i), \quad i = 1, ..., n$$

and

$$R^*(f) = \int_a^b K^*(t) f^{(m)}(t)\, dt, \tag{3.1.10}$$

where

$$K^*(t) = \lambda^x \left[\frac{(x-t)_+^{m-1}}{(m-1)!}\right] - \sum_{i=1}^{n} A_i^* \frac{(x_i-t)_+^{m-1}}{(m-1)!}.$$

Formula (3.1.9) is optimal in the sense of Sard if there are satisfied conditions (3.1.3). The first condition follows directly from (3.1.10), i.e.,

$$R^*(e_j) = 0, \qquad j = 0, 1, ...m-1.$$

To prove the second condition, let us suppose that the formula (3.1.9) is not optimal. Let

$$\lambda(f) = \sum_{i=1}^{n} \bar{A}_i f(x_i) + \bar{R}(f) \tag{3.1.11}$$

be the optimal formula. For the remainder term, we have (condition 1. from (3.1.3)):

$$\bar{R}(f) = \int_a^b \bar{K}(t) f^{(m)}(t)\, dt,$$

with

$$\bar{K}(t) = \lambda^x \left[\frac{(x-t)_+^{m-1}}{(m-1)!}\right] - \sum_{i=1}^n \bar{A}_i \frac{(x_i-t)_+^{m-1}}{(m-1)!}.$$

One considers

$$g = \bar{K} - K^*,$$

i.e.,

$$g(t) = \sum_{i=1}^n \left[\bar{A}_i - A_i^*\right] \frac{(x_i-t)_+^{m-1}}{(m-1)!}. \tag{3.1.12}$$

We have

$$g(t) = 0, \quad \text{for } t \in [a, x_1) \cup (x_n, b]. \tag{3.1.13}$$

For $t < x_1$, we have

$$g(t) = \sum_{i=1}^n (\bar{A}_i - A_i^*) \frac{(x_i-t)^{m-1}}{(m-1)!}.$$

As the degree of exactness of both formulas (3.1.9) and (3.1.11) is $(m-1)$, it follows that $g(t) = 0$, for $t < x_1$.

For $t > x_n$ we have $(x_i - t)_+ = 0$, $i = 1, ..., n$, and we also get $g(t) = 0$.

Now, let us define the function s such that $s^{(m)} = g$. From (3.1.12) and (3.1.13) it follows that

$$\begin{cases} s|_{(x_i, x_{i+1})} \in \mathbb{P}_{2m-1}, & i = 1, ..., n-1, \\ s|_{[a,x_1)\cup(x_n,b]} \in \mathbb{P}_{m-1}, \end{cases}$$

and

$$s \in C^{2m-2}[a, b].$$

Whence, we have $s \in S(\Lambda)$. Since

$$Sf = f, \quad f \in S(\Lambda),$$

we have

$$R^*(s) := \int_a^b K^*(t) s^{(m)}(t)\, dt = 0,$$

i.e.,

$$\int_a^b K^*(t)\, g(t)\, dt = 0$$

or

$$\int_a^b K^*(t) \left[\bar{K}(t) - K^*(t)\right] dt = 0.$$

It follows that

$$\begin{aligned}\int_a^b \left(\bar{K}(t)\right)^2 dt &= \int_a^b \left[\bar{K}(t) - K^*(t) + K^*(t)\right]^2 dt \\ &\geq \int_a^b \left(K^*(t)\right)^2 dt + \int_a^b \left[\bar{K}(t) - K^*(t)\right]^2 dt,\end{aligned}$$

whence,

$$\int_a^b \left(\bar{K}(t)\right)^2 dt \geq \int_a^b \left(K^*(t)\right)^2 dt.$$

As $\bar{K}$ is the kernel of the optimal formula it must be $\bar{K} = K^*$, i.e., $g = 0$ or $\bar{A}_i = A_i^*$, $i = 1, ..., n$. ■

Remark 3.1.10. Analogous results can be proved when Λ is a set of Hermite or Birkhoff-type functionals for which the corresponding spline interpolation formula exists.

Suppose now that Λ is a Hermite-type set of functionals on $H^{m,2}[a,b]$,

$$\Lambda := \Lambda_H = \left\{\lambda_{ij} \mid \lambda_{ij} = f^{(j)}(x_i),\ i = 1, ..., r;\ j = 0, ..., \nu_i\right\},$$

with $0 \leq \nu_i < m$ and $n = r + \nu_1 + ... + \nu_r$.

Remark 3.1.11. For $f \in H^{m,2}[a,b]$ formula (3.1.2) becomes

$$\lambda(f) = \sum_{i=1}^{r} \sum_{j=0}^{\nu_i} A_{ij} f^{(j)}(x_i) + R(f), \tag{3.1.14}$$

where

$$R(f) = \int_a^b K(t)\, f^{(n)}(t)\, dt,$$

with

$$K(t) = \lambda^x \left[\frac{(x-t)_+^{m-1}}{(m-1)!}\right] - \sum_{i=1}^{r} \sum_{j=0}^{\nu_i} A_{ij} \frac{(x_i-t)_+^{m-j-1}}{(m-j-1)!}.$$

Theorem 3.1.12. *If $n > m$ formula (3.1.14), that is optimal in sense of Sard, exists and is unique.*

Proof. As in the previous case, one considers the functional $F_H(A,\gamma)$ given by

$$F_H(A,\gamma) = \frac{1}{2}\int_a^b K^2(t)\,dt + \sum_{k=0}^{m-1}\gamma_k R\left(\frac{e_k}{k!}\right),$$

where $A = (A_{10},...,A_{1\nu_1},...,A_{r0},...,A_{r\nu_r})$ and $\gamma = (\gamma_0,...,\gamma_{m-1})$. The parameters A_{ij}, $i = 1,...,r$; $j = 0,1,...,\nu_i$ and γ_k, $k = 0,1,...,m-1$ are determined as a solution of the linear algebraic system:

$$\begin{cases} \frac{\partial F(A,\gamma)}{\partial A_{pq}} = 0, & p = 1,...,r; \quad q = 0,1,...,\nu_p, \\ \frac{\partial F(A,\gamma)}{\partial \gamma_\mu} = 0, & \mu = 0,1,...,m-1. \end{cases} \tag{3.1.15}$$

Let S_H be the spline interpolation operator, corresponding to the set Λ_H, i.e.,

$$(S_H f)(x) = \sum_{k=0}^{m-1} a_k x^k + \sum_{i=1}^{r}\sum_{j=0}^{\nu_i} b_{ij}\frac{(x-x_i)_+^{2m-j-1}}{(2m-j-1)!},$$

where the coefficients a_k, $k = 0,1,...,m-1$ and b_{ij}, $i = 1,...,r$; $j = 0,1,...,\nu_i$ are obtained as a solution of the following linear algebraic system:

$$\begin{cases} (S_H f)^{(j)}(x_k) = f^{(j)}(x_k), & k = 1,...,r; \quad j = 0,1,...,\nu_k \\ (S_H f)^{(p)}(\alpha) = 0, & p = m,...,2m-1, \quad \alpha > r. \end{cases} \tag{3.1.16}$$

After some transformations (as in the previous case) we can see that the systems (3.1.15) and (3.1.16) are equivalent. But, for $n > m$ the spline function $S_H f$ exists and is unique, hence the system (3.1.16) has a unique solution. It means that the system (3.1.15) has also a unique solution and the proof follows. ■

Remark 3.1.13. A theorem analogous to Theorem 3.1.12 can be also proved when Λ is a set of functionals of Birkhoff-type in the additional condition that Λ contains a subset of at least m Hermite-type functionals (condition under which the corresponding spline interpolation function exists and is unique).

In particular, let λ be the defined integral functional:

$$\lambda(f) := I(f) = \int_a^b f(x)\,dx, \text{ for } \quad f \in H^{m,2}[a,b].$$

For

$$\Lambda = \left\{ \lambda_i \mid \lambda_i : H^{m,2}[a,b] \to \mathbb{R},\ i = 1, ..., n \right\},$$

one considers the quadrature formula

$$I(f) = Q_n(f) + R_n(f), \tag{3.1.17}$$

where

$$Q_n(f) = \sum_{i=1}^{n} A_i \lambda_i(f)$$

and $R_n(f)$ is the remainder term.

The quadrature formula (3.1.17) is optimal in the sense of Sard if

$$R_n(e_j) = 0, \quad j = 0, 1, ..., m-1$$

and

$$\int_a^b K^2(t)\, dx \text{ is minimum,}$$

with

$$K(t) := R_n\left[\frac{(\cdot - t)_+^{m-1}}{(m-1)!}\right].$$

Now, if Λ is a Birkhoff-type set of functionals that contains a subset of at least m functionals of Hermite-type then the corresponding spline interpolation formula,

$$f = Sf + Rf,$$

exists and is unique.

By Theorem 3.1.9, it follows that the formula

$$\int_a^b f(x)\, dx = \int_a^b (Sf)(x)\, dx + \int_a^b (Rf)(x)\, dx$$

is the optimal quadrature formula in sense of Sard corresponding to the set Λ.

Next, there will be given some examples of such optimal quadrature formulas for Lagrange-type, Hermite-type and Birkhoff-type sets of functionals.

Example 3.1.14. Let $f \in C[0,1]$ and

$$\Lambda_L(f) = \left\{ f\left(\tfrac{i}{n}\right) \mid i = 0, ..., n \right\}.$$

We construct the optimal quadrature formula in sense of Sard, using linear splines ($m = 1$).

We have

$$f = S_1 f + R_1 f,$$

where

$$(S_1 f)(x) = \sum_{i=0}^{n} s_i(x) f\left(\tfrac{i}{n}\right)$$

and

$$(R_1 f)(x) = \int_0^1 \varphi_1(x,t) f'(t)\, dt,$$

with

$$\varphi_1(x,t) = (x-t)_+^0 - \sum_{i=0}^{n} s_i(x) \left(\tfrac{i}{n} - t\right)_+^0.$$

Next, we determine the functions $s_i, i = 0, 1, ..., n$. We have that

$$s_i(x) = a_0^i + \sum_{j=0}^{n} b_j^i \left(x - \tfrac{j}{n}\right)_+, \quad i = 0, 1, ..., n,$$

with the coefficients $a_0^i, b_0^i, ..., b_n^i$, determined, for any $i = 0, 1, ..., n$, from the conditions:

$$\begin{cases} s_i\left(\frac{k}{n}\right) = \delta_{ik}, & k = 0, 1, ..., n \\ s_i'(\alpha) = 0, & \alpha > 1, \quad \text{(we take } \alpha = 2\text{)}. \end{cases}$$

One obtains

$$\begin{aligned}
s_0(x) &= 1 - nx + n\left(x - \tfrac{1}{n}\right)_+ \\
s_1(x) &= nx - 2n\left(x - \tfrac{1}{n}\right)_+ + n\left(x - \tfrac{2}{n}\right)_+ \\
s_2(x) &= n\left(x - \tfrac{1}{n}\right)_+ - 2n\left(x - \tfrac{2}{n}\right)_+ + n\left(x - \tfrac{3}{n}\right)_+ \\
&\cdots \\
s_{n-1}(x) &= n\left(x - \tfrac{n-2}{n}\right)_+ - 2n\left(x - \tfrac{n-1}{n}\right)_+ + n(x-1)_+ \\
s_n(x) &= n\left(x - \tfrac{n-1}{n}\right)_+ - n(x-1)_+.
\end{aligned}$$

It follows that

$$\int_0^1 f(x)\,dx = \sum_{i=0}^{n} A_i^* f\left(\tfrac{i}{n}\right) + R^*(f),$$

where

$$A_i^* = \int_0^1 s_i(x)\,dx, \quad i = 0, ..., n,$$

i.e.,

$$\begin{aligned} A_0^* &= \tfrac{1}{2n} \\ A_1^* &= \tfrac{1}{n} \\ &\vdots \\ A_i^* &= \tfrac{1}{n} \\ &\vdots \\ A_{n-1}^* &= \tfrac{1}{n} \\ A_n^* &= \tfrac{1}{2n}, \end{aligned}$$

and

$$R^*(f) = \int_0^1 K_1(t) f'(t)\,dt,$$

with

$$K_1(t) = 1 - t - \sum_{i=0}^{n} A_i^* \left(\tfrac{i}{n} - t\right)_+^0.$$

Hence, the optimal quadrature formula is

$$\int_0^1 f(x)\,dx = \tfrac{1}{n}\left[\tfrac{1}{2}f(0) + f\left(\tfrac{1}{n}\right) + ... + f\left(\tfrac{n-1}{n}\right) + \tfrac{1}{2}f(n)\right] + R^*(f).$$

For the remainder term we have

$$|R^*(f)| \le \|f'\|_2 \int_0^1 K_1^2(t)dt = \|f'\|_2 \sum_{i=1}^{n} \int_{\frac{i-1}{n}}^{\frac{i}{n}} K_{1,i}^2(t)dt \le \tfrac{1}{2n\sqrt{3}} \|f'\|_2,$$

where $K_{1,i} = K_1|_{\left[\frac{i-1}{n}, \frac{i}{n}\right]}$.

Example 3.1.15. Let $f \in C^2[0,1]$ and

$$\Lambda_H(f) = \left\{f(0), f'(0), f\left(\tfrac{1}{2}\right), f(1), f'(1)\right\}$$

be given. We construct the optimal quadrature formula (in the sense of Sard). To this end, one considers the cubic spline interpolation formula:

$$f = S_3 f + R_3 f,$$

where

$$(S_3 f)(x) = s_{00}(x) f(0) + s_{01}(x) f'(0) + s_{10}(x) f\left(\tfrac{1}{2}\right) + s_{20}(x) f(1) + s_{21}(x) f'(1)$$

and the fundamental interpolation splines have the expressions

$$s_{kj}(x) = a_0^{kj} + a_1^{kj} x + b_{00}^{kj} x^3 + b_{01}^{kj} x^2 + b_{10}^{kj}\left(x - \tfrac{1}{2}\right)_+^3 + b_{20}^{kj}(x-1)_+^3 + b_{21}^{kj}(x-1)_+^2,$$

for $(k,j) \in \{(0,0);(0,1);(1,0);(2,0);(2,1)\}$. For $(k,j) = (0,0)$ we have the system:

$$\begin{cases} s_{00}(0) := a_0^{00} & = 1 \\ s'_{00}(0) := a_1^{00} & = 0 \\ s_{00}\left(\tfrac{1}{2}\right) := a_0^{00} + \dfrac{1}{2}a_1^{00} + \dfrac{1}{8}b_{00}^{00} + \tfrac{1}{4}b_{01}^{00} & = 0 \\ s_{00}(1) := a_0^{00} + a_1^{00} + b_{00}^{00} + b_{01}^{00} + \tfrac{1}{8}b_{10}^{00} & = 0 \\ s'_{00}(1) := a_1^{00} + 3b_{00}^{00} + 2b_{01}^{00} + \dfrac{3}{4}b_{10}^{00} & = 0 \\ s''_{00}(2) := 12b_{00}^{00} + 2b_{01}^{00} + 9b_{10}^{00} + 6b_{20}^{00} + 2b_{21}^{00} & = 0 \\ s'''_{00}(2) := b_{00}^{00} + b_{10}^{00} + b_{20}^{00} & = 0. \end{cases}$$

Solving this system, one obtains

$$s_{00}(x) = 1 + 10x^3 - 9x^2 - 16\left(x - \tfrac{1}{2}\right)_+^3.$$

The systems that generate the coefficients of the other fundamental interpolation spline functions have the same matrix, free terms being changed successively in $(0,1,0,0,0,0,0)$, $(0,0,1,0,0,0,0)$, $(0,0,0,1,0,0,0)$, $(0,0,0,0,1,0,0)$.

So, we have

$$\begin{aligned} s_{01}(x) &= x + 3x^3 - \tfrac{7}{2}x - 4\left(x - \tfrac{1}{2}\right)_+^3, \\ s_{10}(x) &= -16x^3 + 12x^2 + 32\left(x - \tfrac{1}{2}\right)_+^3, \\ s_{20}(x) &= 6x^3 - 3x^2 - 16\left(x - \tfrac{1}{2}\right)_+^3, \\ s_{21}(x) &= -x^3 + \tfrac{1}{2}x^2 + 4\left(x - \tfrac{1}{2}\right)_+^3. \end{aligned}$$

For the remainder term, we have:

$$(R_3 f)(x) = \int_0^1 K_2(x,t) f''(t)\, dt,$$

where

$$K_2(x,t) = (x-t)_+ - s_{10}(x)\left(\tfrac{1}{2}-t\right)_+ - s_{20}(x)(1-t) - s_{21}(x).$$

It follows that

$$\int_0^1 f(x)\, dx = A_{00}^* f(0) + A_{01}^* f'(0) + A_{10}^* f\left(\tfrac{1}{2}\right) + A_{20}^* f(1) + A_{21}^* f'(1) + R^*(f),$$

with

$$A_{00}^* = \int_0^1 s_{00}(x)\, dx = \tfrac{1}{4},$$
$$A_{01}^* = \int_0^1 s_{01}(x)\, dx = \tfrac{1}{48},$$
$$A_{10}^* = \int_0^1 s_{10}(x)\, dx = \tfrac{1}{2},$$
$$A_{20}^* = \int_0^1 s_{20}(x)\, dx = \tfrac{1}{4},$$
$$A_{21}^* = \int_0^1 s_{21}(x)\, dx = -\tfrac{1}{48},$$

and

$$R^*(f) = \int_0^1 K_2(t) f''(t)\, dt,$$

where

$$K_2(t) = \tfrac{(1-t)^2}{2} - A_{10}^*\left(\tfrac{1}{2}-t\right)_+ - A_{20}^*(1-t) - A_{21}^*.$$

For the remainder we have

$$|R^*(f)| \le \left(\int_0^1 K_2^2(t)\, dt\right)^{1/2} \|f''\|_2.$$

But,

$$\int_0^1 K_2^2(t)\, dt = \int_0^{1/2} K_2^2(t)\, dt + \int_{1/2}^1 K_2^2(t)\, dt.$$

We obtain

$$\int_0^1 f(x)\,dx = \left[f(0) + \tfrac{1}{48}f'(0) + \tfrac{1}{2}f\left(\tfrac{1}{2}\right) + f(1) - \tfrac{1}{48}f'(1)\right] + R^*(f),$$

with

$$|R^*(f)| \leq \tfrac{1}{48\sqrt{5}}\|f''\|_2.$$

Example 3.1.16. Let $f \in C^2[0,1]$ and

$$\Lambda_B(f) = \left\{f'(0), f\left(\tfrac{1}{2}\right), f'(1)\right\}$$

be given. We need to construct the optimal quadrature formula (in the sense of Sard). Similarly to the Hermite case, we consider the cubic spline formula

$$(S_3 f)(x) = s_{01}(x) f'(0) + s_{10}(x) f\left(\tfrac{1}{2}\right) + s_{21}(x) f'(1),$$

and we determine the functions s_{01}, s_{10} and s_{21}.

The coefficients of this functions are solutions of a 5×5 system. For s_{01} we have

$$\begin{cases} s'_{01}(0) := a_1^{01} & = 1 \\ s_{01}\left(\frac{1}{2}\right) := a_0^{01} + \frac{1}{2}a_1^{01} + \frac{1}{4}b_{01}^{01} & = 0 \\ s'_{01}(1) := a_1^{01} + 2b_{01}^{01} + \frac{3}{4}b_{10}^{01} & = 0 \\ s''_{01}(2) := 12b_{01}^{01} + 9b_{10}^{01} + 2b_{21}^{01} & = 0 \\ s'''_{01}(2) := 6b_{10}^{01} & = 0. \end{cases}$$

It follows that

$$s_{01}(x) = -\tfrac{3}{8} + x - \tfrac{1}{2}x^2 + \tfrac{1}{2}(x-1)_+^2.$$

In the same way we obtain the other functions:

$$\begin{aligned} s_{10}(x) &= 1, \\ s_{21}(x) &= -\tfrac{1}{8} + \tfrac{1}{2}x^2 - \tfrac{1}{2}(x-1)_+^2. \end{aligned}$$

For the remainder term, we have:

$$(R_3 f)(x) = \int_0^1 K_2(x,t) f''(x,t)\,dt,$$

where

$$K_2(x,t) = (x-t)_+ - \left(\tfrac{1}{2} - t\right)_+ - s_{21}(x).$$

One obtains the optimal formula:

$$\int_0^1 f(x)\,dx = A_{01}^* f'(0) + A_{10}^* f\left(\tfrac{1}{2}\right) + A_{21}^* f'(1) + R^*(f),$$

where

$$A_{01}^* = \int_0^1 s_{01}(x)\,dx = -\tfrac{1}{24},$$
$$A_{10}^* = \int_0^1 s_{10}(x)\,dx = 1,$$
$$A_{21}^* = \int_0^1 s_{21}(x)\,dx = \tfrac{1}{24},$$

and

$$R^*(f) = \int_0^1 K_2^*(t)\, f''(t)\,dt,$$

with

$$K_2^*(t) = \int_0^1 K_2(x,t)\,dx = \tfrac{(1-t)^2}{2} - \left(\tfrac{1}{2} - t\right)_+ - \tfrac{1}{24}.$$

Finally, we get

$$\int_0^1 f(x)\,dx = -\tfrac{1}{24} f'(0) + f\left(\tfrac{1}{2}\right) + \tfrac{1}{24} f'(1) + R^*(f),$$

with

$$|R^*(f)| \leq \tfrac{1}{12\sqrt{5}} \|f''\|_2.$$

3.1.2 Optimality in the sense of Nikolski

Consider a quadrature formula of the form given in (3.0.2). One suppose that all the parameters of the quadrature formula are free. So, we have to find the coefficients and the nodes of the quadrature formula such that $|R_n(f)|$ takes the minimum value. Sometimes there can be supposed that the quadrature formula has a given degree of exactness, i.e., some relations as in (3.1.1) are satisfied. Here there are used two ways to solve such an optimality problem.

A FIRST WAY is based on the relationship between the optimality in the sense of Sard and the spline interpolation. For the beginning, suppose that the quadrature nodes are fixed and find the coefficients such that $|R_n(f)|$ is

minimum. But, this is a problem of optimality in the sense of Sard and it can be solved using the suitable spline interpolation formula. So, if

$$f = S_r f + R_r f, \tag{3.1.18}$$

is the spline interpolation formula, with

$$(S_r f)(x) = \sum_{k=0}^{m} \sum_{j \in I_k} s_{kj}(x) f^{(j)}(x_k),$$

and

$$(R_r f)(x) = \int_a^b \varphi_r(x,t) f^{(r)}(t) dt,$$

then it follows that

$$\int_a^b f(x)dx = \sum_{k=0}^{m} \sum_{j \in I_k} \overline{A}_{kj} f^{(j)}(x_k) + \overline{R}_n(f), \tag{3.1.19}$$

with

$$\overline{A}_{kj} = \int_a^b s_{kj}(x)dx$$

and

$$\overline{R}_n(f) = \int_a^b \overline{K}_r(t) f^{(r)}(t) dt, \tag{3.1.20}$$

where

$$\overline{K}_r(t) = \int_a^b \varphi_r(x,t)dx = \frac{(b-t)^r}{r!} - \sum_{k=0}^{m} \sum_{j \in I_k} \overline{A}_{kj} \frac{(x_k - t)_+^{2r-j-1}}{(2r-j-1)!}$$

is the optimal in the sense of Sard quadrature formula.

As $\overline{A}_{kj}$ are functions of variables x_k, $k = 0, 1, ..., m$, i.e.,

$$\overline{A}_{kj} = F_{kj}(x_0, ...x_m), \tag{3.1.21}$$

it follows that $\overline{R}_n(f)$ is a function of the nodes,

$$\overline{R}_n(f) = F(x_0, ..., x_m). \tag{3.1.22}$$

Now, we have to minimize the function F, with respect to the variables $x_0, ..., x_m$. Let $(x_0^*, ..., x_m^*)$ be the minimum point of F. It follows that $x_0^*, ..., x_m^*$ are the optimal nodes, in the sense of Nikolski.

Then, from (3.1.21) and (3.1.22), one obtains the optimal coefficients:

$$A_{kj}^* = F_{kj}(x_0^*, ..., x_m^*), \qquad k = 0, 1, ..., m;\ j \in I_k, \tag{3.1.23}$$

respectively, the optimal remainder term:

$$R_n^*(f) = F(x_0^*, ..., x_m^*). \tag{3.1.24}$$

Thus, the quadrature formula

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

is optimal in the sense of Nikolski.

Conclusion 3.1.17. *The optimal quadrature in the sense of Nikolski, can be obtained in two steps:*

- it is constructed the optimal quadrature in the sense of Sard (3.1.19), for fixed nodes, using for example the relationship with the spline interpolation.

- it is minimized the error functional (3.1.20), that depends only of the nodes, with respect to these parameters. There are obtained the optimal nodes, x_k^*, $k = 0, 1, ..., m$. *Then (3.1.23) and (3.1.24) give the optimal coefficients* A_{kj}^*, $k = 0, 1, ..., m, ; j \in I_k$, *respectively, the optimal remainder term,* $R_n^*(f)$.

A SECOND WAY is given by the φ-function method and it is based on the minimal norm properties of orthogonal polynomials.

Let $f \in C^r[a, b]$ and $a = x_0 < x_1 < ... < x_n = b$. The φ-function method consists in associating to each interval $[x_{k-1}, x_k]$ of a function φ_k, with $\varphi_k^{(r)} = 1$, $k = 1, ..., m$.

Then,

$$\int_a^b f(x)dx = \sum_{k=1}^{m} \int_{x_{k-1}}^{x_k} f(x)dx = \sum_{k=1}^{m} \int_{x_{k-1}}^{x_k} \varphi_k^{(r)}(x) f(x)dx,$$

and applying r times the formula of integrating by parts to the last integrals, one obtains:

$$\begin{aligned}\int_a^b f(x)dx =& \sum_{k=1}^{m} \{[\varphi_k^{(r-1)}(x)f(x) - \varphi_k^{(r-2)}(x)f'(x) + ... + (-1)^{r-1}\varphi_k(x)f^{(r-1)}(x)] \Big|_{x_{k-1}}^{x_k} \\ &+ (-1)^r \int_{x_{k-1}}^{x_k} \varphi_k(x) f^{(r)}(x)dx\},\end{aligned}$$

and further,

$$\begin{aligned}\int_a^b f(x)dx =& \sum_{j=0}^{r-1}(-1)^{j+1}\varphi_1^{(r-j-1)}(x_0)f^{(j)}(x_0)\\ &+\sum_{k=1}^{m-1}\sum_{j=0}^{r-1}(-1)^j(\varphi_k-\varphi_{k+1})^{(r-j-1)}(x_i)f^{(j)}(x_i)\\ &+\sum_{j=0}^{r-1}(-1)^j\varphi_m^{(r-j-i)}(x_m)f^{(j)}(x_m)\\ &+(-1)^r\sum_{k=1}^{m}\int_{x_{k-1}}^{x_k}\varphi_k(x)f^{(r)}(x)dx.\end{aligned}$$

Therefore, considering $\varphi\,\big|_{[x_{k-1},x_k]}=\varphi_k,\ k=1,...,m,$, we have

$$\int_a^b f(x)dx=\sum_{k=0}^{m}\sum_{j=0}^{r-1}A_{kj}f^{(j)}(x_k)+R_n(f), \tag{3.1.25}$$

where

$$\begin{array}{ll}A_{0j}=(-1)^{j+1}\varphi_1^{(r-j-1)}(x_0), & j=0,1,...,r-1\\ A_{kj}=(-1)^j(\varphi_k-\varphi_{k+1})^{(r-j-1)}(x_k), & \begin{array}{l}k=1,...,\ m-1;\\ j=0,1,...,r-1\end{array}\\ A_{mj}=(-1)^j\varphi_m^{(r-j-1)}\left(x_n\right), & j=0,1,...,r-1\end{array} \tag{3.1.26}$$

and

$$R_n(f)=(-1)^r\int_a^b\varphi(x)f^{(r)}(x)dx, \tag{3.1.27}$$

with

$$\varphi(x)=\tfrac{(x_m-x)^r}{r!}+(-1)^{r+1}\sum_{k=0}^{m}\sum_{j=0}^{r-1}A_{kj}\tfrac{(x_k-x)_+^{r-j-1}}{(r-j-1)!}.$$

If $f\in H^{m,2}[a,b]$, then

$$|R_n(f)|\le\left\|f^{(r)}\right\|_2\left(\int_a^b\varphi^2(x)dx\right)^{1/2}.$$

This way, the optimal quadrature formula in the sense of Nikolski is determined by the coefficients $A := (A_{kj})_{k=0,\overline{m};\overline{j=0,r-1}}$ and the nodes $X := (x_k)_{k=\overline{0,m}}$, for which the functional

$$F(A,X) := \int_a^b \varphi^2(x)dx$$

takes the minimum value.

Remark 3.1.18. From (3.1.27) it follows that the quadrature formula (3.1.25) has the degree of exactness at least $(r-1)$.

Next, we must find the coefficients A and the nodes X that minimize the functional $F(A,X)$. We have,

$$F(A,X) = \sum_{k=1}^{m} \int_{x_{k-1}}^{x_k} \varphi_k^2(x)dx,$$

where

$$\varphi_k(x) = \frac{(x_m - x)^r}{r!} + (-1)^{r+1} \sum_{k=0}^{m} \sum_{j=0}^{r-1} A_{kj} \frac{(x_k - x)_+^{r-j-1}}{(r-j-1)!}$$

is a polynomial of r degree. So, we have to find the polynomial φ_k with the minimum $L_w^2[x_{k-1}, x_k]$ norm $(w = 1)$. This is the Legendre polynomial:

$$\tilde{l}_{r,k}(x) = \frac{r!}{(2r)!} \frac{d^r}{dx^r}[(x - x_{k-1})^r (x - x_k)^r],$$

i.e., the coefficients A_{kj} of φ_k, say $\overline{A}_{kj}$, are obtained from the identity

$$\varphi_k = \tilde{l}_{r,k}, \quad \text{for all } k = 1, ..., m.$$

Evidently, $\overline{A}_{kj} = \overline{A}_{kj}(x_1, ..., x_{m-1})$. One obtains

$$F(\overline{A}, X) = \sum_{k=1}^{m} \int_{x_{k-1}}^{x_k} \tilde{l}_{r,k}^2(x)dx,$$

that must be minimized with respect to the nodes x_k, $k = 1, ..., m-1$, $(x_0 = a,\ x_m = b)$. Let $X^* = (a, x^*, ..., x_{m-1}^*, b)$ be the minimum point. It follows that $X^* = (a, x^*, ..., x_{m-1}^*, b)$ are the optimal nodes, respectively,

$$A_{kj}^* = \overline{A}_{kj}(a, x^*, ..., x_{m-1}^*, b), \qquad k = 0, 1, ..., m;\ j = 0, 1, ..., r-1$$

are the optimal coefficients.

For the remainder term, one obtains

$$|R_m^*(f)| \leq [F(A^*, X^*)]^{1/2} \left\| f^{(r)} \right\|_2 .$$

Example 3.1.19. Let $f \in H^{1,2}[0,1]$, $0 = x_0 < x_1 < ... < x_m = 1$ be a partition of the interval $[0,1]$, and

$$I(f) := \int_0^1 f(x)dx = \sum_{k=0}^{m} A_k f(x_k) + R_m(f).$$

Applying the φ-function method ($\varphi_k' = 1$), one obtains

$$\int_0^1 f(x)dx = -\varphi_1(0) + \sum_{k=1}^{m-1} (\varphi_k - \varphi_{k+1})(x_k) f(x_k) + \varphi_m(1) f(1) - \int_0^1 \varphi(x) f'(x)dx,$$

with

$$\varphi|_{[x_{k-1}, x_k]} = \varphi_k(x) = x - \sum_{i=0}^{k-1} A_i.$$

We have

$$\begin{aligned} A_0 &= -\varphi_1(0), \\ A_k &= (\varphi_k - \varphi_{k+1})(x_k), \qquad k = 1, ..., m-1, \\ A_m &= \varphi_m(1), \end{aligned}$$

and

$$R_m(f) = -\int_0^1 \varphi(x) f'(x)dx.$$

Here

$$F(A, X) := \int_0^1 \varphi^2(x)dx = \sum_{k=1}^{m} \int_{x_{k-1}}^{x_k} \varphi_k^2(x)dx,$$

with φ_k being a polynomial of the first degree, for all $k = 1, ..., m$. But it is known that $\|\varphi_k\|_2$ takes the minimum value when $\varphi_k \equiv \tilde{l}_{1,k}$, the Legendre polynomial of first degree. It follows that

$$\sum_{i=0}^{k-1} \overline{A}_i = \tfrac{x_{k-1} + x_k}{2} \tag{3.1.28}$$

and

$$F(\overline{A}, X) = \frac{1}{12}\sum_{k=1}^{m}(x_k - x_{k-1})^3.$$

From the system of equations

$$\frac{\partial F(\overline{A}, X)}{\partial x_k} := \frac{1}{4}[(x_k - x_{x-1})^2 - (x_{k+1} - x_k)^2] = 0, \qquad k = 1, ..., m,$$

it follows that

$$x_k^* - x_{k-1}^* = x_{k+1}^* - x_k^*, \qquad k = 1, ..., m-1,$$

respectively,

$$x_k^* = \frac{k}{m}, \qquad k = 0, 1, ..., m.$$

The relations (3.1.28) become

$$\sum_{i=0}^{k-1} A_i^* = \frac{x_{k-1}^* + x_k^*}{2} = \frac{2k-1}{2}, \qquad k = 1, ..., m.$$

whence,

$$\begin{aligned} A_0^* &= \frac{1}{2m}, \\ A_1^* &= ... = A_{m-1}^* = \frac{1}{m}, \\ A_m^* &= \frac{1}{2m}. \end{aligned}$$

We also have

$$F(A^*, X^*) = \frac{1}{12m^2},$$

respectively,

$$|R_m^*(f)| \leq \frac{1}{2m\sqrt{3}} \|f'\|_2 .$$

3.2 Some optimality aspects in numerical integration of bivariate functions

Let D be a given domain in $\mathbb{R}^2$, $f : D \to \mathbb{R}$ an integrable function on D and $\Lambda(f) = \{\lambda_1(f), ..., \lambda_N(f)\}$ a set of information on f. Next, one supposes that $\lambda_i(f)$, $i = 1, ..., N$, are the values of f or of certain of its derivatives at some points from D, called *the cubature nodes*.

One considers the cubature formula:

$$I_w^{xy} f := \iint_D w(x,y) f(x,y) dxdy = \sum_{i=1}^{N} C_i \lambda_i(f) + R_N(f),$$

where w is a given weight function (often $w(x,y) = 1$ on D), C_i, $i = 1, ..., N$ are its coefficients and $R_N(f)$ is the remainder term.

The problem to construct such a cubature formula consists in the determination of the coefficients C_i, $i = 1, ..., N$ and its nodes, in some given conditions, and to evaluate the corresponding remainder term.

For some particular cases, a cubature formula can be constructed using the product or the boolean sum of two quadrature rules.

The most results have been obtained when D is a regular domain in $\mathbb{R}^2$ (rectangle, triangle, etc.) and the cubature nodes are regularly spaced.

First, one supposes that D is a rectangle, $D = [a,b] \times [c,d]$ and $w(x,y) = 1$, $(x,y) \in D$. If

$$\Lambda^x(f) = \{\lambda_i^x f \mid i = 0, 1, ..., m\},$$
$$\Lambda^y(f) = \{\lambda_j^y f \mid j = 0, 1, ..., n\},$$

are sets of partial information on f, with respect to x, respectively to y, one considers the quadrature formulas:

$$I^x f := \int_a^b f(x,y) dx = (Q_1^x f)(\cdot, y) + (R_1^x f)(\cdot, y)$$

and

$$I^y f = \int_c^d f(x,y) dy = (Q_1^y f)(x, \cdot) + (R_1^y f)(x, \cdot),$$

where the quadrature rules Q_1^x and Q_1^y are given by:

$$Q_1^x f)(\cdot, y) = \sum_{i=0}^{m} A_i (\lambda_i^x f)(\cdot, y),$$

respectively,

$$(Q_1^y f)(x, \cdot) = \sum_{j=0}^{m} B_j (\lambda_j^y f)(x, \cdot),$$

with R_1^x and R_1^y the corresponding remainder operators,

$$R_1^x = I^x - Q_1^x,$$
$$R_1^y = I^y - Q_1^y.$$

It is easy to check the following decompositions of the double integral operator I^{xy}:

$$I^{xy} = Q_1^x Q_1^y + (I^y R_1^x + I^x R_1^y - R_1^x R_1^y), \tag{3.2.1}$$

and

$$I^{xy} = (Q_1^x I^y + Q_1^y I^x - Q_1^x Q_1^y) + R_1^x R_1^y. \tag{3.2.2}$$

The identities (3.2.1) and (3.2.2) generate the product cubature formula:

$$I^{xy} f := Q_P f + R_P f = Q_1^x Q_1^y f + (R_1^x I^y + I^x Q_1^y - R_1^x R_1^y) f, \tag{3.2.3}$$

respectively, the boolean sum cubature formula:

$$I^{xy} f := Q_S f + R_S f = (Q_1^x I^y + Q_1^y I^x - Q_1^x Q_1^y) f + R_1^x R_1^y f. \tag{3.2.4}$$

For example, if $\lambda_i^x f = f(x_i, y),\ \lambda_j^y f = f(x, y_j),\ i = 0, 1, ..., m;\ j = 0, 1, ...n$ and Q_1^x, respectively Q_1^y, are the trapezoidal rules, i.e.,

$$(Q_1^x f)(\cdot, y) = \tfrac{b-a}{2}[f(a, y) + f(b, y)]$$
$$(Q_1^y f)(x, \cdot) = \tfrac{d-c}{2}[f(x, c) + f(x, d)]$$

then the formulas (3.2.3) and (3.2.4) become

$$\int_a^b \int_c^d f(x, y)dxdy = \tfrac{(b-a)(d-c)}{4}[f(a, c) + f(a, d) + f(b, c) + f(b, d)] + R_P(f), \tag{3.2.5}$$

with

$$R_P(f) = -\tfrac{(b-a)^3}{12} \int_c^d f^{(2,0)}(\xi_1, y)dy - \tfrac{(d-c)^3}{12} \int_a^b f^{(0,2)}(x, \eta_1)dx -$$
$$- \tfrac{(b-a)^3}{12} \tfrac{(d-c)^3}{12} f^{(2,2)}(\xi_2, \eta_2),$$

respectively,

$$\int_a^b \int_c^d f(x, y)dxdy = \tfrac{b-a}{2} \int_c^d [f(a, y) + f(b, y)]dy + \tfrac{d-c}{2} \int_a^b [f(x, c) + f(x, d)]dx \tag{3.2.6}$$
$$- \tfrac{(b-a)}{2} \tfrac{(d-c)}{2} [f(a, c) + f(a, d) + f(b, c) + f(b, d)] + R_S(f),$$

with

$$R_S(f) = -\frac{(b-a)^3}{12}\frac{(d-c)^3}{12} f^{(2,2)}(\xi, \eta),$$

where

$$\xi, \xi_1, \xi_2 \in [a, b] \text{ and } \eta, \eta_1, \eta_2 \in [c, d].$$

This way, there are obtained the trapezoidal product cubature formula (3.2.5) and the trapezoidal boolean-sum cubature formula (3.2.6).

From (3.2.3) and (3.2.4), it follows that

$$\text{ord}(Q_P) = \min\{\text{ord}(Q_1^x), \text{ord}(Q_1^y)\},$$

and

$$\text{ord}(Q_S) = \text{ord}(Q_1^x) + \text{ord}(Q_1^y).$$

Remark 3.2.1. The boolean-sum cubature operator has the remarkable property of having high approximation order, which can be seen as an optimality property. But, the boolean-sum formula contains the simple integrals $I^x f$ and $I^y f$. These simple integrals can be approximated, in a second level of approximation, using new quadrature operators, say Q_2^x and Q_2^y.

From (3.2.4) there is obtained

$$I^{xy} f = Qf + R_Q f, \tag{3.2.7}$$

with

$$Q = Q_1^x Q_2^y + Q_2^x Q_1^y - Q_1^x Q_1^y \tag{3.2.8}$$

and

$$R_Q = Q_1^x R_2^y + Q_1^y R_2^x + R_1^x R_1^y, \tag{3.2.9}$$

where

$$R_2^x = I^x - Q_2^x,$$
$$R_2^y = I^y - Q_2^y.$$

From (3.2.9) it follows that

$$\text{ord}(Q) = \min\{\text{ord}(Q_2^y) + 1,\ \text{ord}(Q_2^x) + 1,\ \text{ord}(Q_S)].$$

The quadrature operators Q_2^x and Q_2^y can be chosen in many ways. First of all, it depends on the given information of the function f.

Remark 3.2.2. A natural way to select them is such that the approximation order of the initial boolean-sum operators to be preserved. It is obvious that its approximation order cannot be increased.

Definition 3.2.3. *A cubature formula of the form (3.2.7), derived from the boolean-sum formula (3.2.4), which preserves the approximation order of* Q_S, $(\mathrm{ord}(Q_2^x)+1 \geq \mathrm{ord}(Q_S)$ *and* $\mathrm{ord}(Q_2^y)+1 \geq \mathrm{ord}(Q_S))$, *is called* a consistent cubature formula.

As the approximation order of the boolean-sum cubature operator Q_S cannot be increased, it is preferable to choose the quadrature operators Q_2^x and Q_2^y such that each term of the remainder of (3.2.7) to have the same approximation order.

Definition 3.2.4. *A cubature formula of the form (3.2.7), for which*

$$\mathrm{ord}(Q_2^x) = \mathrm{ord}(Q_2^y) = \mathrm{ord}(Q_s) - 1,$$

is called a homogeneous cubature formula.

Example 3.2.5. Let $D_h = [0,h] \times [0,h]$ and $f \in B_{2,2}(0,0)$. From the trapezoidal boolean-sum cubature formula (3.2.6) it can be obtained a homogenous cubature formula, if in the second level of approximation it is used the Simpson quadrature formula:

$$\int_0^h g(t)dt = \tfrac{h}{6}[g(0) + 4g\left(\tfrac{h}{2}\right) + g(h)] - \tfrac{h^5}{2880} g^{(4)}(\xi), \quad \xi \in [0,h],$$

where $g \in C^4[a,h]$.

One obtains the following trapezoidal-Simpson homogeneous formula:

$$\begin{aligned}\iint_{D_h} f(x,y)dxdy =& \tfrac{h^2}{4}\left\{-\tfrac{1}{3}[f(0,0)+f(0,h)+f(h,0)+f(h,h)]\right. \\ &\left. + \tfrac{4}{3}[f(0,\tfrac{h}{2})+f(h,\tfrac{h}{2})+f(\tfrac{h}{2},0)+f(\tfrac{h}{2},h)] + R(f),\right.\end{aligned}$$

where

$$R(f) = -\tfrac{h^5}{144}\left[\tfrac{1}{20} f^{(4,0)}(\xi_1,\eta_1) + \tfrac{1}{20} f^{(0,4)}(\xi_2,\eta_2) + f^{(2,2)}(\xi_3,\eta_3)\right],$$

with $(\xi_i,\eta_i) \in D_h,\ i = 1,2,3.$

Example 3.2.6. Also, from (3.2.6), using in the second level of approximation, the following quadrature formula:

$$\int_0^h g(t)dt = \tfrac{h}{2}[g(0)+g(h)] + \tfrac{h^2}{12}[g'(0)-g'(h)] + R(g),$$

with

$$R(g) = \tfrac{h^5}{720}g^{(4)}(\xi), \qquad \xi \in [0,h],\ g \in C^4[0,h],$$

it is obtained:

$$\begin{aligned}\iint_{D_h} f(x,y)dxdy =& \tfrac{h^2}{4}[f(0,0)+f(0,h)+f(h,0)+f(h,h)]+ \\ &+ \tfrac{h^3}{24}[(f^{(1,0)}+f^{(0,1)})(0,0)+(f^{(1,0)}-f^{(0,1)})(0,h) \\ &+ (f^{(0,1)}-f^{(1,0)})(h,0)-(f^{(1,0)}+f^{(0,1)})(h,h)] + R(f),\end{aligned} \tag{3.2.10}$$

with

$$R(f) = \tfrac{h^5}{144}[\tfrac{1}{5}f^{(4,0)}(\xi_1,\eta_1)+\tfrac{1}{5}f^{(0,4)}(\xi_2,\eta_2)-f^{(2,2)}(\xi_3,\eta_3)],$$

for $(\xi_i,\eta_i) \in D_h$, $i=1,2,3$. This is a homogenous cubature formula.

Remark 3.2.7. Formula (3.2.10) contains the same nodes as the initial one (3.2.6), while in the trapezoidal-Simpson formula (3.2.8) there appear some new nodes.

Example 3.2.8. Let

$$I^{xy}f = \int_0^h \int_0^h f(x,y)dxdy,$$

and

$$\begin{aligned}(Q_1^x f)(\cdot,y) &= hf(\tfrac{h}{2},y),\\ (Q_1^y f)(x_1\cdot) &= hf(x,\tfrac{h}{2}),\end{aligned}$$

be the gaussian quadrature rules. The suitable boolean-sum cubature formula is:

$$I^{xy}f = h\int_0^h f(\tfrac{h}{2},y)dy + h\int_0^h f(x,\tfrac{h}{2})dx - h^2 f(\tfrac{h}{2},\tfrac{h}{2}) + R_S(f), \tag{3.2.11}$$

where

$$R_S(f) = \tfrac{h^6}{576}f^{(2,2)}(\xi,\eta).$$

In order to get a homogeneous cubature formula in a second level of approximation we must use some quadrature rules Q_2^x, Q_2^y, with $\text{ord}(Q_2^x) = \text{ord}(Q_2^y) = 5$. Such quadrature rules can be

$$(Q_2^x f)(\cdot,\eta) = \tfrac{h}{2}[f(0,y) + f(h,y)] + \tfrac{h^2}{12}[f^{(1,0)}(0,y) - f^{(1,0)}(h,y)] + R_2^x(f),$$

with

$$(R_2^x f)(\cdot,y) = \tfrac{h^5}{720} f^{(4,0)}(\xi,y),$$

respectively,

$$(Q_2^y f)(x,\cdot) = \tfrac{h}{2}[f(x,0) + f(x,h)] + \tfrac{h^2}{12}[f^{(0,1)}(x,0) - f^{(0,1)}(x,h)] + R_2^y(f),$$

with

$$(R_2\xi f)(x,\cdot) = \tfrac{h^5}{720} f^{(0,4)}(x,\eta).$$

Theorem 3.2.9. *If $f \in B_{22}(0,0)$ then we have the following homogeneous cubature formula, derived from (3.2.11):*

$$\begin{aligned}\iint_{D_h} f(x,y)dxdy =& \tfrac{h^2}{2}[f(\tfrac{h}{2},0) + f(\tfrac{h}{2},h) + f(0,\tfrac{h}{2}) + f(h,\tfrac{h}{2}) - 2f(\tfrac{h}{2},\tfrac{h}{2})] \\ &+ \tfrac{h^3}{12}[f^{(1,0)}(0,\tfrac{h}{2}) - f^{(1,0)}(h,\tfrac{h}{2}) + f^{(0,1)}(\tfrac{h}{2},0) \\ &- f^{(0,1)}(\tfrac{h}{2},h)] + R(f),\end{aligned}$$

where

$$R(f) = \tfrac{h^6}{144}[\tfrac{1}{5} f^{(4,0)}(\tfrac{h}{2},\eta) + \tfrac{1}{5} f^{(0.4)}(\xi,\tfrac{h}{2}) + \tfrac{1}{4} f^{(2,2)}(\xi_1,\eta_1)].$$

Another optimality criterion for the cubature formulas is that regarding its number of nodes. The optimality problem is to find the cubature formula with the minimum number of nodes from a class of cubature formulas of the same degree of exactness.

The product and boolean-sum cubature formulas can be constructed applying the integral operator I^{xy} to both members of the product, respectively boolean-sum interpolation formulas, corresponding to the function f and the data $\lambda_i^x f,\ i = 0,1,...,m$ and $\lambda_j^y f,\ j = 0,1,...,n$. Moreover, if the operator I^{xy} is applied to a homogeneous interpolation formula it is obtained a homogeneous cubature formula. Next, we give some examples for a triangular domain. Let us consider the standard triangle:

$$T_h = \{(x,y) \in \mathbb{R}^2,\ x \geq 0,\ y \geq 0,\ x + y \leq h\}, \quad \text{for } h > 0.$$

Example 3.2.10. The operator

$$P = P_1 P_2 P_3,$$

with

$$\begin{aligned}(P_1 f(x,y) &= \tfrac{h-x-y}{h-y} f(0,y) + \tfrac{x}{h-y}(f(h-y,y),\\ (P_2 f)(x,y) &= \tfrac{h-x-y}{h-x} f(x,0) + \tfrac{y}{h-x} f(x,h-x),\\ (P_3 f)(x,y) &= \tfrac{x}{x+y} f(x+y,0) + \tfrac{y}{x+y} f(0,x+y),\end{aligned}$$

generates the homogenous interpolation formula:

$$f = Pf + Rf,$$

with

$$(Pf)(x,y) = \tfrac{h-x-y}{h} f(0,0) + \tfrac{x}{h} f(h,0) + \tfrac{y}{h} f(0,h),$$

and

$$\begin{aligned}(Rf)(x,y) = &\int_0^h \varphi_{20}(x,y;s) f^{(2,0)}(s,0)ds + \int_0^h \varphi_{02}(x,y;t) f^{(0,2)}(0,t)dt + \\ &+ \iint\limits_{T_h} \varphi_{11}(x,y;s,t) f^{(11)}(s,t)dsdt,\end{aligned}$$

for $f \in B_{11}(0,0)$,where φ_{ij} are the Peano's kernels. Then

$$\iint\limits_{T_h} f(x,y)dxdy = \iint\limits_{T_h} (Pf)(x,y)dxdy + \iint\limits_{T_h} (Rf)(x,y)dxdy$$

is a homogeneous cubature formula.

Indeed, we have

$$\iint\limits_{T_h} f(x,y)dxdy = \tfrac{h^2}{6}[f(0,0) + f(h,0) + f(0,h)] + R(f),$$

where

$$R(f) = \tfrac{h^4}{24}[f^{(2,0)}(\xi,0) + f^{(0,2)}(0,\eta) - f^{(1,1)}(\xi_1,\eta_1)],$$

for $\xi, \eta \in [0,h]$ and $(\xi_1,\eta_1) \in T_h$.

Chapter 4

Numerical linear systems

Many problems of applied mathematics reduce to a set of linear equations, or a linear system

$$A \cdot x = b,$$

with the matrix A and vector b given and the vector x to be determined, so the solving of a linear system is one of the principal problem of numerical analysis. An extensive set of algorithms have been developed for doing this, several of them being presented in the sequel.

4.1 Triangular systems of equations

Let us consider the system of equations

$$A \cdot x = b, \tag{4.1.1}$$

where $A \in \mathcal{M}_{n\times n}(\mathbb{R})$ is a lower triangular matrix of the form

$$\begin{pmatrix} a_{11} & 0 & ... & 0 \\ a_{21} & a_{22} & ... & 0 \\ \vdots & & & \\ a_{n1} & a_{n2} & ... & a_{nn} \end{pmatrix},$$

$x, b \in \mathcal{M}_{n\times 1}(\mathbb{R})$, and $\prod_{i=1}^{n} a_{ii} \neq 0$.

Then, the solution of (4.1.1) is obtained by forward substitution, according with the relations:

$$x_i = \left(b_i - \sum_{k=1}^{i-1} a_{ik} x_k \right) \Big/ a_{ii}, \quad i = \overline{1,n}. \tag{4.1.2}$$

Let us consider the system

$$A \cdot x = b, \tag{4.1.3}$$

with $x, b \in \mathcal{M}_{n\times 1}(\mathbb{R})$, and $A \in \mathcal{M}_{n\times n}(\mathbb{R})$ an upper triangular matrix of the form

$$\begin{pmatrix} a_{11} & a_{12} & ... & a_{1n} \\ 0 & a_{22} & ... & a_{2n} \\ \vdots & & & \\ 0 & 0 & ... & a_{nn} \end{pmatrix},$$

with $\prod_{i=1}^{n} a_{ii} \neq 0$.

Then, the solution of (4.1.3) is obtained by backward substitution, according with the relations:

$$x_i = \left(b_i - \sum_{k=i+1}^{n} a_{ik} x_k \right) \Big/ a_{ii}, \quad i = \overline{1,n}.$$

4.1.1 The Blocks version

If we consider the system (4.1.1) and x_1 has been found, then, after substitution of x_1 into the equations from the second to the n-th, we obtain a new $(n-1)\times(n-1)$ lower triangular system

$$\begin{pmatrix} a_{22} & 0 & \dots & 0 \\ a_{32} & a_{33} & \dots & 0 \\ \vdots & & & \\ a_{n2} & a_{n3} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_2 - a_{21}x_1 \\ b_3 - a_{32}x_1 \\ \vdots \\ b_n - a_{n1}x_1 \end{pmatrix}.$$

Continuing in this way, we obtain the solution of system (4.1.1).

Note 1. In the same way, we may consider the column version. In that situation, matrix A is considered upper triangular, and by substituting

x_n into the equations from the first to the $(n-1)$-th, we obtain a new $(n-1)\times(n-1)$ upper triangular system.

Note 2. The block version of the triangular systems is more appropriate to the parallel calculus.

4.2 Direct methods for solving linear systems of equations

Let us consider a system of linear equations

$$A \cdot x = b, \tag{4.2.1}$$

where $A \in \mathcal{M}_{n\times n}(\mathbb{R})$, $x, b \in {}_{n\times 1}(\mathbb{R})$.

Here A is any matrix of the form

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & & & \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}.$$

The direct methods for solving such systems give the exact solution, let it be $x^* \in \mathcal{M}_{n\times 1}(\mathbb{R})$.

In the literature, many direct methods for solving systems of type (4.2.1) are known. The Cramer method is one of it. It is a performant method, from computational point of view, for systems of small dimension. For larger systems, there are other methods, more appropriate, which generate efficient algorithms, both for serial and for parallel computers. Generally, all these methods are based on some factorization techniques, as can be seen in what follows.

4.2.1 Factorization techniques

These type of methods are based on the idea of converting A into products of the form LU or LDU, where L is zero above the main diagonal, U is zero below it and D has only diagonal elements different from zero. If L or U has all diagonal elements equal to 1, it is called unit triangular. Several methods, as Gauss, Gauss-Jordan, Doolittle, Crout, Cholesky, etc. produce factorization.

4.2.1.1 The Gauss method

The Gauss method, known also as the method of partial elimination, is one of the oldest algorithm and still one of the most popular. Many authors present it. We remind here the main characteristics of this method, in principal due to the fact that it generates a factorization of the matrix A.

Let write the system (4.2.1) in the detailed form:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n. \end{cases} \tag{4.2.2}$$

By Gauss method, the system (4.2.2) can be solved in two stages.

The first stage. Replacing the equations by combinations of equations, (by using algebraic elementary transformations on the extended matrix $\overline{A}$), the following triangular system is obtained:

$$\begin{cases} a_{11}^{(1)}x_1 + a_{12}^{(1)}x_2 + \cdots + a_{1n}^{(1)}x_n = b_1^{(1)} \\ a_{21}^{(2)}x_2 + \cdots + a_{2n}^{(2)}x_n = b_2^{(2)} \\ \vdots \\ a_{nn}^{(n)}x_n = b_n^{(n)} \end{cases} \tag{4.2.3}$$

The elements of the new matrix are given by the relations:

$$a_{ij}^{(1)} = a_{ij}, \quad b_i^{(1)} = b_i, \quad i,j = \overline{1,n}$$

and

$$\begin{aligned} a_{ij}^{(p+1)} &= a_{ij}^{(p)} - a_{pj}^{(p)} \cdot \frac{a_{ip}^{(p)}}{a_{pp}^{(p)}} \\ b_i^{(p+1)} &= b_i^{(p)} - b_p^{(p)} \cdot \frac{a_{ip}^{(p)}}{a_{pp}^{(p)}} \end{aligned} \qquad p = \overline{1,n-1},\ i,j = \overline{p+1,n},$$

for $a_{pp}^{(p)} \neq 0$, $p = \overline{1,n}$.

The second stage. The components of the vector x are easily found, one after another, by a process called back-substitution. So, the last equation determines x_n, which is then substituted in the next-to-last equation to get x_{n-1}, and so on.

The relations are:

$$\begin{aligned} x_n &= \frac{b_n^{(n)}}{a_{nn}^{(n)}} \\ x_p &= \frac{1}{a_{pp}^{(p)}} \left(b_p^{(p)} - \sum_{j=p+1}^{n} a_{pj}^{(p)} x_j \right) \end{aligned} . \tag{4.2.4}$$

The process of elimination can be described more accurate by means of a sequence of matrices. If we denote $b_p^{(p)}$ by $a_{p,n+1}^{(p)}$, for $p = \overline{1,n}$, and use the extended matrix of the system (4.2.3),

$$\bar{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} & a_{1,n+1} \\ 0 & a_{22} & & a_{2n} & a_{2,n+1} \\ \vdots & & & & \\ 0 & 0 & 0 & a_{nn} & a_{n,n+1} \end{pmatrix}, \tag{4.2.5}$$

we get the following sequence of matrices: $\overline{A}^{(1)}, \overline{A}^{(2)}, \dots, \overline{A}^{(n)}$, where $\overline{A}^{(1)}$ is matrix $\overline{A}$ given in (4.2.5) and $\overline{A}^{(p)}$, for every $p = \overline{2,n}$ contains the elements:

$$a_{ij}^{(p)} = \begin{cases} a_{ij}^{(p-1)}, & \text{when } i = \overline{1,p-1} \text{ and } j = \overline{1,n+1} \\ 0, & \text{when } i = \overline{p,n} \text{ and } j = \overline{1,p-1} \\ a_{ij}^{(p-1)} - \dfrac{a_{i,p-1}^{(p-1)}}{a_{p-1,p-1}^{(p-1)}} \cdot a_{p-1,j}^{(p-1)}, & \text{when } i = \overline{1,p-1} \text{ and } j = \overline{1,n+1}. \end{cases}$$

Remark 4.2.1. The Gauss method yields a factorization of matrix A in the form

$$A = L \cdot U,$$

where U is the upper triangular matrix shown in (4.2.3) and L is a lower triangular one with 1 on the diagonal.

Remark 4.2.2. If A is nonsingular, then condition (4.2.4) can always be fulfilled. So, by using the technique of pivoting, even if one $a_{pp}^{(p)} = 0$, we look for

$$a_{i_p j_p}^{(p)} = \max\{|a_{ij}^{(p)}| \mid i, j = \overline{p,n}\}$$

Then, replacing the rows p and i_p, respectively the columns p and j_p, the element $a_{i_p j_p}^{(p)}$ will take the place of $a_{pp}^{(p)}$. Obviously, $a_{i_p j_p}^{(p)} \neq 0$.

Remark 4.2.3. On opposite, if A is singular, then $Ax = b$ will have either no solution at all or infinitely many solutions. In fact, the Gauss method can be used to prove the fundamental theorem of algebra, which deals with the question of whether or not a solution exists.

4.2.1.2 The Gauss-Jordan method

The Gauss-Jordan method, known also as "the method of complete elimination", is based in fact on the same idea of computation as in the Gauss method, with the only difference that in the process of elimination, the system matrix will become a diagonal matrix, so the system (4.2.1) will be, finally, of the form:

$$\begin{cases} a_{11}^{(1)} x_1 & & & & = b_1^{(1)} \\ & a_{22}^{(2)} x_2 & & & = b_2^{(2)} \\ & & \ddots & & \vdots \\ & & & a_{nn}^{(n)} x_n & = b_n^{(n)}. \end{cases}$$

Then the components of solution x^* will be:

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

Obviously, all $a_{ii}^{(i)} \neq 0$.

4.2.1.3 The LU methods

Under certain conditions, the system matrix A of (4.2.1) can be expressed in the form of a product of a lower triangular matrix L and an upper triangular matrix U and, as the result, we have to solve two systems with triangular matrices.

So,

$$A = LU,$$

then the system (4.2.1) becomes:

$$LUx = b. \tag{4.2.6}$$

which can be solved in two steps:

Step 1. Solve $L \cdot y = b$.
Step 2. Solve $U \cdot x = y$.
The matrix L will have the form:

$$L = \begin{pmatrix} l_{11} & 0 & ... & 0 \\ l_{21} & l_{22} & ... & 0 \\ \vdots & & & \\ l_{n1} & l_{n2} & & l_{nn} \end{pmatrix}$$

and the matrix U will have the form:

$$U = \begin{pmatrix} u_{11} & u_{12} & ... & u_{1n} \\ 0 & u_{22} & ... & u_{2n} \\ \vdots & & & \\ 0 & 0 & & u_{nn} \end{pmatrix}.$$

The LU decompositions are basically modified forms of Gaussian elimination. The way in which the entries are computed is given in what follows, known as the Doolittle algorithm.

4.2.1.4 The Doolittle algorithm

This algorithm does the elimination column by column starting from the left, by multiplying A to the left with lower triangular matrices. It results in a **unit lower triangular** matrix and an **upper triangular matrix**. More precisely, for system (4.2.1), with $a_{kk} \neq 0$, $k = \overline{1, n}$ we denote

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

$$t^{(k)} = [\underbrace{0 \dots 0}_{k\ zeros} l_{k+1,k} \dots l_{n,k}]$$

and

$$M_k = I - t^{(k)} e_k^T, \tag{4.2.7}$$

where

$$e_k = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix}$$

is the k-unit vector.

Then

$$M_k x = \begin{pmatrix} 1 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 1 & 0 & \dots & 0 \\ 0 & \dots & l_{k+1,k} & 1 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & l_{n,k} & 0 & \dots & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_k \\ x_{k+1} \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} x_1 \\ \vdots \\ x_k \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Definition 4.2.4. *A matrix M_k of the form (4.2.7) is called a* Gauss matrix*, the components $l_{i,k}$, $i = \overline{k+1, n}$ are called* Gauss multipliers *and the vector $t^{(k)}$ is called the* Gauss vector*. The transformation defined with the Gauss matrix M_k is called the* Gauss transformation*.*

Note 1. If $A \in \mathcal{M}_{n\times n}(\mathbb{R})$ then the Gauss matrices $M_1, \dots, M_{n-1}$ can be found such that

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

is upper triangular.

Moreover, if we choose

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

then

$$A = L \cdot U$$

and so we obtain our factorization.

Example 4.2.5. Find the LU factorization of the matrix

$$A = \begin{pmatrix} 2 & 1 \\ 8 & 7 \end{pmatrix}$$

using the Doolittle algorithm.

Solution. Find the Gauss matrix M_1 for A:

$$M_1 = I - t^{(1)} e_1^T = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 \\ \frac{8}{2} \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & 0 \\ 4 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -4 & 1 \end{pmatrix}.$$

Thus

$$M_1 \cdot A = \begin{pmatrix} 1 & 0 \\ -4 & 1 \end{pmatrix} \cdot \begin{pmatrix} 2 & 1 \\ 8 & 7 \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 0 & 3 \end{pmatrix} = U$$

$$M_1^{-1} = \begin{pmatrix} 1 & 0 \\ 4 & 1 \end{pmatrix} = L.$$

So

$$A = \begin{pmatrix} 2 & 1 \\ 8 & 7 \end{pmatrix} = L \cdot U = \begin{pmatrix} 1 & 0 \\ 4 & 1 \end{pmatrix} \cdot \begin{pmatrix} 2 & 1 \\ 0 & 3 \end{pmatrix}.$$

Note 2. There is another algorithm, called **the Crout Algorithm** which is slightly different. It constructs a lower triangular matrix and an **unit upper triangular** matrix.

4.2.1.5 Block LU decomposition

The Block LU decomposition generates a lower block triangular matrix L and an upper block triangular matrix U. This decomposition is used in order to reduce the complexity computation and is appropriate to parallel calculus.

Consider a block matrix:

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} I \\ CA^{-1} \end{pmatrix} \cdot A \cdot \begin{pmatrix} I & A^{-1}B \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & D - CA^{-1}B \end{pmatrix},$$

when matrix A is assumed to be non-singular, I is an identity matrix with proper dimension, and O is a null matrix.

We can also rewrite the above equation using the half matrices:

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} A^{\frac{1}{2}} \\ CA^{-\frac{1}{2}} \end{pmatrix} \begin{pmatrix} A^{\frac{*}{2}} & A^{-\frac{1}{2}}B \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & Q^{\frac{1}{2}} \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & Q^{\frac{*}{2}} \end{pmatrix},$$

where $Q = D - CA^{-1}B$ is called the **Schur complement** of A, and the half matrices can be calculated by means of Cholesky decomposition (see section 4.2.1.6).

The half matrices satisfy that

$$A^{\frac{1}{2}} \cdot A^{\frac{*}{2}} = A, \quad A^{\frac{1}{2}} \cdot A^{-\frac{1}{2}} = I, \quad A^{-\frac{*}{2}} \cdot A^{\frac{*}{2}} = I,$$

$$Q^{\frac{1}{2}} Q^{\frac{*}{2}} = Q.$$

Thus, we have

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} = LU,$$

where

$$LU = \begin{pmatrix} A^{\frac{1}{2}} & 0 \\ CA^{-\frac{*}{2}} & 0 \end{pmatrix} \begin{pmatrix} A^{\frac{*}{2}} & A^{-\frac{1}{2}}B \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & Q^{\frac{1}{2}} \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & Q^{\frac{1}{2}} \end{pmatrix}.$$

The matrix LU can be decomposed in an algebraic manner into

$$L = \begin{pmatrix} A^{\frac{1}{2}} & 0 \\ CA^{-\frac{*}{2}} & Q^{\frac{1}{2}} \end{pmatrix} \quad \text{and} \quad U = \begin{pmatrix} A^{\frac{*}{2}} & A^{-\frac{1}{2}}B \\ 0 & Q^{\frac{*}{2}} \end{pmatrix}.$$

4.2.1.6 The Cholesky decomposition

The Cholesky decomposition is a matrix decomposition of a symmetric positive-definite matrix into a lower triangular matrix and the transpose of the lower triangular matrix. The lower triangular matrix is the **Cholesky triangle** of the original, positive definite, matrix.

As we saw in the previous section, any square matrix A can be written as the product of a lower triangular matrix L and an upper triangular matrix U. However, if A is symmetric and positive definite, we can choose the factors such that U is the transpose of L.

Remark 4.2.6. When is applicable, the Cholesky decomposition is twice as efficient as the LU decomposition.

Let's consider the system

$$A \cdot x = b, \tag{4.2.8}$$

where $A \in \mathcal{M}_{n\times n}(\mathbb{R})$ is symmetric ($A = A^T$) and positive definite ($x^T Ax > 0,\ \forall\ x \in \mathcal{M}_{n\times 1}(\mathbb{R})$). Also, $b \in \mathcal{M}_{n\times 1}(\mathbb{R})$.

In order to solve (4.2.8), first we have to compute the Cholesky decomposition

$$A = L \cdot L^T,$$

then we solve

$$L \cdot y = b$$

to get $y \in \mathcal{M}_{n\times 1}(\mathbb{R})$ and

$$L^T \cdot x = y$$

to get x, the solution of the initial system.

The Cholesky algorithm

It is a modifed version of the Gauss recursive algorithm.

Step 1. $i := 1,\ A^{(1)} := A$

For $i := 1$ to n do

Step i.

$$A^{(i)} = \begin{pmatrix} I_{i-1} & 0 & 0 \\ 0 & a_{ii} & b_i^T \\ 0 & b_i & B^{(i)} \end{pmatrix},$$

where I_{i-1} denotes the identity matrix of dimension $i-1$

$$L_i := \begin{pmatrix} I_{i-1} & 0 & 0 \\ 0 & \sqrt{a_{ii}} & 0 \\ 0 & \dfrac{1}{\sqrt{a_{ii}}} b_i & I_{n-i} \end{pmatrix}.$$

So,

$$A^{(i)} = L_i \cdot A^{(i+1)} \cdot L_i^T,$$

where

$$A^{(i+1)} = \begin{pmatrix} I_{i-1} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & B^{(i)} - \dfrac{1}{a_{ii}} b_i b_i^T \end{pmatrix}.$$

After n steps, we get $A^{(n+1)} = I$. Hence, the lower triangular matrix L we are looking for is calculated as

$$L := L_1 L_2 \dots L_n.$$

Remark 4.2.7. The following formula for the entries of L holds

$$l_{11} = \sqrt{a_{11}}$$
$$l_{i1} = a_{i1}/l_{11},\ i = \overline{2,n}$$
$$l_{ii} = \left[a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2 \right]^{1/2},\ i = \overline{2,n}$$
$$l_{ij} = \frac{1}{l_{jj}} \left[a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk} \right],\ i = \overline{3,n},\ j = \overline{2,i-1}.$$

Remark 4.2.8. The expression under the square root is always positive as A is real and positive definite.

4.2.1.7 The QR decomposition

The QR decomposition of a matrix is a method which permits the writing of matrix A into a product of an orthogonal matrix and a triangular one.

So, let's consider the system of linear equations

$$A \cdot x = b, \tag{4.2.9}$$

where $A \in \mathcal{M}_{n\times n}(\mathbb{R})$, $x, b \in \mathcal{M}_{n\times 1}(\mathbb{R})$.

According with the QR decomposition, matrix A can be written

$$A = Q \cdot R,$$

where Q is an orthogonal matrix (meaning that $Q^T \cdot Q = I$) and R is an upper triangular matrix.

Remark 4.2.9. This factorization is unique if we require that the diagonal elements of R to be positive.

There are several methods for computing the QR decomposition, such as: by means of Givens rotations, Householder transformation or the Gram-Schmidt decomposition.

Computing QR by means of Gram-Schmidt decomposition. According with the Gram-Schmidt method, the matrix A can be written as columns:

$$A = \left(\begin{array}{c|c|c} a_1 & \dots & a_n \end{array} \right).$$

Then, denoting by $\langle \cdot, \cdot \rangle$ the inner vectorial product, we have

$$\begin{aligned}
&u_1 = a_1, && e_1 = \frac{u_1}{\|u_1\|} \\
&u_2 = a_2 - \langle e_1, a_2 \rangle, && e_2 = \frac{u_2}{\|u_2\|} \\
&\vdots \\
&u_k = a_k - \sum_{j=1}^{k-1} \langle e_j, e_k \rangle, && a_k = \frac{u_k}{\|u_k\|}
\end{aligned}$$

or, rearranging the terms,

$$\begin{aligned} a_1 &= e_1\|u_1\| \\ a_2 &= \langle e_1, a_2\rangle + e\|u_2\| \\ a_3 &= \langle e_1, a_3\rangle + \langle e_2, a_3\rangle + e_3\|u_3\| \\ &\vdots \\ a_k &= \sum_{j=1}^{k-1}\langle e_j, a_k\rangle + e_k\|u_k\|. \end{aligned}$$

In the matrix form, these equations can be written:

$$\left(\begin{array}{c|c|c|c} e_1 & \dots & e_n \end{array} \right) \begin{pmatrix} \|u_1\| & \langle e_1, a_2\rangle & \langle e_1, a_3\rangle & \dots \\ 0 & \|u_2\| & \langle e_2, a_3\rangle & \dots \\ 0 & 0 & \|u_3\| & \dots \\ \vdots & \vdots & \vdots & \end{pmatrix}.$$

But the product of each row and column of the matrices above gives us a respective column of A that we started with. So, the matrix Q is the matrix of $e_k s$:

$$Q = \left(\begin{array}{c|c|c} e_1 & \dots & e_n \end{array} \right).$$

Then,

$$R = Q^T \cdot A = \begin{pmatrix} \langle e_1, a_1\rangle & \langle e_1, a_2\rangle & \langle e_1, a_3\rangle & \dots \\ 0 & \langle e_2, a_2\rangle & \langle e_2, a_3\rangle & \dots \\ 0 & 0 & \langle e_3, a_3\rangle & \dots \\ \vdots & \vdots & \vdots & \end{pmatrix}.$$

Note that $\langle e_j, a_j\rangle = \|u_j\|$, $\langle e_j, a_k\rangle = 0$ for $j > k$ and $QQ^T = I$, so $Q^T = Q^{-1}$.

Example 4.2.10. Let us consider the decomposition of

$$A = \begin{pmatrix} 12 & -51 & 4 \\ 6 & 167 & -68 \\ -4 & 24 & -41 \end{pmatrix}.$$

We compute the matrix Q by means of Gram-Schmidt method, as follows:

$$U = \left(\begin{array}{ccc} u_1 & u_2 & u_3 \end{array} \right) \begin{pmatrix} 12 & -69 & -58 \\ 6 & 158 & 6 \\ -4 & 30 & -165 \end{pmatrix},$$

$$Q = \begin{pmatrix} \frac{u_1}{\|u_1\|} & \frac{u_2}{\|u_2\|} & \frac{u_3}{\|u_3\|} \end{pmatrix} = \begin{pmatrix} 6/7 & -69/175 & -58/175 \\ 3/7 & 158/175 & 6/175 \\ -2/7 & 6/35 & -33/35 \end{pmatrix}.$$

Thus, we have

$$A = \underbrace{Q \cdot Q^T}_{I} \cdot A = Q \cdot R,$$

so

$$R = Q^T \cdot A = \begin{pmatrix} 14 & 21 & -14 \\ 0 & 175 & -70 \\ 0 & 0 & 35 \end{pmatrix}.$$

Computing QR by means of Householder reflections. A Householder reflection (or Householder transformation) is a transformation that takes a vector and "reflects" it about some plane. We can use this property to calculate the QR factorization of a matrix.

Q can be used to reflect a vector in such a way that all coordinates but one disappear.

Let x be an arbitrary m-dimensional column vector such that $\|x\| = |\alpha|$ for a scalar α.

Then, for $e_1 = (1, 0, \ldots, 0)^T$ and $\|\cdot\|$ the Euclidean norm, set

$$\begin{aligned} u &= x - \alpha e_1, \\ v &= \frac{u}{\|u\|}, \\ Q &= I - 2v \cdot v^T, \end{aligned}$$

Q is a Householder matrix and

$$Qx = (\alpha, 0, \ldots, 0)^T.$$

This can be used to gradually transfer an m-by-n matrix A to upper triangular form. First, we multiply A with the Householder matrix Q_1 that we obtain when we choose the first matrix column for x. This results in a matrix QA with zeros in the left column (except for the first row)

$$Q_1 A = \begin{pmatrix} \alpha_1 & * & \ldots & * \\ 0 & & & \\ \vdots & & A' & \\ 0 & & & \end{pmatrix}.$$

This can be repeated for A' (obtained from A by deleting the first row and first column) resulting in a Householder matrix Q_2'. Note that Q_2' is smaller than Q_1. Since we want it really to operate on $Q_1 A$ instead of A' we need to expand it to the upper left, filling in a 1, or in general:

$$Q_k = \begin{pmatrix} I_{k-1} & 0 \\ 0 & Q_k' \end{pmatrix}.$$

After t iterations of the process, $t = \min(m-1, n)$,

$$R = Q_t \dots Q_2 Q_1 A$$

is an upper triangular matrix. So, with

$$Q = Q_1 Q_2 \dots Q_t,$$

$$A = QR,$$

so we have needed QR decomposition of A.

Example 4.2.11. Let us consider the matrix A like in Example 4.2.10:

$$A = \begin{pmatrix} 12 & -51 & 4 \\ 6 & 167 & -68 \\ -4 & 24 & -41 \end{pmatrix}.$$

We decompose $A = Q \cdot R$ by means of the Householder transformation.

First, we need to find a reflection that transforms the first column of matrix A, vector $a_1 = (\ 12 \quad 6 \quad -4\ )^T$, to $\|a_1\|\, e_1 = (\ 14 \quad 0 \quad 0\ )^T$.

Now, for $\alpha = 14$ and $x = a_1 = (\ 12 \quad 6 \quad -4\ )^T$, we compute

$$u = x - \alpha e_1 = \begin{pmatrix} -2 & 6 & -4 \end{pmatrix}^T$$

and

$$v = \frac{u}{\|u\|} = \frac{1}{\sqrt{14}} \begin{pmatrix} -1 & 3 & -2 \end{pmatrix}^T.$$

Then,

$$\begin{aligned} Q_1 &= I - \frac{2}{\sqrt{14} \cdot \sqrt{14}} \begin{pmatrix} -1 \\ 3 \\ -2 \end{pmatrix} \begin{pmatrix} -1 & 3 & -2 \end{pmatrix} \\ &= I - \frac{1}{7} \begin{pmatrix} 1 & -3 & 2 \\ -3 & 9 & -6 \\ 2 & -6 & 4 \end{pmatrix} = \begin{pmatrix} 6/7 & 3/7 & -2/7 \\ 3/7 & -2/7 & 6/7 \\ -2/7 & 6/7 & 3/7 \end{pmatrix}. \end{aligned}$$

Now observe:

$$Q_1 \cdot A = \begin{pmatrix} 14 & 21 & -14 \\ 0 & -49 & -14 \\ 0 & 168 & -77 \end{pmatrix}.$$

Further, we need to make zero the $(3,2)$ entry. We apply the same process to

$$A' = M_{11} = \begin{pmatrix} -49 & -14 \\ 168 & -77 \end{pmatrix}.$$

Finally, we get the matrix of the Householder transformation:

$$Q_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -7/25 & 24/25 \\ 0 & 24/25 & 7/25 \end{pmatrix}.$$

We find

$$Q = Q_1 Q_2 = \begin{pmatrix} 6/7 & -69/175 & 58/175 \\ 3/7 & 158/175 & -6/175 \\ -2/7 & 6/35 & 33/35 \end{pmatrix}$$

$$R = Q^T \cdot A = \begin{pmatrix} 14 & 21 & -14 \\ 0 & 175 & -70 \\ 0 & 0 & -35 \end{pmatrix}.$$

The matrix Q is orthogonal and R is upper triangular, so $A = QR$ is the required QR-decomposition.

Computing QR by means of Givens rotations. The QR decomposition can also be computed with a series of Givens rotations, where a Givens rotation is defined by

$$R(\theta) := \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}$$

and rotates the vector $x \in \mathbb{R}^2$ with an angle θ. Applying such a rotation to matrix A in the system, this will make zeros an element in the sub-diagonal of the matrix. The concaternation of all the given rotations forms the orthogonal Q matrix.

Example 4.2.12. Let us consider the same previous example

$$A = \begin{pmatrix} 12 & -51 & 4 \\ 6 & 167 & -68 \\ -4 & 24 & -41 \end{pmatrix}.$$

First, we need to form a rotation matrix that will zero the element $a_{31} = -4$. We get this matrix using the Givens rotation method, and call the matrix G_1. We rotate the vector $(6, -4)$, using the angle $\theta = \arctan\left(-\frac{4}{6}\right)$. Then,

$$G_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0,83 & -0,55 \\ 0 & 0,55 & 0,83 \end{pmatrix}$$

and

$$G_1 A = \begin{pmatrix} 12 & -51 & 4 \\ 7,21 & 125,6 & -33,83 \\ 0 & 112,6 & -71,83 \end{pmatrix}.$$

The matrix G_2 and G_3 will make zeros the sub-diagonal elements a_{21} and a_{32}, forming a rectangular matrix R. The orthogonal matrix Q^T is formed from the concatenation of all the given matrices:

$$Q^T = G_3 G_2 G_1.$$

Thus, we have $G_3 G_2 G_1 A = Q^T A = R$ and the QR-decomposition is

$$A = Q \cdot R.$$

Remark 4.2.13. The Givens method for a QR decomposition is more appropriate to parallelization than the previous methods.

4.2.1.8 The WZ decomposition

Let us consider the system of linear equations

$$A \cdot x = b$$

where $A \in \mathcal{M}_{n\times n}(\mathbb{R})$, $x, b \in \mathcal{M}_{n\times 1}(\mathbb{R})$.

The WZ decomposition will decompose matrix A, dense and nonsingular, into a product of matrices W and Z, consisting of following columns w_i and

rows z_i, i.e.:

$$w_i = (\underbrace{0\dots01}_{i} w_{i+1,i}\dots w_{n-i,i}0\dots0)^T,\ i=\overline{1,m},$$
$$w_i = \underbrace{0\dots01}_{i}0\dots0)^T,\ i=p,q$$
$$w_i = (\underbrace{0\dots0}_{n-i+1} w_{n-i+2,i}\dots w_{i-1,i}10\dots0)^T,\ i=\overline{q+1,n},$$
$$z_i = (\underbrace{0\dots0}_{i-1} z_{ii}\dots z_{i,n-i+1}0\dots0),\ i=\overline{1,p}$$
$$z_i = (0\dots0 z_{i,n-i+1}\dots z_{ii}0\dots0),\ i=\overline{p+1,n}$$

where

$$m=[(n-1)/2],\quad p=[(n+1)/2],\quad q=[(n+1)/2].$$

For example, for $n=5$, we have:

$$W=\begin{pmatrix} 1 & 0 & 0 & 0 & 0\\ w_{21} & 1 & 0 & 0 & w_{25}\\ w_{31} & w_{32} & 1 & w_{34} & w_{35}\\ w_{41} & 0 & 0 & 1 & w_{45}\\ 0 & 0 & 0 & 0 & 1 \end{pmatrix},\quad Z=\begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} & z_{15}\\ 0 & z_{22} & z_{23} & z_{24} & 0\\ 0 & 0 & z_{33} & 0 & 0\\ 0 & z_{42} & z_{43} & z_{44} & 0\\ z_{51} & z_{52} & z_{53} & z_{54} & z_{55} \end{pmatrix}.$$

After factorization, we can solve the two linear systems:

$$Wc=b$$

and

$$Zx=c,$$

instead of one.

Computing the WZ-decomposition. The WZ algorithm consists of two parts: reduction of the matrix A (and the vector b) to the matrix Z (and the vector c), and next, solving equation $Z\cdot x=c$.

Reduction of matrix A

Step 1. We zero the elements from the 2^{nd} to $n-1^{st}$ rows in the 1^{st} and n^{th} columns. Finally, we can write this as follows:

Step 1.1. We compute $\overline{w}_{i1}$ and $\overline{w}_{in}$ from the linear system

$$\begin{cases} a_{11}\overline{w}_{i1}+a_{n1}\overline{w}_{in}=-a_{i1}\\ a_{1n}\overline{w}_{i1}+a_{nn}\overline{w}_{in}=-a_{in},\ \text{for } i=\overline{2,n-1} \end{cases}$$

and we get the matrix:

$$W^{(1)} = \begin{pmatrix} 1 & & & 0 \\ \overline{w}_{21} & 1 & & \overline{w}_{2n} \\ \vdots & & \ddots & \vdots \\ \overline{w}_{n-1,1} & & 1 & \overline{w}_{n-1,n} \\ 0 & & & 1 \end{pmatrix}.$$

Step 1.2. Compute

$$A^{(1)} = W^{(1)}A, \quad b^{(1)} = W^{(1)}b.$$

So, we get the linear system $A^{(1)}x = b^{(1)}$, where

$$A^{(1)} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1,n-1} & a_{1n} \\ 0 & a_{22}^{(1)} & \dots & a_{2,n-1}^{(1)} & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & a_{n-1,2}^{(1)} & \dots & a_{n-1,n-1}^{(1)} & 0 \\ a_{n1} & a_{n2} & \dots & a_{n,n-1} & a_{nn} \end{pmatrix}, \quad b^{(1)} = \begin{pmatrix} b_1 \\ b_2^{(1)} \\ \vdots \\ b_{n-1}^{(1)} \\ b_n \end{pmatrix}$$

and

$$a_{ij}^{(1)} = a_{ij} + \overline{w}_{i1}a_{1j} + \overline{w}_{in}a_{nj}, \quad j = \overline{2, n-1},\ i = \overline{2, n-1},$$

$$b_i^{(1)} = b_i + \overline{w}_{i1}b_1 + \overline{w}_{in}b_n, \quad j = \overline{1, n-1}.$$

Similarly, we carry out the second step (and the next steps) which consists in the same operations as before, but only on the submatrices of $A^{(1)}$ obtained by deleting the first and the last rows and columns of the matrix $A^{(1)}$.

After m such steps we get the matrix $Z = A^{(m)}$ (as defined in (4.2.1.8)) and the vector $c = b^{(m)}$. We have

$$W^{(m)} \dots W^{(1)}A = Z,$$

so

$$A = (W^{(1)})^{-1} \dots (W^{(m)})^{-1}Z = WZ.$$

Step 2. The second part of the method is to solve the linear system $Zx = b^{(m)}$, which consists in solving a linear system with two unknown quantities x_p and x_q and next updating the vector b. Formally, we have:

Step 2.1. We find x_p and x_q from the system

$$\begin{cases} z_{pp}x_p + z_{pq}x_q = b_p^{(m)} \\ z_{qp}x_p + z_{qq}x_q = b_q^{(m)}. \end{cases} \tag{4.2.10}$$

Step 2.2. Compute

$$c_i^{(1)} = c_i - z_{ip}x_p - z_{iq}x_q, \quad i = 1, \ldots, p-1, q+1, \ldots, n.$$

For the odd n, system (4.2.10) consists (in the last step) of one equation. Similarly, we make the next steps for the inner two-equations system. There are $m+1$ such steps.

Remark 4.2.14. The WZ-decomposition can be very well parallelized, as we shall see in Chapter 7.

4.3 Iterative methods

The iterative methods for solving a linear system of equations of the form

$$A \cdot x = b, \tag{4.3.1}$$

with $A \in \mathcal{M}_{n\times n}(\mathbb{R})$, $x, b \in {}_{n\times 1}(\mathbb{R})$, generate an approximation of the solution vector x, by means of a sequence of successive approximations, in case of the convergence of the method. So, in order to solve iteratively a system of equations, the following steps have to be performed:

Step 1. Determine the convergence of the iterative method.

In order to do this, we decompose matrix A of the system (4.3.1) in a sum of three matrices:

$$A = L + D + U, \tag{4.3.2}$$

where

$$L = \begin{pmatrix} 0 & 0 & \ldots & 0 \\ a_{21} & 0 & \ldots & 0 \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \ldots & 0 \end{pmatrix}, \quad U = \begin{pmatrix} 0 & a_{12} & \ldots & a_{1n} \\ 0 & 0 & \ldots & a_{2n} \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \ldots & 0 \end{pmatrix}$$

and

$$D = \begin{pmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \vdots & & & \\ 0 & 0 & \dots & a_{nn} \end{pmatrix}.$$

So, all iterative methods are of the type

$$Mx^{(k+1)} = Nx^{(k)} + b, \tag{4.3.3}$$

where $A = M - N$ and $x^{(k)}$, $x^{(k+1)}$ denote the approximation vectors of the solution x at the step k, respectively $k+1$, based on an initial approximation $x^{(0)}$.

Recalling the notion of *spectral radius of a matrix* G as the quantity

$$\rho(G) = \max\{|\lambda| \mid \lambda \text{ eingenvalue of } G\}$$

the following result holds.

Theorem 4.3.1. *Let $A = M - N$ be the decomposition of the regular matrix $A \in \mathcal{M}_{n\times n}(\mathbb{R})$ and $b \in \mathcal{M}_{n\times 1}(\mathbb{R})$. If the matrix M is regular and $\rho(M^{-1}N) < 1$, then the iterative method is convergent.*

Remark 4.3.2. If the iterative method is convergent, then it makes sense to continue with Step 2, in order to get the approximate solution. Otherwise, the method cannot be applied.

Step 2. Starting with an initial value $x^{(0)}$, the sequence of successive approximations

$$x^{(1)}, x^{(2)}, \dots, x^{(k)}, \dots, x^{(n)}, \dots \tag{4.3.4}$$

is generated, according with the iterative method used.

Because the method is convergent, the sequence (4.3.4) will converge to the exact solution x of system (4.3.1).

Step 3. The stop condition is $\|x^{(n)} - x^{(n-1)}\| \leq \varepsilon$, for given ε, where $\|\cdot\|$ is the Euclidean norm. Then x is approximated by $x^{(n)}$ or $x^{(n-1)}$.

Remark 4.3.3. The initial approximation $x^{(0)}$ does not influence the convergence of the method, only its speed-up.

In what follows, we shall present different ways of generating the sequence (4.3.4), according with different iterative methods.

4.3.1 The Jacobi method

Let us write the system (4.3.1) in the following form, having $a_{ii} \neq 0$, $i = \overline{1,n}$:

$$x_i = \left(b_i - \sum_{j=1, j\neq i}^{n} a_{ij} x_j \right) \Big/ a_{ii}, \quad i = \overline{1,n}. \tag{4.3.5}$$

Starting with an initial approximation $x_i^{(0)}$, $i = \overline{1,n}$, Jacobi's iterative process is defined by the algorithm

$$x_i^{(k+1)} = \left(b_i - \sum_{j=1, j\neq i}^{n} a_{ij} x_j^{(k)} \right) \Big/ a_{ii}, \quad i = \overline{1,n}, \tag{4.3.6}$$

or, in matriceal form, based on decomposition (4.3.2) of A,

$$Dx^{(k+1)} = -(L+U)x^{(k)} + b. \tag{4.3.7}$$

So, $M_J = D$ and $N_J = -(L+U)$, and we have to verify for convergence the property

$$\rho(M_J^{-1} N_J) < 1.$$

4.3.2 The Gauss-Seidel method

Considering again the system (4.3.1) in the form (4.3.5), and starting with $x_i^{(0)}$, $i = \overline{1,n}$, the initial approximation, the Gauss-Seidel iterative process is defined by the algorithm:

$$x_i^{(k+1)} = \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right) \Big/ a_{ii}, \quad i = \overline{1,n} \tag{4.3.8}$$

or, in matriceal form, based on (4.3.2),

$$(D+L)x^{(k+1)} = -Ux^{(k)} + b,$$

so $M_{GS} = D + L$ and $N_{GS} = -U$.

To see if the method is convergent, we have to verify

$$\rho(M_{GS}^{-1} N_{GS}) < 1.$$

Remark 4.3.4. If the matrix $A \in \mathcal{M}_{n\times n}(\mathbb{R})$ has a strictly dominant diagonal, it means

$$|a_{ii}| > \sum_{\substack{j=1\\ j\neq i}}^{n} |a_{ij}|, \quad i = \overline{1,n},$$

then the spectral radius $\rho(M_J^{-1}N_J) < 1$.

Also, if $A \in \mathcal{M}_{n\times n}(\mathbb{R})$ is a symmetric positive definite matrix, then the Gauss-Seidel process is convergent.

Example 4.3.5. Solve, with an error of 10^{-1}, the following system:

$$A \cdot x = b,$$

where

$$A = \begin{pmatrix} 1 & 1 & -1 \\ -1 & 3 & 0 \\ 1 & 0 & -2 \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} 0 \\ 2 \\ -3 \end{pmatrix},$$

using Jacobi method.

Let us represent the matrix A in the form

$$\begin{aligned} A &= L + D + U \\ &= \begin{pmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & -2 \end{pmatrix} + \begin{pmatrix} 0 & 1 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \end{aligned}$$

Then, we form the matrices M_J and N_J:

$$M_J = D = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & -2 \end{pmatrix},$$

$$N_J = -(L+U) = \begin{pmatrix} 0 & -1 & 1 \\ 1 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}.$$

In order to determine the convergence of the method, we have to compute

$\rho(M_J^{-1}N_J)$, where $N_J^{-1}N_J$ is:

$$M_J^{-1}N_J = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & -\frac{1}{2} \end{pmatrix} \begin{pmatrix} 0 & -1 & 1 \\ 1 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & -1 & 1 \\ \frac{1}{3} & 0 & 0 \\ \frac{1}{2} & 0 & 0 \end{pmatrix}.$$

The eigenvalues of $M_J^{-1}N_J$ are:

$$\lambda(M_J^{-1}N_J) = \left\{0, -\tfrac{1}{\sqrt{6}}, \tfrac{1}{\sqrt{6}}\right\}$$

and

$$\rho(M_J^{-1}N_J) = \tfrac{1}{\sqrt{6}} < 1,$$

so the Jacobi method converges. It makes sense to continue by generating the successive approximation sequence, with

$$M_J^{-1}b = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & -\frac{1}{2} \end{pmatrix} \begin{pmatrix} 0 \\ 2 \\ -3 \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{2}{3} \\ \frac{3}{2} \end{pmatrix}$$

and $x^0 = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^T$, according with the relation

$$x^{(k+1)} = M_J^{-1}N_J x^{(k)} + M_J^{-1}b.$$

We get

$$x^{(1)} = \begin{pmatrix} 0 & -1 & 1 \\ \frac{1}{3} & 0 & 0 \\ \frac{1}{2} & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{2}{3} \\ \frac{3}{2} \end{pmatrix} = \begin{pmatrix} 0 \\ 0.66667 \\ 1.5 \end{pmatrix}$$

$$x^{(2)} = \begin{pmatrix} 0 & -1 & 1 \\ \frac{1}{3} & 0 & 0 \\ \frac{1}{2} & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0.66667 \\ 1.5 \end{pmatrix} = \begin{pmatrix} 0.83333 \\ 0.66667 \\ 1.5 \end{pmatrix},$$

and so on. After 5 steps, we obtain

$$x^{(5)} = \begin{pmatrix} 0.97226 & 0.99075 & 0.9861 \end{pmatrix}^T,$$

with the required error, 10^{-1}, taking into account that the exact solution of the system is $x = \begin{pmatrix} 1 & 1 & 2 \end{pmatrix}^T$.

Remark 4.3.6. Following the same scheme, we can solve the system with the Gauss-Seidel method.

4.3.3 The method of relaxation

Normally, the Gauss-Seidel method is twice as fast as the Jacobi method but, if the quantity $\rho(M_G^{-1}N_G)$ is smaller than one but close to one, then even the Gauss-Seidel speed of convergence decreases. It arises a problem: how to accelerate the convergence of the sequence of approximations? The method which solve this problem is known as the method of relaxation (or, sometimes, called "successive overrelaxation", or SOR).

It is based on the algorithm:

$$x_i^{(k+1)} = \omega \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right) \Bigg/ a_{ii} + (1-\omega) x_i^{(k)}, \quad i = \overline{1,n},$$

which can be written in matrix form:

$$(D + \omega L) x^{(k+1)} = ((1-\omega)D - \omega U) x^{(k)} + \omega b,$$

where $M_{SOR} = D + \omega L$ and $N_{SOR} = (1-\omega)D - \omega U$.

The problem lies in finding the parameter ω so that $\rho(M_{SOR}^{-1}N_{SOR})$ were the least. According with Kahan's theorem, the method converges for $0 < \omega < 2$.

Remark 4.3.7. For $\omega = 1$, SOR method is the Gauss-Seidel method.

4.3.4 The Chebyshev method

In order to accelerate the convergence of an iterative process, the following Chebyshev's method can be used.

Let us suppose that, using an iterative method, we have found the approximation $x^{(1)}, \ldots, x^{(k)}$ of the solution x of system (4.3.1).

Let be

$$y^{(k)} = \sum_{j=0}^{k} \nu_j(k) x^{(j)}. \tag{4.3.9}$$

The problem lies in finding the coefficients $\nu_j^{(k)}$ in formula (4.3.9) so that the error vector $y^{(k)} - x$ of the approximation $y^{(k)}$ is smaller than the error vector $x^{(k)} - x$. In case of

$$x^{(j)} = x, \quad j = \overline{1,k},$$

it is natural to demand $y^{(k)} = x$. This is just a case when

$$\sum_{j=0}^{k} \nu_j(k) = 1. \tag{4.3.10}$$

In order to determine the coefficients needed, we can write

$$x^{(k)} - x = (M^{-1}N)^k e^{(0)},$$

then

$$\begin{aligned} y^{(k)} - x &= \sum_{j=0}^{k} \nu_j(k)(x^{(j)} - x) = \sum_{j=0}^{k} \nu_j(k)(M^{-1}N)^j e^{(0)} \\ &= \sum_{j=0}^{k} \nu_j(k) G^{(j)} e^{(0)} = p_k(G) e^{(0)}, \end{aligned}$$

where $G = M^{-1}N$ and

$$p_k(z) = \sum_{j=0}^{k} \nu_j(k) z^j. \tag{4.3.11}$$

From (4.3.10) it follows that

$$p_k(1) = 1. \tag{4.3.12}$$

In addition,

$$\|y^{(k)} - x\|_2 \le \|p_k(G)\|_2 \|e^{(0)}\|_2. \tag{4.3.13}$$

In the case of a symmetric matrix G, it eigenvalues satisfy the following inequalities:

$$-1 < \alpha \le \lambda_n \le \cdots \le \lambda_1 \le \beta < 1.$$

If λ is the eigenvalue corresponding to the eigenvector x, then

$$p_k(G)x = \sum_{j=0}^{k} \nu_j(k) G^j x = \sum_{j=0}^{k} \nu_j(k) \lambda^j x,$$

i.e., the vector x is also the eigenvector of the matrix $p_k(G)$ corresponding to the eigenvalue

$$\sum_{j=0}^{k} \nu_j(k) \lambda^j = p_k(\lambda).$$

In case of G symmetric, matrix $p_k(G)$ is also symmetric. Hence,

$$\|p_k(G)\|_2 = \max_{\lambda_i \in \lambda(G)} \left| \sum_{j=0}^{k} \nu_j(k) \lambda_i^j \right| \leq \max_{\lambda \in [\alpha,\beta]} |p_k(\lambda)|. \tag{4.3.14}$$

To decrease the quantity $\|p_k(G)\|_2$ one must find such a polynomial $p_k(z)$ that has small values on the segment $[\alpha, \beta]$ and satisfies the condition (4.3.12). The polynomials which have such properties are the Chebyshev's. They are defined on the interval $[-1, 1]$ by the recurrence relation

$$c_j(z) = 2zc_{j-1}(z) - c_{j-2}(z), \quad j = 2, 3, \ldots,$$

with $c_0(z) = 1$ and $c_1(z) = z$.

These polynomials satisfy the inequality

$$|c_j(z)| \leq 1, \ z \in [-1, 1]$$

and $c_j(1) = 1$, and the values $|c_j(z)|$ grow quickly outside the segment $[-1, 1]$.

Further, the polynomial

$$p_k(z) = \frac{c_k(-1 + 2(z - \alpha)/(\beta - \alpha))}{c_k(\mu)},$$

where

$$\mu = -1 + 2\frac{1 - \alpha}{\beta - \alpha} = 1 + 2\frac{1 - \beta}{\beta - \alpha} > 1$$

satisfies the condition (4.3.12) and

$$|p_k(z)| \leq 1 \text{ for } z \in [\alpha, \beta].$$

Taking into account relations (4.3.13) and (4.3.14), we find

$$\|y^{(k)} - x\|_2 \leq \frac{\|x^{(k)} - x\|_2}{|c_k(\mu)|}.$$

Therefore, the greater μ is, the greater $|c_k(\mu)|$ is and, consequently, the greater will be the convergence of the method.

4.3.5 The method of Steepest Descendent

This method is a very important one, and also contains the main ideas of another well known iterative method: the Conjugate Gradient (CG) method.

For didactical reason, we choose to present in this section only the Steepest Descendent method, because it is easy understandable, and maybe not so well known as CG, in spite of the fact that it has comparable performances.

Let us consider the linear system of equations

$$A \cdot x = b, \tag{4.3.15}$$

where $A \in \mathcal{M}_{n\times n}(\mathbb{R})$ is a symmetric, positive-definite matrix, $x, b \in \mathcal{M}_{n\times 1}(\mathbb{R})$.

With the information given by (4.3.15) we generate the following quadratic form:

$$f(x) = \frac{1}{2}x^T \cdot A \cdot x - b^T \cdot x + c, \tag{4.3.16}$$

where c is a scalar constant, which will have the gradient

$$f'(x) = \frac{1}{2}A^T \cdot x + \frac{1}{2}Ax - b. \tag{4.3.17}$$

If A is symmetric, (4.3.17) reduces to

$$f'(x) = Ax - b. \tag{4.3.18}$$

Remark 4.3.8. We recall that the gradient of a quadratic form is defined to be

$$f'(x) = \begin{pmatrix} \frac{\partial}{\partial x_1}f(x) \\ \frac{\partial}{\partial x_2}f(x) \\ \vdots \\ \frac{\partial}{\partial x_n}f(x) \end{pmatrix}.$$

Setting the gradient $f'(x)$ to zero, in (4.3.18), we get exactly the system (4.3.15) we have to solve. Therefore, the solution to $Ax = b$ is a critical point of $f(x)$. Under the assumption we made on A, then this solution is a minimum of $f(x)$, so $Ax = b$ can be solved by finding an x that minimizes $f(x)$.

Computing the Steepest Descendent method. We start with an initial approximation $x^{(0)}$ and we shall generate the sequence of successive approximation $x^{(1)}, x^{(2)}, \ldots$ choosing the direction in which f decreases most

quickly, which is the direction opposite $f'(x^{(i)})$. According with equation (4.3.18) this direction is

$$-f'(x^{(i)}) = b - Ax^{(i)}. \tag{4.3.19}$$

Recalling the notion of *the residual,* denoted by

$$r_i = b - Ax^{(i)}, \tag{4.3.20}$$

which means how far we are from the correct value of b, and recalling the notion of *the error*,

$$e_i = x^{(i)} - x, \tag{4.3.21}$$

it is easy to see that

$$r_i = -Ae_i \tag{4.3.22}$$

or, more important,

$$r_i = -f'(x^{(i)}). \tag{4.3.23}$$

So, the residual can be considered the direction of the steepest descendent.

Under this considerations, with the initial approximation $x^{(0)}$ given, we shall determine

$$x^{(1)} = x^{(0)} + \alpha r_0, \tag{4.3.24}$$

with α a real value chosen to minimize f along a line.

From a basic calculus, α minimizes f when the directional derivative $\dfrac{d}{d\alpha} f(x^{(1)})$ is equal to zero

$$\frac{d}{d\alpha} f(x^{(1)}) = f'(x^{(1)})^T \frac{d}{d\alpha} x^{(1)} = f'(x^{(1)})^T r_0 = 0. \tag{4.3.25}$$

Relation (4.3.25) indicates that α minimizes f if $f'(x^{(1)})$ and r_0 are orthogonal.

To determine explicitly α, note that $f'(x^{(1)}) = -r_1$ and then we have:

$$\begin{aligned}
&r_1^T \cdot r_0 = 0\\
&(b - Ax^{(1)})^T r_0 = 0\\
&(b - A(x^{(0)} + \alpha r_0))^T r_0 = 0\\
&(b - Ax^{(0)})^T r_0 - \alpha (Ar_0)^T r_0 = 0\\
&(b - Ax^{(0)})^T r_0 = \alpha (Ar_0)^T r_0\\
&r_0^T r_0 = \alpha r_0^T (Ar_0)\\
&\alpha = \frac{r_0^T \cdot r_0}{r_0^T \cdot A \cdot r_0}.
\end{aligned}$$

Putting it all together, the method of Steepest Descent is:

$$\begin{aligned}
r_i &= b - A \cdot x^{(i)}\\
\alpha_i &= \frac{r_i^T \cdot r_i}{r_i^T \cdot A \cdot r_i}\\
x^{(i+1)} &= x^{(i)} + \alpha_i \cdot r_i.
\end{aligned}$$

Remark 4.3.9. It can be proved that the method of Steepest Descent is convergent.

4.3.6 The multigrid method

Originally introduced as a way to numerically solve elliptic boundary-value problems, multigrid methods and their various multiscale descendants have been developed and applied to various problems, in many disciplines. It is among the fastest solution techniques known today.

Let us consider the linear system of equations

$$A \cdot x = b, \tag{4.3.26}$$

with $A \in \mathcal{M}_{n\times n}(\mathbb{R})$, $x, b \in M_{n\times 1}(\mathbb{R})$ which models our problem. So, x will be exact solution. If it is difficult to determine analytically, we try to find an approximation $x_h \in M_{n\times 1}(\mathbb{R})$, which approximates x with the smallest error possible. In order to do this, let's consider the discretized problem attached to (4.3.26), denoted

$$A \cdot x_h = b, \tag{4.3.27}$$

obtained by means of any discretization method (finite difference, finite elements, etc.).

The continuous definition domain will be covered with a mesh of N points (for a 1D problem), or N^2 point (for 2D problem), etc., with the mesh size $h = \frac{1}{N}$, or $h = \frac{1}{N^2}$, etc.

Remark 4.3.10. As in the literature, we use the name "grid" for this mesh.

Computing the Multigrid method. In this case, the algebraic error is

$$e = x - x_h \tag{4.3.28}$$

and the residual is

$$r = b - A \cdot x_h. \tag{4.3.29}$$

Also, the residual equation which may be deduced from (4.3.28) and (4.3.29) is:

$$A \cdot e = r. \tag{4.3.30}$$

which means that the algebraic error satisfies the system (4.3.26), when in r.h.s. we have the residual.

Having all this in mind, we may have a first, very intuitively, image of the multigrid method:

1. Determine an approximation x_h for x using a classic iteration (Jacobi, Gauss-Seidel, SOR, etc.).
2. Compute the residual, according with (4.3.29).
3. Solve the residual equation (4.3.30).
4. Improve the approximation x_h:

$$x = x_h + e.$$

Having these scheme in mind, we need to understand why the method is called multigrid. The answer is find in the way in which the previous steps affect the approximation error.

Using the Fourier analysis, one notices that the error vector is composed by two types of waves: more oscillatory and smoother. In order to reduce the error, we want to smooth as much as possible the components of the error, which means, in fact to reduce it.

So, the multigrid method consists in two parts: applying the smoothing property and the coarse grid correction.

The smoothing properties. It has been proved that the classical iterations, as Jacobi, Gauss-Seidel, etc. has this property, it means that they take off the oscilatory components of the error, but leave the smoother components unchanged. That's why we start the solving of the system (4.3.26) with a number of steps of some classical iterative method, in order to obtain a first approximation, x_h, of the solution.

The Coarse-grid correction. In order to take off the smoother components, too, the information is transferred from the fine grid to a coarser one, with half number of points, because there the components of the error look less smooth.

Remark 4.3.11. The transfer of information is made by means of two operators: one for **prolongation** (I_{2h}^h) and one for **reduction** (I_h^{2h}).

The prolongation operator makes the transfer between the coarse grid to the fine grid; according to the formulas:

$$I_{2h}^h x^{2h} = x^h,$$

where

$$\begin{aligned} &x_{2j}^h = \tfrac{1}{2} x_j^{2h}, \quad 0 \le j \le \tfrac{N}{2} - 1 \\ &x_{2j+1}^h = \tfrac{1}{2}(x_j^{2h} + x_{j+1}^{2h}). \end{aligned} \tag{4.3.31}$$

The restriction operator makes the transfer between the fine grid to the coarse grid, according with the formulas:

$$I_h^{2h} \cdot x^h = x^{2h},$$

where

$$x_j^{2h} = \frac{1}{4}(x_{2j-1}^h + x_{2j}^h + x_{2j+1}^h), \quad 1 \le j \le \frac{N}{2} - 1. \tag{4.3.32}$$

Remark 4.3.12. Formulas (4.3.31) and (4.3.32) shows that

$$I_h^{2h} = c(I_{2h}^h)^T, \text{ with } c \in \mathbb{R}.$$

This fact is very important from computational point of view.
We may write, by short, as follows.
The Two-Grid-Algorithm:
Relax $A^h \cdot x = b^h$ on fine grid Ω^h and get x^h an initial approximation.
Compute $r^{2h} = I_h^{2h}(b^h - A^h x^h)$.

Solve $A^{2h} \cdot e^{2h} = r^{2h}$ on coarse grid Ω^{2h}.
Correct $x^h \leftarrow x^h + I_{2h}^h \cdot e^{2h}$.

The Multigrid Algorithm is a recursive one, and implies this change of information among several grids, the coarsest one being formed only with one point, where the solution of the system can be determined exactly.

So, **the Multigrid Algorithm (MV)** is the following:

2. Relax $A^h x = b^h$ and get x^h.
2. If Ω^h is the coarsest grid, then go to 3.
   Else
   $b^{2h} \leftarrow I_h^{2h}(b^h - A^h x^h)$
   $x^{2h} \leftarrow 0$
   $x^{2h} \leftarrow MV(x^{2h}, b^{2h})$
3. Correct $x^h \leftarrow x^h + I_{2h}^h x^{2h}$.

Remark 4.3.13. The relaxation procedure may appear also at the end of step 3. The previous Multigrid Algorithm is called V-cycle algorithm. There exists, also, W-cycle or combined multigrid algorithms.

Chapter 5

Non-linear equations on $\mathbb{R}$

Let $\Omega \subset \mathbb{R}$ and $f : \Omega \to \mathbb{R}$. Consider the equation

$$f(x) = 0, \quad x \in \Omega, \tag{5.0.1}$$

and attach a mapping

$$F : D \to D, \quad D \subset \Omega^n.$$

Let $(x_0, ..., x_n) \in D$. Using the mapping F and the numbers $x_0, ..., x_{n-1}$ we construct iteratively the sequence

$$x_0, x_1, ..., x_{n-1}, x_n, ..., \tag{5.0.2}$$

where

$$x_i = F(x_{i-n}, ...x_{i-1}), \quad i = n, n+1, ... \tag{5.0.3}$$

The problem is to choose F and the numbers $x_0, ..., x_{n-1} \in D$ such that the sequence (5.0.2) converges to a solution of equation (5.0.1).

Definition 5.0.14. *The method of approximating a solution of equation (5.0.1) by the elements of the sequence (5.0.2), computed as in (5.0.3), is called* F-method *attached to the equation (5.0.1) and to the values* $x_0, ..., x_{n-1}$. *Numbers* $x_0, ..., x_{n-1}$ *are called* starting values*, and the* p*th element of the sequence (5.0.2) is called* p*th approximation order of the solution.*

If the set of the starting values consists in a single element, then the corresponding F-method is called *one-step method*, otherwise it is called *multistep method*.

Definition 5.0.15. *If the sequence (5.0.2) converges to a solution of the equation (5.0.1), then the F-method is said to be* convergent*, otherwise it is* divergent*.*

Definition 5.0.16. *Let $x^* \in \Omega$ be a solution of the equation (5.0.1) and let $x_0, ..., x_n, ...$ be a sequence generated by a given F-method. The number p having the property that*

$$\lim_{x_i \to x^*} \frac{x^* - F(x_{i-n+1}, ..., x_i)}{(x^* - x_i)^p} = C \neq 0,$$

is called the order *of the F-method, i.e.,* $\text{ord}(F) = p$ *and the constant C is* the asymptotical error*.*

In the sequel we describe some general procedures of constructing classes of one-step and multistep F-methods, and we discuss also some simple particular cases.

5.1 One-step methods

Let F be an one-step method, i.e., for a given x_i we have

$$x_{i+1} = F(x_i). \tag{5.1.1}$$

Theorem 5.1.1. *If x^* is a solution of the equation (5.0.1), $V(x^*)$ is a neighborhood of x^* and $F \in C^p[V(x^*)]$, $p \in \mathbb{N}^*$, then*

$$\text{ord}(F) = p,$$

if and only if

$$F(x^*) = x^*, \ F'(x^*) = \ldots = F^{(p-1)}(x^*) = 0, \ F^{(p)}(x^*) \neq 0. \tag{5.1.2}$$

Proof. Since $F \in C^p[V(x^*)]$, Taylor's formula gives rise to

$$F(x_i) = F(x^*) + \sum_{k=1}^{p-1} \tfrac{1}{k!}(x_i - x^*)^k F^{(k)}(x^*) + \tfrac{1}{p!}(x_i - x^*)^p F^{(p)}(\xi_i), \tag{5.1.3}$$

where $x_i \in V(x^*)$ and ξ_i belongs to the interval determined by x_i and x^*.

Assume that conditions (5.1.2) are verified. Then (5.1.3) implies

$$F(x_i) = x^* + \tfrac{1}{p!}(x_i - x^*)^p F^{(p)}(\xi_i), \tag{5.1.4}$$

and it follows

$$\lim_{x_i - x^*} \frac{F(x_i) - x^*}{(x_i - x^*)^p} = \tfrac{1}{p!} F^{(p)}(x^*) \neq 0.$$

Thus,

$$\mathrm{ord}(F) = p.$$

Let now consider $p = \mathrm{ord}(F)$. We have to prove that conditions (5.1.2) are satisfied. Assume either that there exists a number $r < p - 1$ such that $F^{(r)}(x^*) \neq 0$, or that $F^{(p)}(x^*) = 0$. In both cases, from (5.1.3) and (5.1.4), respectively, it follows that

$$\mathrm{ord}(F) \neq p.$$

■

Theorem 5.1.2. *Let x^* be a solution of the equation (5.0.1), $x_0 \in \Omega$,*

$$I = \{x \mid |x - x^*| \leq |x_0 - x^*|, \qquad x \in \Omega\},$$

$F \in C^p(I)$, p is the order of F and

$$\tfrac{1}{p!} \left|F^{(p)}(x)\right| \leq M, \quad x \in I.$$

If

$$M |x_0 - x^*|^{p-1} < 1 \tag{5.1.5}$$

then

1) $x_i \in I, \quad i = 0, 1, ...,$
2) $\lim\limits_{i \to \infty} x_i = x^*,$

where

$$x_{i+1} = F(x_i), \quad i = 0, 1, ...$$

Proof. 1) Obviously, $x_0 \in I$. Assume that $x_i \in I$. Since,

$$\mathrm{ord}(F) = p$$

it follows that (5.1.4) holds, and we can write it as

$$x_{i+1} - x^* = M_i(x_i - x^*)^p, \quad M_i = \frac{1}{p!}F^{(p)}(\xi). \tag{5.1.6}$$

Using (5.1.6), the definition of I and inequality $M_i \leq M$, on interval I, it yields

$$\begin{aligned} |x_{i+1} - x^*| &\leq M\,|x_i - x^*|^p \\ &\leq M\,|x_0 - x^*|^p \\ &= M\,|x_0 - x^*|^{p-1}\,|x_0 - x^*| \\ &< |x_0 - x^*|\,, \end{aligned}$$

which implies $x_{i+1} \in I$ and, according to complete induction, it follows

$$x_i \in I, \quad i = 0, 1, \ldots.$$

2) One considers again the inequality:

$$|x_{i+1} - x^*| \leq M\,|x_i - x^*|^p\,, \quad i = 0, 1, ...,$$

which implies successively

$$\begin{aligned} |x_i - x^*| &\leq M\,|x_{i-1} - x^*|^p \\ &\leq MM^p\,|x_{i-2} - x^*|^{p^2} \\ &\leq ... \leq M^{1+p+...+p^{i-1}}\,|x_0 - x^*|^{p^i}\,. \end{aligned}$$

Hence,

$$\begin{cases} |x_i - x^*| \leq M^{-\frac{1}{p-1}}(M\,|x_0 - x^*|^{p-1})^{p^i/(p-1)}, & \text{for } p > 1, \\ |x_i - x^*| \leq M^i\,|x_0 - x^*|\,, & \text{for } p = 1, \end{cases} \tag{5.1.7}$$

and, taking into account (5.1.5), it follows that the sequence $(x_i)_{i\in N}$ converges to x^*. ■

Remark 5.1.3. Inequalities (5.1.7) also provide upper bounds for the absolute error in approximating x^* by x_i.

Remark 5.1.4. If $p = 1$, the convergence condition is $M < 1$, i.e.,

$$|F'(x)| < 1, \qquad x \in I.$$

Remark 5.1.5. If $p > 1$ there exists always a neighborhood of x^* where the F-method converges.

Next we present some construction procedures for F-methods.

5.1.1 Successive approximations method

One of the easiest way to construct F-method consists in writing the equation (5.0.1) in the equivalent form:

$$x = F(x), \quad \text{with} \quad F(x) = x + f(x),$$

and applying successive approximations method for approximating the fixed points of F, namely,

$$x_{n+1} = F(x_n), \quad n = 0, 1, ...,$$

where x_0 is given. From (5.0.1) it follows that $f(x^*) = 0$ if and only $F(x^*) = x^*$, whence, any root of f is a fixed point of F, and viceversa. This method is based upon the following result.

Theorem 5.1.6. *(Picard-Banach). Let (X, d) be a complete metric space and let $F : X \to X$ be a contraction (i.e., it satisfies Lipschitz condition with a constant $\alpha < 1$). Then, the following statements hold:*

i) The mapping F has a unique fixed point x^*.

ii) If $x_0 \in X$, then the sequence $x_{n+1} = F(x_n)$, $n = 0, 1, ...$ converges to x^*.

iii) One has

$$d(x_n, x^*) \leq \frac{\alpha^n}{1-\alpha} d(x_0, x_1).$$

Proof. From

$$\begin{aligned} d(x_n, x_{n+1}) &= d(F(x_{n-1}), F(x_n)) \\ &\leq \alpha d(x_{n-1}, x_n) \leq ... \leq \alpha^n d(x_0, x_1). \end{aligned}$$

it follows that

$$\begin{aligned} d(x_n, x_{n+p}) &\leq d(x_n, x_{n+1}) + ... + d(x_{n+p-1}, x_{n+p}) \\ &\leq (\alpha^n + ... + \alpha^{n+p-1}) d(x_0, x_1) \\ &\leq (\alpha^n + ...\alpha^{n+p-1} + ...) d(x_0, x_1) \\ &= \frac{\alpha^n}{1-\alpha} d(x_0, x_1). \end{aligned} \tag{5.1.8}$$

Hence, the successive approximations sequence is a Cauchy sequence. Since X is a complete metric space, it follows that the latter sequence is also convergent. Denote its limit by x^*. The mapping F satisfies Lipschitz condition, so it is continuous. Passing to the limit in equality

$$x_{n+1} = F(x_n),$$

it yields

$$x^* = F(x^*),$$

thus x^* is a fixed point of F.

We prove now the uniqueness of the fixed point. Assume the contrary, namely that x_1^* and x_2^* are two distinct fixed points of F. One has

$$0 < d(x_1^*, x_2^*) = d(F(x_1^*), F(x_2^*)) \leq \alpha d(x_1^*, x_2^*),$$

which yields the contradiction $\alpha \geq 1$.

Finally, passing to the limit in (5.1.8) with respect to p, we obtain

$$d(x_n, x^*) \leq \tfrac{\alpha^n}{1-\alpha} d(x_0, x_1).$$

■

Remark 5.1.7. The Picard-Banach theorem provides, beside the existence and uniqueness conditions, an effective method for constructing the successive approximations sequence, as well as an upper bound for the error in approximating the fixed point by the approximation of order n. Since condition

$$M := \sup_{x \in X} |F'(x)| < 1$$

implies $\alpha < 1$, one can use in practical applications the inequality $M < 1$ as convergence test.

A more practical upper bound for the approximation error is given in the next result.

Lemma 5.1.8. *Let x^* be a fixed point of F and let $(x_n)_{n \in N}$ be the successive approximations sequence generated by F, for a given x_0, which converges to x^*. If*

$$|F'(x)| < \alpha, \qquad x \in V(x^*),$$

then the following inequality holds:

$$|x_n - x^*| \leq \tfrac{\alpha}{1-\alpha} |x_n - x_{n-1}|. \tag{5.1.9}$$

Proof. Denoting $f(x) = x - F(x)$, one has

$$f'(x) = 1 - F'(x).$$

Conditions in the hypothesis imply

$$|f'(x)| = |1 - F'(x)| \geq 1 - \alpha, \quad x \in V(x^*).$$

Taking into account that $f(x^*) = 0$, we obtain

$$\begin{aligned} |x_n - F(x_n)| &= |f(x_n) - f(x^*)| \\ &= |f'(\xi)|\,|x_n - x^*| \\ &\geq (1-\alpha)\,|x_n - x^*|\,, \end{aligned}$$

where ξ belongs to the interval determined by x and x^*. We have

$$|x_n - x^*| \leq \tfrac{1}{1-\alpha}\,|x_n - F(x_n)| = \tfrac{1}{1-\alpha}\,|x_{n+1} - x_n|\,.$$

Using again Lagrange formula, this yields

$$|x_{n+1} - x_n| = |F(x_n) - F(x_{n-1})| \leq \alpha\,|x_n - x_{n-1}|\,,$$

whence

$$|x_n - x^*| \leq \tfrac{\alpha}{1-\alpha}\,|x_n - x_{n-1}|\,.$$

■

Remark 5.1.9. Denoting by ε the maximal admissible absolute error, i.e, $|x - x^*| \leq \varepsilon$, from (5.1.9) we obtain

$$|x_n - x_{n-1}| \leq \tfrac{1-\alpha}{\alpha}\varepsilon,$$

which can be used as a stop criterium for the computation.

5.1.2 Inverse interpolation method

Let $x^* \in \Omega$ be a solution of equation (5.0.1) and $V(x^*)$ a neighborhood of x^*. Assume that f has the inverse on $V(x^*)$ and denote $g = f^{-1}$. Since $f(x^*) = 0$ it follows that $x^* = g(0)$.

The inverse interpolation method for approximating a solution x^* of equation (5.0.1) consists in approximating the inverse g by means of a certain interpolation method, and x^* by the value of the interpolating element at point

zero. This approach generates a large number of approximation methods for the solution of an equation (thus, for the zeros of a function), according to the employed interpolation method.

In order to obtain a one-step F-method, all information on f must be given at a single point, namely the starting value. Hence, we are lead to Taylor interpolation.

Theorem 5.1.10. *Let x^* be a solution of equation (5.0.1), $V(x^*)$ a neighborhood of x^*, $f \in C^m[V(x^*)]$, such that*

$$f'(x) \neq 0, \qquad \text{for } x \in V(x^*),$$

and consider $x_i \in V(x^*)$. Then we have the following method for approximating x^*, denoted by F_m^T :

$$F_m^T(x_i) = x_i + \sum_{k=1}^{m-1} \frac{(-1)^k}{k!} f_i^k g^{(k)}(f_i), \tag{5.1.10}$$

where g is the inverse of f and $f_i = f(x_i)$.

Proof. From the hypothesis of this theorem it follows that there exists $g = f^{-1}$ and that $g \in C^m[V(0)]$,where $V(0) = f(V(x^*))$.

Let $y_i = f(x_i)$ and consider Taylor interpolation formula:

$$g(y) = (T_{m-1}g)(y) + (R_{m-1}g)(y), \tag{5.1.11}$$

relative to the function g and the node y_i, where $T_{m-1}g$ is $(m-1)$th degree Taylor polynomial, namely,

$$(T_{m-1}g)(y) = \sum_{k=0}^{m-1} \tfrac{1}{k!}(y - y_i)^k g^{(k)}(y_i),$$

and $R_{m-1}g$ is the corresponding remainder. Since $x^* = g(0)$ and $g \approx T_{m-1}g$, it follows

$$x^* \approx (T_{m-1}g)(0) = x_i + \sum_{k=1}^{m-1} \frac{(-1)^k}{k!} y_i^k g^{(k)}(y_i),$$

whence,

$$x_{i+1} := F_m^T(x_i) = x_i + \sum_{k=1}^{m-1} \frac{(-1)^k}{k!} f_i^k g^{(k)}(f_i)$$

is an approximation of x^*, and F_m^T is an approximation method for the solution x^*. ■

Concerning the order of the method F_m^T we state the following result.

Theorem 5.1.11. *Let x^* be a solution of equation (5.0.1), $f \in C^m[V(x^*)]$ and $f'(x^*) \neq 0$. If $g = f^{-1}$ satisfies condition*

$$g^{(m)}(0) \neq 0,$$

then the method F_m^T is of order m.

Proof. Consider the remainder of the Taylor interpolation formula written in Lagrange form, namely,

$$(R_{m-1}g)(y) = \tfrac{1}{m!}(y - y_i)^m g^{(m)}(y_i + \theta(y - y_i)), \quad 0 < \theta < 1.$$

Since $(R_{m-1}g)(0)$ is the error of approximating x^* by $(T_{m-1}g)(0)$, one has

$$x^* - F_m^T(x_i) = \tfrac{(-1)^m}{m!} f_i{}^m g^{(m)}(y_i(1-\theta)). \tag{5.1.12}$$

Taking into account that $f(x^*) = 0$, it follows

$$f(x_i) = f(x_i) - f(x^*) = (x_i - x^*)f'(\xi_i), \tag{5.1.13}$$

where ξ_i belongs to the interval determined by x_i and x^*. Then (5.1.13) implies

$$x^* - F_m^T(x_i) = \tfrac{(-1)^m}{m!}(x_i - x^*)^m [f'(\xi_i)]^m g^{(m)}(y_i(1-\theta)),$$

and thus,

$$\frac{x^* - F_m^T(x_i)}{(x^* - x_i)^m} = \tfrac{1}{m!}[f'(\xi_i)]^m g^{(m)}(y_i(1-\theta)).$$

Since $g^{(m)}(0) \neq 0$ and $f'(x^*) \neq 0$, we obtain

$$\lim_{x_i \to x^*} \frac{x^* - F_m^T(x_i)}{(x^* - x_i)^m} = \tfrac{1}{m!}[f'(x^*)]^m g^{(m)}(0) \neq 0,$$

whence,

$$\operatorname{ord}(F_m^T) = m.$$

■

Remark 5.1.12. From (5.1.13) we get

$$\left|x^* - F_m^T(x_i)\right| \leq \tfrac{1}{m!}|f_i|^m M_m g, \quad \text{with} \quad M_m = \sup_{y \in V(0)} \left|g^{(m)}(y)\right|,$$

which provides an upper bound for the absolute error in approximating x^* by x_{i+1}.

Next we present two particular cases.

1) Newton-Raphson method

This method is obtained for $m = 2$, namely

$$F_2^T(x_i) = x_i - \frac{f(x_i)}{f'(x_i)}.$$

Thus, the Newton-Raphson method (also called *the Newton's method* or *the tangent method*) is of order $p = 2$. According to Remark 5.1.5, there always exists a neighborhood of x^* where F_2^T is convergent. Choosing the starting value x_0 in such a neighborhood, allows approximating the solution x^* by terms of the sequence generated by F_2^T, namely

$$x_{i+1} = F_2^T(x_i), \quad i = 0, 1, ..., \tag{5.1.14}$$

with a prescribed error. As far as the approximation error is concerned, Remark 5.1.12 gives

$$\left|x^* - F_2^T(x_n)\right| \leq \tfrac{1}{2}[f(x_n)]^2 M_2 g.$$

Lemma 5.1.13. *Let $x^* \in (a, b)$ be a solution of equation (5.0.1) and let*

$$x_n = F_2^T(x_{n-1}).$$

Then

$$|x^* - x_n| \leq \tfrac{1}{m_1} \left|f(x_n)\right|, \qquad m_1 \leq m_1 f = \inf_{a \leq x \leq b} \left|f'(x)\right|.$$

Proof. We use the mean's formula

$$f(x^*) - f(x_n) = f'(\xi)(x^* - x_n),$$

with ξ belonging to the interval determined by x^* and x_n. From

$$f(x^*) = 0$$

and

$$|f'(x)| \geq m_1,$$

for $x \in (a, b)$, it follows

$$|f(x_n)| \geq m_1 |x^* - x_n|,$$

i.e,

$$|x^* - x_n| \leq \tfrac{1}{m_1} |f(x_n)|.$$

■

In practical applications the next evaluation is more useful.

Lemma 5.1.14. *Let $x^* \in (a, b)$ be a solution of equation (5.0.1) and let $x_n = F_2^T(x_{n-1})$ be the nth order approximation generated by F_2. If $f \in C^2[a, b]$ and F_2^T is convergent, then there exists $n_0 \in \mathbb{N}$ such that*

$$|x_n - x^*| \leq |x_n - x_{n-1}|, \qquad n > n_0.$$

Proof. We start with Taylor's 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

$$f(x_{n-1}) + (x_n - x_{n-1})f'(x_{n-1}) = 0.$$

Thus, we obtain

$$f(x_n) = \tfrac{1}{2}(x_n - x_{n-1})^2 f''(\xi).$$

Consequently,

$$|f(x_n)| \leq \tfrac{1}{2} M_2 f (x_n - x_{n-1})^2,$$

and using (5.1.13) yields

$$|x^* - x_n| \leq \tfrac{1}{2m_1}(x_n - x_{n-1})^2 M_2 f.$$

Since F_2^T is convergent, there exists $n_0 \in \mathbb{N}$ such that

$$\tfrac{1}{2m_1} |x_n - x_{n-1}| M_2 f < 1, \quad n > n_0$$

whence,

$$|x^* - x_n| \leq |x_n - x_{n-1}|, \quad n > n_0.$$

■

In general, choosing the starting value is a non-trivial problem. Often this value is chosen randomly, though taking into account the available information about the function f. If after a fixed number of iterations the required precision was 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.

For certain classes of functions the problem of choosing the starting value can be significantly simplified. For example, if the solution x^* is isolated in the interval (a, b) and $f''(x) \neq 0$, $x \in (a, b)$, then one can choose as a starting value in F_2^T-method one of the endpoints a or b. More precisely,

$$x_0 = \begin{cases} a, & f(a)f''(c) > 0 \quad \text{or} \quad f(b)f''(c) < 0, \\ b, & f(a)f''(c) < 0 \quad \text{or} \quad f(b)f''(c) > 0, \end{cases}$$

where $c \in (a, b)$, (for example, $c = (a + b)/2$).

This statement can be intuitively justified using the geometric interpretation of the method. Consider, for example, the case $f(a) < 0$, $f(b) > 0$ and $f''(x) < 0$, for $x \in (a, b)$. It follows immediately $x_0 = a$. Indeed,

$$f(a)f''(\tfrac{a+b}{2}) > 0.$$

Next we consider the second particular case.

2) Case $m = 3$.

Using (5.1.10), we obtain

$$F_3^T(x_i) = x_i - \frac{f(x_i)}{f'(x_i)} - \frac{1}{2}\left[\frac{f(x_i)}{f'(x_i)}\right]^2 \frac{f''(x_i)}{f'(x_i)}, \tag{5.1.15}$$

which is a method of order $p = 3$ and thus converges faster than F_2^T.

Remark 5.1.15. The higher the order of a method is, the faster the method converges. Still this doesn't mean that a higher order method of approximation is more efficient (taking into account the computation requirements) than a lower order approximation method. By the contrary, the results in the literature show that the methods of relatively low order are the most efficient ones, due to their simplicity. Within the one-step methods class, the most efficient methods (in the above sense) are considered F_2^T or F_3^T.

5.2 Multistep methods

Recall that if the set of the starting values consists of more than one element, the corresponding F-method is called *multistep method*.

In the sequel we will also use the inverse interpolation method to generate multistep methods. Obviously, in this case there has to be used an interpolation method with more that one interpolation node.

Let $x^* \in \Omega$ be a solution of equation (5.0.1), let $(a, b) \subset \Omega$ be a neighborhood of x^* that isolates this solution and $x_0, ..., x_n \in (a, b)$ some given values. Denote by g the inverse function of f, supposing it exists. Because $x^* = g(0)$, the problem reduces to approximating the inverse function g by means of an interpolation method with $n > 1$ nodes, for example Lagrange, Hermite, Birkhoff, spline, etc. We discuss first the use of Lagrange interpolation. Let

$$y_k = f(x_k), \qquad k = 0, ..., n,$$

whence,

$$x_k = g(y_k).$$

To the data y_k and $g(y_k)$, $k = 0, ..., n$, we attach the Lagrange interpolation formula:

$$g = L_n g + R_n g, \tag{5.2.1}$$

where

$$(L_n g)(y) = \sum_{k=0}^{n} \frac{(y-y_0)...(y-y_{k-1})(y-y_{k+1})...(y-y_n)}{(y_k-y_0)...(y_k-y_{k-1})(y_k-y_{k+1})...(y_k-y_n)} g(y_k).$$

If $g \in C^{n+1}[c, d]$, where $[c, d]$ is the image of the interval $[a, b]$ through the function f, then

$$(R_n g)(y) = \frac{u(y)}{(n+1)!} g^{(n+1)}(\eta), \quad c < \eta < d. \tag{5.2.2}$$

Taking

$$F_n^L(x_0, ..., x_n) = (L_n g)(0),$$

it follows that F_n^L is a $(n + 1)$-steps method defined by

$$F_n^L(x_0, ..., x_n) = \sum_{k=0}^{n} \frac{f_0...f_{k-1}f_{k+1}...f_n}{(f_0-f_k).../...(f_n-f_k)} x_k, \tag{5.2.3}$$

with $f_k = f(x_k)$.

Concerning the convergence of this method we state the following result.

Theorem 5.2.1. *If $x^* \in (a,b)$ is a solution of the equation (5.0.1), f' is bounded on (a,b), and the starting values satisfy*

$$|x^* - x_k| < \tfrac{1}{c}, \qquad k = 0, ..., n,$$

with the constant c given in the proof, then the sequence

$$x_{i+1} = F_n^L(x_{n-i}, ..., x_i), \quad i = n,\ n+1, ...$$

converges to x^*.

Proof. From formulas (5.2.1)–(5.2.2) we get

$$x^* - x_{n+1} = (R_n g)(0)$$

and

$$x^* - x_{n+1} = (-1)^{n+1} c_g \prod_{k=0}^{n} f_k,$$

where

$$c_g = \tfrac{1}{(n+1)!} g^{(n+1)}(\eta).$$

Using the mean's formula, we have

$$f(x_k) = f(x_k) - f(x^*) = (x_k - x^*) f'(\xi_k),$$

whence,

$$x^* - x_{n+1} = c_g \prod_{k=0}^{n} (x^* - x_k) f'(\xi_k).$$

Since f' is bounded, it follows that there exists a constant c such that

$$c\,|x^* - x_{n+1}| \le \prod_{k=0}^{n} c\,|x^* - x_k|\,. \tag{5.2.4}$$

Taking into account that

$$|x^* - x_k| < \tfrac{1}{c},$$

we obtain

$$c\,|x^* - x_{n+1}| < 1.$$

Denoting $M = \max\limits_{0\le k\le n}(c\,|x^* - x_k|)$, it follows that there exist the numbers $a_k > 0$ such that

$$c\,|x^* - x_k| = M^{a_k},$$

for $k = 0, ..., n$. Then, from (5.2.4) we get

$$M^{a_{n+1}} \le M^{a_0+...+a_n},$$

thus,

$$a_{n+1} \ge a_0 + ... + a_n.$$

Considering as starting values $x_{n-i}, ..., x_i$ and applying the above arguments for $i = n+1,\ n+2, ...$, we obtain

$$a_{n+i} \ge a_{i-1} + ... + a_{n+i-1}, \quad i = 1, 2, \dots. \tag{5.2.5}$$

Since $a_i > 0$, it follows that the sequence $a_0, ..., a_n, ...$ is monotonically increasing and

$$\lim_{n\to\infty} a_i = \infty.$$

Indeed, if

$$\lim_{n\to\infty} a_n = \alpha < \infty,$$

inequalities (5.2.5) lead to the contradiction $\alpha \ge (n+1)\alpha$, for $n > 1$. Consequently, condition $0 < M < 1$ implies

$$\lim_{i\to\infty} c\,|x^* - x_i| = \lim_{i\to\infty} M^{a_i} = 0,$$

thus,

$$\lim_{i\to\infty} x_i = x^*, \qquad (c \ne 0).$$

■

Remark 5.2.2. For $n = 1$, we get

$$F_1^L(x_0, x_1) = x_1 - \frac{(x_1 - x_0)f(x_1)}{f(x_1) - f(x_0)},$$

which is called *the secant method.* Thus,

$$x_{k+1} := F_1^L(x_{k-1}, x_k) = x_k - \tfrac{(x_k - x_{k-1})f(x_k)}{f(x_k) - f(x_{k-1})}, \quad k = 1, 2, ...$$

is the new approximation obtained using the previous approximations x_{k-1}, x_k.

Practical F-methods for approximating the zeros of a function can be obtained using inverse Hermite or Birkhoff interpolation. Assume x^* is a solution of equation (5.0.1), $V(x^*)$ is a neighborhood of x^* such that $g = f^{-1}$ exists on $V(x^*)$ and $x_0, x_1 \in V(x^*)$.

Consider the Birkhoff-type interpolation formula

$$g(y) = (B_1 g)(y) + (R_1 g)(y),$$

where

$$(B_1 g)(y) = (y - y_1)g'(y_0) + g(y_1),$$

with $y_0 = f(x_0)$, $y_1 = f(x_1)$, and for $y_0 < y_1$

$$(R_1 g)(y) = \tfrac{1}{2}(y - y_1)(y + y_1 - 2y_0)g''(\eta), \tag{5.2.6}$$

with $\eta \in [y_0, y_1]$, for every $y \in [y_0, y_1]$. Taking $F_1^B(x_0, x_1) = (B_1 g)(0)$, we obtain the F_1^B-method, defined by

$$F_1(x_0, x_1) = x_1 - \tfrac{f(x_1)}{f'(x_0)}.$$

Therefore we can state the following result.

Theorem 5.2.3. *If the following conditions hold:*

i) $f(x_0) < 0 < f(x_1)$;

ii) there exists $f'(x)$, *it is finite, and* $f'(x) > 0$, *for* $x \in [x_0, x_1]$;

iii) there exists $f''(x)$, *it is finite, and* $f''(x) \leq 0$, *for* $x \in (x_0, x_1)$;

then
1) equation (5.0.1) has the unique solution x^* *and* $x_0 < x^* < x_1$.
2) the sequence $(x_n)_{n\in\mathbb{N}}$, *defined by the method* F_1^B, *namely,*

$$x_{n+1} = F_1^B(x_0, x_n), \quad n = 1, 2, ... \tag{5.2.7}$$

converges to x^*. *Moreover, we have*

$$x^* \leq x_{n+1} \leq x_n, \quad n = 1, 2, ... \tag{5.2.8}$$

Proof. Existence and uniqueness follow immediately from conditions **i**) and **ii**). Using complete induction we first prove the inequalities (5.2.8). Indeed, we have $x^* < x_1$. Assume $x^* < x_n$. Using the remainder expression (5.2.6), we obtain

$$x^* - x_{n+1} = (R_1 g)(0) = \frac{1}{2}[f(x_n) - 2f(x_0)]\frac{f''(\xi_n)}{f'(\xi_n)]^3}.$$

From the inequalities

$$f(x_n) \geq 0,\ (x_n \geq x^*), \qquad f(x_0) < 0, \qquad f'(x) > 0, \qquad f''(x) \leq 0,$$

for $x \in (x_0, x_1)$, it follows

$$x^* - x_{n+1} \leq 0,$$

so

$$x^* \leq x_{n+1},$$

and thus the first inequality in (5.2.8) is proved. From (5.2.7), we get

$$x_{n+1} - x_n = -\frac{f(x_n)}{f'(x_0)} \leq 0,$$

whence $x_{n+1} \leq x_n$, and (5.2.8) is completely verified. Consequently, $(x_n)_{n\in\mathbb{N}}$ is a decreasing and bounded sequence, thus it is convergent. We denote by $\overline{x}$ its limit. Then, (5.2.7) implies

$$\lim_{n\to\infty} \frac{f(x_n)}{f'(x_0)} = \lim_{n\to\infty} (x_{n+1} - x_n) = 0,$$

so

$$\lim_{n\to\infty} f(x_n) = f(x^*) = 0,$$

and

$$x_0 < x^* < x_1.$$

Since x^* is unique, it follows $\overline{x} = x^*$, thus, the theorem is completely proved. ■

In the sequel we present the F_2^H-method obtained by using the inverse Hermite interpolation relative to the function $g = f^{-1}$, the simple node $y_0 = f(x_0)$ and the double node $y_1 = f(x_1)$, namely,

$$g(y) = (H_2 g)(y) + (R_2 g)(y),$$

where

$$(H_2g)(y) = \frac{(y-y_1)^2}{(y_0-y_1)^2}g(y_0) + \frac{(y-y_0)(2y_1-y_0-y)}{(y_0-y_1)^2}g(y_1) - \frac{(y-y_0)(y-y_1)}{y_0-y_1}g'(y_1)$$

and

$$(R_2g)(y) = \frac{(y-y_0)(y-y_1)^2}{6}g'''(\eta), \quad y_0 \leq \eta \leq y_1.$$

We obtain

$$F_2^H(x_0,x_1) = x_1 - \left[\frac{f(x_1)}{f(x_0)-f(x_1)}\right]^2 (x_1-x_0) - \frac{f(x_1)}{f(x_0)-f(x_1)}\frac{f(x_0)}{f'(x_1)} \tag{5.2.9}$$

and

$$x^* - F_2^H(x_0,x_1) = -\frac{f(x_0)f^2(x_1)}{6}g'''(\eta_0). \tag{5.2.10}$$

Theorem 5.2.4. *If f satisfies the requirements of Theorem 5.2.3, $g'''(y) \geq 0$, for $y \in [y_0,y_1]$, and $(x_n)_{n\in\mathbb{N}}$ is the sequence generated by F_2^H, i.e.,*

$$x_{n+1} = F_2^H(x_{n-1},x_n), \qquad n = 1,2,...$$

then

$$(x^*-x_n)(x^*-x_{n+1}) \leq 0, \qquad n = 1,2,...$$

and

$$\lim_{n\to\infty} x_n = x^*.$$

Proof. From equalities (5.2.9), (5.2.10) and the hypothesis of the theorem it follows $x_2 < x_1$ and $x_2 \leq x^*$. Since $f(x_2) < 0$ ($f(x_2) = 0$ implies $x_2 = x^*$) and $f(x_1) > 0$ we also get $x_3 > x_2$ and $x^* \leq x_3$. Assume now that $x_{n-1} \leq x^* < x_n$. Then, $f(x_{n-1}) \leq 0$, $f(x_n) \geq 0$, thus $x_{n+1} \leq x_n$ and $x_{n+1} \leq x^*$, so the first claim is fulfilled.

In order to prove that $\lim\limits_{n\to\infty} x_n = x^*$, notice first from the construction of F_2^H that $x_n \in (x_0,x_1)$, $n = 2,...$, i.e., $(x_n)_{n\in\mathbb{N}}$ is a bounded sequence, hence it contains a convergent subsequence $(x_{n_k})_{k\in\mathbb{N}}$. Let $\overline{x} = \lim_{k\to\infty} x_{n_k}$. Then

$$\lim_{k\to\infty}(x_{n_{k+1}} - x_{n_k}) = 0,$$

and (5.2.9) implies

$$\lim_{k\to\infty} f(x_{n_k}) = 0 = f(\overline{x}),$$

where $\overline{x} \in (x_0,x_1)$. Finally, from the uniqueness of the solution x^* we get $\overline{x} = x^*$. ■

5.3 Combined methods

In applications one needs to approximate the solution of an equation with a prescribed error. Usually, this error is evaluated by means of the distance between two consecutive approximations, say $|x_n - x_{n-1}|$. Since inequality $|x_n - x_{n-1}| \leq \varepsilon$ does not always imply $|x_n - x^*| \leq \varepsilon$, where $\varepsilon \in \mathbb{R}_+$ is given, it turned out that very useful are the so-called *combined methods*, where the solution x^* is approximated both from bellow and from above, thus it always lays between two consecutive approximations, namely

$$x_n^s \leq x^* \leq x_n^d, \quad n = 1, 2, ...$$

In such cases, the error in approximating x^* by $(x_n^s + x_n^d)/2$ is not bigger than $(x_n^d - x_n^s)/2$. In other words, the approximation error can be strictly controlled. In the sequel we describe some simple combined methods.

1) The tangent-secant method

If function f is convex or concave on an interval (a, b) that isolates the solution x^* of equation (5.0.1), i.e., $f''(x) \neq 0$, $x \in (a, b)$, then both successive approximation sequences generated by the tangent method and the secant method are monotone. Namely, if the first sequence is increasing, the second one is decreasing, and viceversa. The endpoints of the interval a and b can be respectively chosen as starting values. Thus, in order to approximate $x^* \in (a, b)$ we can apply simultaneously the tangent and the secant methods. Denoting by x_k^t and x_k^s, respectively, the corresponding approximations of order k, we obtain:

$$(x_0^t, x_0^s) \supset (x_1^t, x_1^s) \supset ... \supset (x_n^t, x_n^s) \supset ..., \tag{5.3.1}$$

or

$$(x_0^s, x_0^t) \supset (x_1^s, x_x^t) \supset ... \supset (x_n^s, x_n^t) \supset ..., \tag{5.3.2}$$

where

$$x_{k+1}^t = x_k^t - \frac{f(x_k^t)}{f'(x_k^t)}, \quad k = 0, 1, ...,$$
$$x_{k+1}^s = x_k^s - \frac{(x_k^s - x_{k+1}^t) f(x_k^s)}{f(x_k^s) - f(x_{k+1}^t)}, \quad k = 0, 1, ...,$$

depending whether $f(a)f''\left(\frac{a+b}{2}\right) > 0$ or $f(a)f''\left(\frac{a+b}{2}\right) < 0$. In the first case we have $x_0^t = a$, $x_0^s = b$, and $x^* \in (x_n^t, x_n^s)$, while in the second one $x_0^s = a$,

$x_0^t = b$ and $x^* \in (x_n^s, x_n^t)$. Successive approximations will be computed until there holds

$$\left|x_n^t - x_n^s\right| \leq 2\varepsilon,$$

where ε is a prescribed positive number. When the above inequality is satisfied, we take

$$x^* \approx \overline{x} = \frac{x_n^t + x_n^s}{2},$$

and so,

$$|x^* - \overline{x}| \leq \varepsilon.$$

2) The tangent method $-F_1^B$

This method is based on the following result.

Lemma 5.3.1. *If the requirements of Theorem 5.2.3 are fulfilled, namely, if there take place:*

i) $f(x_0) < 0 < f(x_1)$;

ii) there exists $f'(x)$, *it is finite, and* $f'(x) > 0$, *for* $x \in [x_0, x_1)$;

iii) there exists $f''(x)$, *it is finite, and* $f''(x) \leq 0$, *for* $x \in (x_0, x_1)$;

then the sequence

$$x_0, x_1, ..., x_n, ...$$

generated by the following iterations

$$x_{2n} = x_{2n-2} - \tfrac{f(x_{2n-2})}{f'(x_{2n-2})} \tag{5.3.3}$$

and

$$x_{2n+1} = x_{2n-1} - \tfrac{f(x_{2n-1})}{f'(x_{2n-2})}, \tag{5.3.4}$$

$n = 1, 2, ...,$ *converges to* x^*. *Moreover,* $x^* \in [x_{2n}, x_{2n+1}]$, *for all* $n \in \mathbb{N}$.

Proof. Notice that the iteration in (5.3.3) is the one in Newton's method (i.e., tangent method), and inequalities

$$x_{2n} \leq x_{2n+2} \leq x^*, \qquad n = 0, 1, ...,$$

are satisfied in the given hypothesis.

Furthermore, from Theorem 5.2.3 we get

$$x^* \leq x_{2n+1} \leq x_{2n-1},\ n = 1, 2, \ldots.$$

■

This method has the advantage that each component uses the value of the derivative f' on the same point.

Remark 5.3.2. A much simpler combined method can be obtained using (5.3.3) and (5.3.4), for a fixed x_0, namely

$$\begin{aligned} x_{2n} &= x_{2n-2} - \tfrac{f(x_{2n-2})}{f'(x_0)}, \\ x_{2n+1} &= x_{2n-1} - \tfrac{f(x_{2n-1})}{f'(x_0)}, \qquad n = 1, 2, ..., \end{aligned}$$

where only the value of f' in x_0 is needed. This method is recommended whenever a lot of work is involved in computing a value of the derivative. Other combined methods can be obtained using F_1^B, F_2^H, F_3^T, the tangent and the secant method.

Chapter 6

Non-linear equations on $\mathbb{R}^n$

Let $D \subseteq \mathbb{R}^n$, $f_i : D \to \mathbb{R}$, $i = 1, ..., n$ and consider the system:

$$f_i(x_1, ..., x_n) = 0, \qquad i = 1, ..., n;\ (x_1, ..., x_n) \in D. \tag{6.0.1}$$

Taking the application $f : D \to \mathbb{R}^n$, such that

$$(x_1, ..., x_n) \stackrel{f}{\to} (f_1(x_1, ..., x_n), ..., f_n(x_1, ..., x_n)),$$

system (6.0.1) can be written also as the vectorial equation

$$f(x) = 0, \qquad x \in D, \tag{6.0.2}$$

where 0 is the null element of the space $\mathbb{R}^n$.

In the sequel we describe two methods for approximating the solutions of a non-linear system on $\mathbb{R}^n$.

6.1 Successive approximation method

Consider first the system (6.0.1) written in the form

$$x_i = \varphi_i(x_1, ..., x_n), \qquad i = 1, ..., n;\ (x_1, ..., x_n) \in D, \tag{6.1.1}$$

where the functions $\varphi_i : D \to \mathbb{R}$ are continuous on D and such that for all $(x_1, ..., x_n) \in D$ one has

$$(\varphi_1(x_1, ..., x_n), ..., \varphi_n(x_1, ..., x_n)) \in D.$$

Let $x = (x_1, ..., x_n)$ and $\varphi = (\varphi_1, ..., \varphi_n)$, then the system can be written as

$$x = \varphi(x), \qquad x \in D. \tag{6.1.2}$$

Like in the real case, for a given starting value $x^{(0)}$, one generates the sequence

$$x^{(0)}, x^{(1)}, ..., x^{(n)}, ..., \tag{6.1.3}$$

defined by

$$x^{(m+1)} = \varphi(x^{(m)}), \quad m = 0, 1, \ldots. \tag{6.1.4}$$

If the sequence (6.1.3) is convergent (the φ method is convergent), and $x^* \in D$ denotes its limit, then x^* is a solution of equation (6.1.2). Indeed, passing to the limit in (6.1.4) and taking into account that φ is a continuous function, we obtain

$$\lim_{m\to\infty} x^{(m+1)} = \varphi \lim_{m\to\infty} x^{(m)},$$

that is

$$x^* = \varphi(x^*).$$

The convergence of the φ method remains to be studied. To that end we can use the Picard-Banach theorem: if $\varphi : \mathbb{R}^n \to \mathbb{R}^n$ satisfies the contraction condition

$$\|\varphi(x) - \varphi(y)\| \leq \alpha \|x - y\|, \quad x, y \in \mathbb{R}^n,$$

with $0 < \alpha < 1$, then there exists a unique element $x^* \in \mathbb{R}^n$ which verifies equation (6.1.2) and it is the limit of the sequence (6.1.3). The error in approximating x^* by $x^{(n)}$ is evaluated by

$$\left\|x^* - x^{(n)}\right\| \leq \tfrac{\alpha^n}{1-\alpha} \left\|x^{(1)} - x^{(0)}\right\|.$$

Remark 6.1.1. As in the case of equations on $\mathbb{R}$, a sufficient condition for a continuous function φ, with first order partial derivatives continuous on a domain $D \in \mathbb{R}^n$, to be a contraction is that

$$\|\varphi'\| \leq q < 1, \quad \text{ on } D,$$

where the norm of φ' is the norm of the Jacobian matrix

$$\varphi' = \left(\tfrac{\partial \varphi_i}{\partial x_j}\right), \qquad i, j = 1, ..., n.$$

For example,

$$\|\varphi'\| = \max_{1\leq i\leq m} \sum_{j=1}^{n} \max_{x\in D} \left|\tfrac{\partial \varphi_i(n)}{\partial x_j}\right|.$$

Remark 6.1.2. If φ is defined on a subdomain of $\mathbb{R}^n$, for example, on the sphere

$$S := S(x^{(0)}, r) = \{x \,|\, \|x - x^{(0)}\| < r, \qquad r \in \mathbb{R},\, x^{(0)} \in \mathbb{R}^n\},$$

then the supplementary condition

$$\|x^{(0)} - \varphi(x^{(0)})\| \leq (1 - \alpha)r$$

has to be fulfilled in order to guarantee the convergence of the corresponding iterative process. This condition makes sure that $\varphi(x) \in S$, provided $x \in S$.

The successive approximations method can also be applied for equations of the form (6.0.2), where f is a function defined and continuous in a neighborhood of the solution vector x^*. We first write the equation (6.0.2) in the form

$$x = x + Mf(x),$$

where M is a non-singular matrix. Denoting

$$x + Mf(x) = \varphi(x), \tag{6.1.5}$$

equation (6.0.2) becomes $x = \varphi(x)$, so the successive approximations method can now be applied. Next, we determine the matrix M such that the method $\varphi = I + Mf$ is convergent. For this, we use condition

$$\|\varphi'\| \leq q < 1$$

and the obvious observation is that the smaller $\|\varphi'\|$ is, the faster the iterative process converges. From (6.1.5) we get

$$\varphi'(x) = I + Mf'(x).$$

We determine the matrix M such that for the starting value $x^{(0)}$ one has

$$\varphi'(x^{(0)}) = 0.$$

It follows

$$I + Mf'(x^0) = 0,$$

thus, if $f'(x_0) \neq 0$ we obtain

$$M = -(f'(x^{(0)}))^{-1}.$$

If $f'(x^{(0)}) = 0$ another starting value $x^{(0)}$ has to be chosen.

6.2 Newton's method

Consider again equation (6.0.2). Let $x^* \in D$ be a solution of this equation and let $x^{(p)}$ be an approximation of x^*. Denoting by $\varepsilon^{(p)}$ the error produced in approximating x^* by $x^{(p)}$, we have

$$x^* = x^{(p)} + \varepsilon^{(p)}.$$

Hence,

$$f(x^{(p)} + \varepsilon^{(p)}) = 0. \tag{6.2.1}$$

Assuming that f is derivable in a convex domain that contains x^* and $x^{(p)}$, the left side term in (6.2.1) can be approximated by the first two terms of the Taylor's expansion of f in the neighborhood of $x^{(p)}$, namely,

$$f(x^{(p)} + \varepsilon^{(p)}) \approx f(x^{(p)}) + f'(x^{(p)})\varepsilon^{(p)}.$$

Thus, (6.2.1) can be approximated by

$$f(x^{(p)}) + f'(x^{(p)})\varepsilon^{(p)} = 0, \tag{6.2.2}$$

which leads to the following linear algebraic systems of equations with the unknowns $\varepsilon_i^{(p)}$, $i = 1, ..., n$:

$$\begin{aligned} \tfrac{\partial}{\partial x_1} f_i(x_1^{(p)}, ..., x_n^{(p)})\varepsilon_1^{(p)} + ... + \tfrac{\partial}{\partial x_n} f_i(x_1^{(p)}, ..., x_n^{(p)})\varepsilon_n^{(p)} \\ = -f_i(x_1^{(p)}, ..., x_n^{(p)}), \qquad i = 1, ..., n \end{aligned} \tag{6.2.3}$$

Denoting

$$w(x^{(p)}) = f'(x^{(p)}) = \left(\tfrac{\partial}{\partial x_j} f_i(x^{(p)})\right)_{i,j=1,...,n}$$

and in the hypothesis that $w(x^{(p)})$ is a non-singular matrix, we obtain

$$\varepsilon^{(p)} = -w^{-1}(x^{(p)})f(x^{(p)}),$$

where $w^{-1}(x^{(p)})$ is the inverse of the matrix $w(x^{(p)})$. Thus, we get a value for the correction vector $\varepsilon^{(p)}$.

If we denote by $x^{(p+1)}$ the approximation obtained by applying to $x^{(p)}$ the correction $\varepsilon^{(p)}$, we reach the iterative method:

$$x^{(p+1)} = x^{(p)} - w^{-1}(x^{(p)})f(x^{(p)}), \qquad p = 0, 1, ..., \tag{6.2.4}$$

which is called *Newton's method*.

Passing to the limit in (6.2.4), it immediately follows that if $(x^{(p)})_{p\in\mathbb{N}}$ is a convergent sequence with limit denoted by x^*, then x^* is the solution of the discussed equation.

Concerning the convergence of the sequence we can state the following result.

Theorem 6.2.1. *Let $f \in C^2(D)$, $x^{(0)} \in D$ and $r \in \mathbb{R}$ such that*

$$S_0(x^{(0)}, r) = \{x \,|\, \|x - x^{(0)}\| \leq r\} \subset D.$$

If the following conditions are verified:

i) there exists the matrix $M_0 = w^{-1}(x^{(0)})$ and $\|M_0\| \leq A_0$,

ii) $\|M_0 f(x^{(0)})\| \leq B_0 \leq r/2$,

iii) $\|f''(x)\| \leq C,\ x \in S_0$,

iv) $\eta_0 := 2nA_0B_0 \leq 1$,

then, for the starting value $x^{(0)}$, the iterative process (6.2.4) is convergent, the vector

$$x^* = \lim_{p\to\infty} x^{(p)}$$

is the solution of the given equation and one has

$$\|x^* - x^{(k)}\| \leq \tfrac{1}{2^{k-1}}\eta_0^{2^k-1}B_0.$$

Proof. We start showing that all the conditions **i**) − **iv**), valid for $x^{(0)}$, also hold for the approximation $x^{(1)}$, given by (6.2.4). Indeed, one has

$$\|x^{(1)} - x^{(0)}\| = \|w^{-1}(x^{(0)})f(x^{(0)})\| = \|M_0 f(x^{(0)})\| \leq B_0 \leq r/2,$$

whence $x^{(1)} \in S_1(x^{(0)}, r/2) \subset S_0$.

In order to prove the existence of $M_1 = w^{-1}(x^{1)})$, we use the property $(A \cdot B)^{-1} = B^{-1} \cdot A^{-1}$, and, thus, we obtain

$$M_1 = \omega^{-1}(x^{(1)}) = (w(x^{(0)})M_0 w(x^{(1)})^{-1} = (M_0 w(x^{(1)}))^{-1}M_0.$$

Using condition **i**), we can write

$$\begin{aligned}\left\|I - M_0 w(x^{(1)})\right\| &= \left\|M_0(w(x^{(0)}) - w(x^{(1)}))\right\| \\ &\leq \|M_0\| \left\|w(x^{(0)}) - w(x^{(1)})\right\| \\ &\leq A_0 \left\|w(x^{(1)}) - w(x^{(0)})\right\|.\end{aligned} \tag{6.2.5}$$

Using the inequality

$$\|f'(x+\Delta x) - f'(x)\| \leq n \|\Delta x\| \|f''(\xi)\|, \quad \xi = x + \alpha \Delta x,\ 0 < \alpha < 1$$

and conditions **iii**), **iv**), relation (6.2.5) becomes

$$\left\|I - M_0 w(x^{(1)})\right\| \leq n A_0 B_0 C = \eta_0/2 \leq 1/2. \tag{6.2.6}$$

The latter inequality assures the existence of the inverse matrix

$$[M_0 w(x^{(0)})]^{-1} = [I - (I - M_0 w(x^{(1)})]^{-1}$$

and one has

$$\left\|[M_0 w(x^{(1)})]^{-1}\right\| \leq \tfrac{1}{1-\eta_0/2} \leq 2.$$

Therefore, there exists the matrix

$$M_1 = [M_0 w(x^{(1)})]^{-1} M_0$$

and for its norm we get

$$\|M_1\| = \left\|[M_0 w(x^{(1)})]^{-1} M_0\right\| \leq \left\|M_0 w(x^{(1)})]^{-1}\right\| \|M_0\| \leq 2A_0 = A_1. \tag{6.2.7}$$

Thus condition **i**) is also valid for the approximation $x^{(1)}$.

Using formulas (6.2.4), yields

$$\begin{aligned}\left\|f(x^{(1)})\right\| &= \left\|f(x^{(1)}) - f(x^{(0)}) - f'(x^{(0)})(x^{(1)} - x^{(0)})\right\| \\ &\leq \tfrac{1}{2} \left\|x^{(1)} - x^{(0)}\right\|^2 \|f''(\xi)\| \leq \tfrac{1}{2} n B_n^2 C,\end{aligned} \tag{6.2.8}$$

where $\xi = x^{(0)} + \theta(x^{(1)} - x^{(0)}),\ 0 < \theta < 1$.

We mention that in order to obtain the first inequality above we used relation

$$\|f(x+\Delta x) - f(x) - f'(x)\Delta x\| \leq \tfrac{1}{2} n \|\Delta x\|^2 \|f''(\xi)\|,$$

with $\Delta x = x^{(1)} - x^{(0)}$.

From (6.2.7) and (6.2.8) it follows

$$\begin{aligned}\left\|M_1 f(x^{(1)})\right\| &\leq \|M_1\| \left\|f(x^{(1)})\right\| \leq 2A_0\tfrac{1}{2}nB_0C \\ &= nA_0B_0C = \tfrac{1}{2}\eta_0 B_0 = B_1,\end{aligned}$$

so condition **ii**) also holds for $x^{(1)}$.

Since condition **iv**) is verified, it remains to show $\eta_1 = 2nA_1B_1C \leq 1$. We have

$$\eta_1 = 2n2A_0\tfrac{1}{2}\eta_0 B_0 C = \eta_0 2nA_0B_0C = \eta_o^2 \leq 1.$$

Using the same argument, we deduce that the successive approximations $x^{(p)},\ p = 1, 2, ...$ generated by (6.2.4) can be defined, and one has $x^{(p)} \in S_p(x^{(p-1)}, r/2^{p-1})$, with

$$S_0(x^{(0)}, r) \supset S_1(x^{(1)}, r/2) \supset ... \supset S_p(x^{(p)}, r/2^p) \supset ...$$

Furthermore, we have

$$\begin{aligned}&\|M_p\| = \left\|w^{-1}(x^{(p)}\right\| \leq A_p, \\ &\left\|M_p f(x^{(p)})\right\| = \left\|x^{(p+1)} - x^{(p)}\right\| \leq B_p,\ p = 1, 2, ...,\end{aligned} \tag{6.2.9}$$

where the constants A_p, B_p verify the relations

$$\begin{aligned}A_p &= 2A_{p-1}, \\ B_p &= \tfrac{1}{2}\eta_{p-1}B_{p-1},\end{aligned} \tag{6.2.10}$$

with $\eta_p = 2nA_pB_pC$, $p = 1, 2, ...$

Now, we show that the sequence generated by (6.2.4) is fundamental. Indeed, one has

$$x^{(p+q)} \in S_p(x^{(p)}, r/2^p),\ q \in \mathbb{N}^*,$$

that is

$$\left\|x^{(p+q)} - x^{(p)}\right\| \leq r/2^p.$$

In other words, for every $\varepsilon > 0$, there exists $n_0 \in \mathbb{N}$, such that

$$\left\|x^{(p+q)} - x^{(p)}\right\| < \varepsilon, \quad p > n_0 \quad \text{and } q \in \mathbb{N}^*,$$

so $(x^{(p)})_{p\in\mathbb{N}}$ is a fundamental sequence. Since $x^{(p)} \in S_0(x^{(0)}, r),\ p = 0, 1, ...$ and $S_0(x^{(0)}, r)$ is a closed set, it follows that the above sequence is convergent. Let

$$\lim_{p\to\infty} x^{(p)} = x^* \in S_0(x^{(0)}, r).$$

In order to prove that x^* is the solution of equation (6.0.2), we write equality (6.2.4) in the form

$$w(x^{(p)})(x^{(p+1)} - x^{(p)}) = f(x^{(p)}).$$

Passing to the limit and taking into account the continuity of f and f', it follows $f(x^*) = 0$.

Using (6.2.9) and (6.2.10), we obtain

$$\left\|x^{(p+1)} - x^{(p)}\right\| \leq B_p = \tfrac{1}{2^p}\eta_0^{2^p-1}B_0,$$

which implies

$$\begin{aligned}\left\|x^{(k+p)} - x^{(k)}\right\| &\leq \left\|x^{(k+p)} - x^{(k+p-1)}\right\| + ... + \left\|x^{(k+1)} - x^{(k)}\right\| \\ &\leq B_0\left(\tfrac{1}{2^{k+p-1}}\eta_0^{2^{k+p-1}-1} + ... + \tfrac{1}{2^k}\eta_0^{2^k-1}\right) \\ &= \tfrac{B_0}{2^k}\eta_0^{2^k-1}\left(1 + \tfrac{1}{2}\eta_0^{2^k} + ... + \tfrac{1}{2^{p-1}}\eta_0^{2^k(2^{p-1}-1)}\right) \\ &\leq \tfrac{B_0}{2^k}\eta_0^{2^k-1}\left(1 + \tfrac{1}{2} + ... + \tfrac{1}{2^{p-1}}\right).\end{aligned}$$

Passing to the limit with respect to p, yields

$$\left\|x^* - x^{(k)}\right\| \leq \tfrac{B_0}{2^{k-1}}\eta_0^{2^k-1},$$

so the theorem is completely proved. ■

Chapter 7

Parallel calculus

7.1 Introduction

Parallel calculus, or parallel computation, is the process of solving problems on parallel computers. The 1980s will go down in history as the decade in which parallel computing first had a significant impact in the scientific and commercial worlds. The problems have arisen from real world demand computers that are many order of magnitude faster than the fastest conventional computers. Parallelism represents the most feasible avenue to achieve this kind of performances. The modern improvements in hardware have brought tremendous advances, e.g., by means of ultra large-scale integrated circuits (VLSI). So, the processor has born and today there exist many supercomputers which work with thousands and hundred of thousands processors, so the parallel algorithms have a strong technique support.

From the software point of view, many programs that run well on sequential computers (serial computers) are not easily transformed to programs that efficiently can be performed on parallel computers. Conversely, algorithms that are less efficient in a sequential context often reveal an inherent parallelism that makes then attractive to parallel programs.

Anyway, the parallel computation does not replace all the serial programs (of course, the best and the most efficient of then, remain), but it is a very attractive alternative in order to improve the speed-up of execution. In parallel calculus, the problem have to be rethink in order to be perform with more than one processor. This is the fascinating part of parallel calculus.

In the following section we try to indicate some parallel ways of solving

several of the numerical methods presented in previous chapters.

7.2 Parallel methods to solve triangular systems of linear equations

7.2.1 The column-sweep algorithm

Let us consider the system

$$A \cdot x = b$$

with $A \in M_{n\times n}(\mathbb{R})$ triangular matrix and $x, b \in M_{n\times 1}(\mathbb{R})$,

We begin by defining a linear recurrence system $R < n,\ m >$ of order m for n equations by

$$R < n, m >: \qquad x_k = \begin{cases} 0, & \text{if} \quad k \leq 0, \\ b'_k + \sum\limits_{j=k-m}^{k-1} a'_{kj} x_j, & \text{if} \quad 1 \leq k \leq n, \end{cases} \tag{7.2.1}$$

where $m \leq n-1$. If we write $A = [a_{ik}]$, with $a_{ik} = 0$ for $i \leq k$ or $i - k > m$, $x = [x_1, ..., x_n]^T$ and $b = [b_1, ..., b_n]^T$, then (7.2.1) can be written as:

$$x = A' \cdot x + b'. \tag{7.2.2}$$

Note: A' and b' can be easily obtained by A and b by division with nonzero elements.

Relation (7.2.2) is, e.g., $R < 4, 3 >$:

$$\begin{aligned} x_1 &= b'_1 \\ x_2 &= b'_2 + a'_{21} x_1 \\ x_3 &= b'_3 + a'_{31} x_1 + a'_{32} x_2 \\ x_4 &= b'_4 + a'_{41} x_1 + a'_{42} + a'_{43} x_3. \end{aligned} \tag{7.2.3}$$

In order to illustrate the column-sweep algorithm, we take a 4×4 system,

$$\overline{A} \cdot x = \overline{b},$$

with

$$\overline{A} = \begin{bmatrix} 1 & & & \\ -a_{21} & 1 & & \\ -a_{31} & -a_{32} & 1 & \\ -a_{41} & -a_{42} & -a_{42} & 1 \end{bmatrix}, \tag{7.2.4}$$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \qquad \overline{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}.$$

Using (7.2.2), we get

$$\begin{aligned} x_1 &= b_1 \\ x_2 &= b_2 + a_{21}x_1 \\ x_3 &= b_3 + a_{31}x_1 + a_{32}x_2 \\ x_4 &= b_4 + a_{41}x_1 + a_{42}x_2 + a_{43}x_3, \end{aligned} \tag{7.2.5}$$

from which we can evaluate x_1, x_2, x_3 and x_4 in parallel as follows:

Step 1 Evaluate, in parallel, expressions of the from

$$b_i^{(1)} = b_i + a_{i1}, \quad \text{for } i = \overline{2,n},$$

where $x_1 = b_1$ is known. Now there remain $n-2$ equations to solve.

Step 2. Evaluate, in parallel, expressions of the form

$$b_i^{(2)} = b_i^{(1)} + a_{i2}x_2, \quad i = \overline{3,n},$$

where x_1 and x_2 are known. Now there remain $n-3$ equations to be solved.

...

Step k. Evaluate, in parallel, expressions of the from

$$b_i^{(k)} = b_i^{(k-1)} + a_{ik} \cdot x_k, \qquad i = k+1, ..., n,$$

where $x_1, ..., x_k$.

Remark 7.2.1. Using a network with n processors, the execution time for the column-sweep algorithm is n times reduced comparing with the execution time involved in solving a triangular system on a serial computer.

7.2.2 The recurrent-product algorithm

Recalling the method presented in the previous section, we can express the solution x as a product of elementary matrices that can easily be computed using the recursive-doubling technique. Thinking in this way, Sameth and Brent gave the following method for solving a triangular system of equations, known also as the *recurrent-product algorithm*.

According with it, relation (7.2.2) can be written

$$x = (I - A')^{-1} \cdot b', \tag{7.2.6}$$

where matrix A' is strictly lower triangular and $(I - A')^{-1}$ can be expressed in the product from:

$$(I - A')^{-1} = \prod_{i=1}^{n-1} M_{n-i}, \tag{7.2.7}$$

where

$$M_i = \begin{bmatrix} 1 & & & 0 & & \\ & \ddots & & & & \\ & & 1 & & & \\ 0 & & a_{i+1,i} & 1 & & \\ & & \vdots & & \ddots & \\ & & a_{n,i} & 0 & & 1 \end{bmatrix}.$$

The product $\prod_{i=1}^{n-1} M_{n-i}$ can be carried out using the recursive-doubling technique, on a binary tree connectivity among processors.

Example 7.2.2. Suppose we take a banded lower triangular system $R\langle 5, 2\rangle$. Then, relation (7.2.6) is:

$$\begin{aligned} x_1 &= b'_1 \\ x_2 &= b'_2 + a'_{21}x_1, \\ x_3 &= b'_3 + a'_{31}x_1 + a'_{32}x_2 \\ x_4 &= b'_4 \qquad\quad + a'_{42}x_2 + a'_{43}x_3 \\ x_5 &= b'_5 \qquad\qquad\quad + a'_{53}x_3 +'_{54}\, x_4. \end{aligned} \tag{7.2.8}$$

Hence,

$$x = M_4 \cdot M_3 \cdot M_2 \cdot M_1 \cdot b', \tag{7.2.9}$$

where

$$M_4 \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & 0 \\ & 0 & & 1 & \\ & & & a'_{54} & 1 \end{bmatrix}; \; M_3 = \begin{bmatrix} 1 & & & & \\ & 1 & & 0 & \\ & & 1 & & \\ 0 & & a'_{43} & 1 & \\ & & a'_{53} & 0 & 1 \end{bmatrix}$$

$$M_2 = \begin{bmatrix} 1 & & & 0 & \\ 0 & 1 & & & \\ 0 & a'_{32} & 1 & & \\ 0 & a'_{42} & 0 & 1 & \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}; \; M_1 = \begin{bmatrix} 1 & & & & \\ a'_{21} & 1 & & 0 & \\ a'_{32} & & 1 & & \\ 0 & 0 & & 1 & \\ 0 & & & & 1 \end{bmatrix}.$$

The matrix product in (7.2.9) can be evaluated using the recursive-doubling, as follows: $(((M_4M_3)(M_2M_1)) \cdot b')$ which means that we compute all the inner brackets at **step1**, then all "next" brackets at **step2**, and so on. The computations of the inner brackets, etc. are made simultaneously, by two processors. The whole computation needs three parallel steps.

Remark 7.2.3. The problem of inverting of a lower triangular matrix can be made, with parallel calculus, in several other ways. For example, Borodin and Munro proposed a method that use date partitioning.

7.3 Parallel approaches of direct methods for solving systems of linear equations

In this section we discuss several techniques for solving linear algebraic systems which are particularly suited to parallel computation. It should be noted that the choice of method depends on the type of parallel environment used.

7.3.1 The parallel Gauss method

Let us consider the system

$$A \cdot x = b, \tag{7.3.1}$$

with $A \in \mathcal{M}_{n\times n}(\mathbb{R})$, $x, b \in \mathcal{M}_{n\times 1}(\mathbb{R})$,or, in the explicitly from:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n = b_2 \\ \qquad\qquad ... \\ a_{n1}x_1 + a_{n2}x_2 + ... + a_{nn}x_n = b_n. \end{cases} \tag{7.3.2}$$

A possible parallel execution with n processors is the following:

Stage 1: {all n processors work together}

Step 1: Considering $a_{11} \neq 0$,

for $j : \overline{2,}n$ in parallel do

$a_{1j}^{(1)} := a_{1j}/a_{11}; b_1^{(1)} = b_1/a_{11}$.

Step 2: for $j := \overline{2,}n$ in parallel do

$a_{ij}^{(1)} = a_{ij} - a_{1j}^{(1)} * a_{i1}; b_i^{(1)} = b_i - b_1^{(1)} * a_{i1}$.

After **Step 1** and **Step 2,** the following system is obtained:

$$\begin{cases} x_1 + a_{12}^{(1)}x_2 + ... + a_{1n}^{(1)}x_n = b_1^{(1)} \\ \qquad a_{22}^{(1)}x_2 + ... + a_{2n}^{(1)}x_n = b_2^{(1)} \\ \quad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ \qquad a_{n1}^{(1)}x_2 + ... + a_{nn}^{(1)}x_n = b_n^{(1)}. \end{cases} \tag{7.3.3}$$

Stage 2: {$n-1$ processors work together}

Repeat computations of type from **Step 1** and **Step 2**, with $a_{22} \neq 0$

$\vdots$

Stage $n-1:$ {2 processors work together}

Repeat computations of type from **Step 1** and **Step 2**, with $a_{n-1,n-1} \neq 0$.

Finally, we get the system:

$$\begin{cases} x_1 + a_{12}^{(1)} x_2 + ... + a_{1,n-1}^{(1)} x_{n-1} + a_{1n}^{(1)} x_n = b_1^{(1)} \\ x_2 + ... + a_{2,n-1}^{(2)} x_{n-1} + a_{2n}^{(2)} x_n = b_2^{(2)} \\ \dots \\ x_{n-1} + a_{n-1,n}^{(n-1)} x_n = b_{n-1}^{(n-1)} \\ a_{nn}^{(n-1)} x_n = b_n^{(n-1)}. \end{cases} \tag{7.3.4}$$

So, the problem has reduced to solving of the triangular system (7.3.4). One way to do it in parallel is by means of techniques presented in section 7.2. Another possibility is to reactivate the processors, one by one, and to get the solution starting with the last equation.

Remark 7.3.1. The Gauss method can be parallelized, as we saw, but the final algorithm is not very efficient, because the n processors are not all of them used, at every step. From this point of view, the Gauss-Jordan method is more appropriate for the parallel calculus.

7.3.2 The parallel Gauss-Jordan method

There are several possibilities to perform the Gauss-Jordan method by using more than one processor. So, considering the system

$$A \cdot x = b, \tag{7.3.5}$$

with $A \in M_{n\times n}(\mathbb{R}), x, b \in M_{n\times 1}(\mathbb{R})$, $a_{ii} \neq 0$, $i = \overline{1,n}$, and n processors on a SIMD type machine, the parallel method is the following:

Step 1. The system (7.3.5) is transformed in:

$$\begin{cases} x_1 + a_{12}^{(1)} x_2 + ... + a_{1n}^{(1)} x_n = b_1^{(1)} \\ a_{22}^{(1)} x_2 + ... + a_{2n}^{(1)} x_n = b_2^{(1)} \\ \vdots \qquad\qquad \vdots \qquad\quad \vdots \\ a_{n2}^{(1)} x_2 + ... + a_{nn}^{(1)} x_n = b_n^{(1)}, \end{cases} \tag{7.3.6}$$

where

$$a_{1j}^{(1)} = a_{1j}/a_{11},\ j = \overline{2,n},\ b_1^{(1)} = b_1/a_{11}$$
$$a_{ij}^{(1)} = a_{ij} - a_{1j}^{(1)} \cdot a_{i1},\ b_i^{(1)} = b_i - b_1^{(1)} \cdot a_{i1}, \quad i,j = \overline{2,n}.$$

The n processors work as follows: one processor computes $a_{12}^{(1)}, a_{13}^{(1)}, ..., a_{1n}^{(1)}$, $b_1^{(1)}$, one after another. In the moment where $a_{12}^{(1)}$ is computed, the other $n-1$ processors compute the elements of the second column: $a_{i2}^{(1)}$, $i = \overline{2,n}$, then the elements of the third column, $a_{13}^{(1)}$, etc., until they compute also $b_i^{(1)}$, $i = \overline{2,n}$.

Step 2. Meanwhile, the first processor, which already finished the computation of $b_1^{(1)}$, continues with $a_{23}^{(2)} = a_{23}^{(1)}/a_{22}^{(1)}$. The rest of $n-1$ processors will start the computation which eliminated x_2, it means the values

$$a_{13}^{(2)} = a_{13}^{(1)} - a_{12}^{(1)} \cdot a_{23}^{(2)}, b_1^{(2)} = b_1^{(1)} - a_{12}^{(1)} \cdot b_2^{(2)}$$
$$a_{i3}^{(2)} = a_{i3}^{(1)} - a_{i2}^{(1)} \cdot a_{23}^{(2)}, b_i^{(2)} = b_i^{(1)} - a_{i2}^{(1)} \cdot b_2^{(2)}, \quad i = \overline{3,n}.$$

And so on. After n steps, we get the solution:

$$x_i = b_i^{(n)}, \ i = \overline{1,n}. \tag{7.3.7}$$

Remark 7.3.2. Another parallel approach can be made. Instead of dividing the amount of work among one processor (which performs only a division operation, all the time) and the other $n-1$ processors (which perform a multiplication and an addition, all the time), Modi suggests a parallel execution in which all n processors work together, on the same column, in order to eliminate the elements upon and under the element a_{ii}, $i = \overline{1,n}$, so, after n steps, we get the final from (7.3.7) of the system.

7.3.3 The parallel LU method

To get a parallel LU Method, the idea of parallel Gauss method (see section 7.3.1) can also be used to get the matrices L and U of the decomposition

$$A = L \cdot U.$$

Another possibility is to use the double recursive technique to get the matrix product which form matrix U, and then matrix L, as there was presented in section 4.2.1.4, in the case of Doolittle algorithm. Also, as we already stated, the Block LU decomposition, presented in section 4.2.1.5, is more appropriate to the parallel calculus, due to the fact that the operation in the block-matrices can be perform simultaneously.

7.3.4 The parallel QR method

Recalling the QR method, presented in section 4.2.1.7, a parallel approach can be obtain by using the Givens rotation, because several rotations can be performed simultaneously. So, if we use a different notation and denote by (p_k, q_k), $k = 1$, $mn - \frac{1}{2}n^2 - \frac{1}{2}n$, if $A \in \mathcal{M}_{n\times n}(\mathbb{R})$ the fact that a Givens rotation $G(P_k, q_k)$ is performed between the rows P_k and $P_k - 1$, in order to annihilate the (P_k, q_k) element of

$$G(p_{k-1}, q_{k-1})...G(p_1, q_1)A.$$

According to Gentelman, the set S of all this rotations is partitioned into subsets S_r, $r = \overline{1, m+n-2}$, so that

$$S_r = \{(p, q)\,|1 \leq q < p \leq m,\ q \leq n,\ m + 2q = p + r + 1\}.$$

The rotations S_r are disjoint, and therefore they can be performed simultaneously. The corresponding product matrix

$$Q^{(r)} \prod \{G(p, q)\,|(p, q) \in S_r\},$$

$$Q^T = Q^{(n)}...Q^{(1)}, \quad Q^T \cdot A = R,$$

hence we get

$$A = Q \cdot R.$$

7.3.5 The parallel WZ method

Recalling the WZ Method (see section 4.2.1.8), we notice that the matrix A can be factorized

$$A = WZ,$$

where

$$W = \begin{pmatrix} 1 & & & & & 0 \\ w_{21} & 1 & \ddots & & 0 & w_{2n} \\ \vdots & & & 1 & & \vdots \\ w_{n-1,1} & 0 & & \ddots & 1 & w_{n-1,n} \\ 0 & & & & & 1 \end{pmatrix}$$

and

$$Z = \begin{pmatrix} z_{1,1} & & & \cdots & & z_{1,n} \\ 0 & z_{2,2} & & \cdots & z_{2,n-1} & 0 \\ \vdots & 0 & \ddots & z_{i,j} \quad \cdots & & \\ 0 & & & & \ddots & \\ z_{n,1} & & & \cdots & & z_{n,n} \end{pmatrix},$$

with elements w_{ij} and z_{ij}, $i, j = \overline{1,n}$ computed according to the formulas presented in section 4.2.1.8. The system $A \cdot x = b$, it means $WZ \cdot x = b$, can be solved in two steps:

$$W \cdot p = b, \tag{7.3.8}$$

and we get the n-elements vector p, then

$$Z \cdot x = p, \tag{7.3.9}$$

which generates the solution x. From (7.3.8), we have

$$\begin{pmatrix} 1 & & & & & 0 \\ w_{21} & 1 & & & 0 & w_{2n} \\ \vdots & & \ddots & & & \\ & & & 1 & & \\ & & & \ddots & 1 & w_{n-1,n} \\ w_{n-1,1} & & & & & \\ 0 & & & & & 1 \end{pmatrix} \cdot \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_{n-1} \\ p_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{pmatrix}, \tag{7.3.10}$$

hence, first we compute p_1 and ρ_n, then p_2 and p_{n-1}, and so on, (each time working from the top and the bottom). At the i-th step we have

$$p_i = b_i^{(1)},\ p_{n-i+1} = b_{n-i+2}^{(i)},$$

where

$$b_j^{(1)} = b_j$$

and

$$b_j^{(i+1)} = b_j^{(i)} - w_{j,i}.p_i - w_{j,n-i+1}.p_{n-i+1}\ (j = i+1, ..., n-i). \tag{7.3.11}$$

Relations (7.3.11) can be evaluated in parallel, using the double recursive technique. Then, in order to solve (7.3.9), the computation of x is similar, so it can be made, also, in parallel.

7.4 Parallel iterative methods

As we saw in section 4.3, the iterative methods for solving a linear system of equations are useful when the dimension of the problem that has to be solved increases, and so the direct methods become almost impractical. From a parallel execution point of view, the iterative methods are more attractive, in spite of the fact that they do not guarantee a solution for every system of equations due to the convergent of the method. In what follows, we shall give parallel approaches for some iterative methods.

7.4.1 Parallel Jacobi method

Recalling the expressions for computing the components of vector x at the step k, starting with an initial value $x^{(0)}$ (see section 4.3.1)

$$x_i^{(k)} = (b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij} \cdot x_j^{(k-1)})/a_{ii},\ i = \overline{1,n} \text{ with } a_{ii} \neq 0,$$

there exist several ways to perform them using more than one processor.

If we work with n processors, every processor i can compute a value x_i at a given step k. The only problem, here, is that the processors have to communicate among then, because, at every step, every processor needs to know all the other values computed by all the other processors.

Another parallel approach for Jacobi method was given by Wong.

In order to present it, we rewrite the method in matricial from (see also formula (4.3.7)):

$$\begin{aligned}
&A \cdot x = b \\
&b - A \cdot x = r \\
&A = L + D + U \\
&A \cdot x = (L + D + U)x + b \\
&Dx^{(k+1)} = -(L + U)x^{(k)} + b \\
&x^{(k+1)} = -(L + U)x^{(k)}/D + b/D \\
&x^{(k+1)} = -(L + D + U)x^{(k)}/D + b/D + x^{(k)} \\
&x^{(k+1)} = x^{(k)} + r^{(k)}/D.
\end{aligned}$$

Remark 7.4.1. The method is presented by means of the residual error.

Algorithmically, the Jacobi method of this form is the following:

Remark 7.4.2. Step 1. $k = 0$

Step 2. Initialize $x^{(0)}$.

Step 3. $r^{(0)} = b - A \cdot x^{(0)}$

Step 4. While $(\left\|r^{(k)}\right\| > \varepsilon)$ do

Step 5. $k = k + 1$

Step 6. for $i = 1$ to n do

Step 7. $x_i^{(k)} = r^{(k-1)}/a_{ii} + x_i^{(k-1)}$

Step 8. end for

Step 9. $r^{(k)} = b - A \cdot x^{(k)}$

Step 10. $\left\|r^{(k)}\right\| = (\sum_{i=1}^{n}(r_i^{(k)})^2)^{\frac{1}{2}}$

Step 11. end while

Step 12. $x = x^{(k)}$.

Remark 7.4.3. We have effective computation in steps 7,9 and 10.

The parallel implementation of this algorithm, using n processors, is the following:

Step 0: Initialize k and $x^{(0)}$.

Step 1 and 2: Assign the $x^{(0)}$value to n processors.

Step 3: $r = b - A \cdot x$ matrix-vector multiplication, computation made by all processors, without any communication among them.

Step 3+: Update vector r. Communication is required.

Step 4 and 10: Calculate residue $\|r\|$. Communication is required.

Step 7: Update value of vector x. Communication is required

Step 9: Calculate $r = b - A \cdot x$, matrix-vector multiplication.

Step 9+: Update vector r; communication is required.

Remark 7.4.4. For parallel matrix multiplication, see the bibliography.

7.4.2 Parallel Gauss-Seidel algorithm

As for the Jacobi method, there are several parallel approaches for the Gauss-Seidel method.

For instance, the following idea is used: having n processors, and due to the fact that the computation of a component $x_i^{(k)}$, $i = \overline{1,n}$ involves some values already computed at the same iteration k, according to the formulas:

$$x_i^{(k)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} \cdot x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} \cdot x_j^{(k-1)} \right) i = \overline{1,n}$$

all the processors have to work together in order to determine the value of one $x_i^{(k)}$.

P. Wong presented another possibility, by means of the residual value.

The Gauss-Seidel iteration is given as follows:

$$\begin{aligned}
&Ax = b \\
&b - Ax = r,\ A = (L + D + U) \\
&Ax = (L + D + U)x = b \\
&(D + L)x = -(U)x + b \\
&(D + L)x^{(k+1)} = -(U)x^{(k)} + b \\
&x^{(k+1)} = -(D + L)^{-1} \cdot U \cdot x^{(k)} + (D + L)^{-1} \cdot b \\
&x^{(k+1)} = -(D + L)^{-1} \cdot (U + D + L)x^{(k)} + x^{(k)} + (D + L)^{-1} \cdot b \\
&x^{(k+1)} = x^{(k)} + (D + L)^{-1} \cdot r_k.
\end{aligned} \tag{7.4.1}$$

In algorithmic manner, the relations (7.4.1) are the following:

Step 1: $k = 0$

Step 2: initialize x_0

Step 3: $r_0 = b - A \cdot x_0$

Step 4: while $(\|r_k\| > \varepsilon)$ do

Step 5: $k = k + 1$

Step 6: for $i = 1$ to n do

Step 7: $x_i^{(k)} = x_i^{(k-1)} + (L + D)^{-1} r_{k-1}$

Step 8: end for

Step 9: $r_k = b - A \cdot x_k$

Step 10: $\|r_k\| = \left(\sum_{i=1}^{n} r_k^2(i) \right)^{\frac{1}{2}}$

Step 11: end while

Step 12: $x = x_k$.

Remark 7.4.5. The Gauss-Seidel algorithm appears to be identical with Jacobi's, however, here **step7** is completely sequential because x_i cannot be computed until x_{i-1} has been computed. In fact, **step7** can be rewritten as:

$$\begin{aligned} x_i^{(k)} &= \left(b_i - \sum_{j=1}^{i-1} a_{ij} \cdot x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} \cdot x_j^{(k-1)} \right) / a_{ii} \\ &= \left(\left(b - U x^{(k-1)} \right) \right) - L x^{(k)} \Big) / a_{ii}. \end{aligned}$$

Hence, we can reuse only portion of the residue r calculation. These calculations can be done in parallel.

Remark 7.4.6. The Gauss-Seidel method used to solve discretized problems attached to PDE equations generates different type of parallel execution, e.q. the checker-board execution (or red-black execution). The type used takes into account the numbering of the nodes in the discretized network and their updating. These ideas may be found also in the next section.

7.4.3 The parallel SOR method

As we stated in the previous section, the different types of parallel execution can be shown also by means of the modified Gauss-Seidel method, which is SOR method.

As we have seen in section 4.3.3, using a parameter ω $(0 < \omega < 2)$ to improve the convergence speed, the values of x are updated as follows:

$$x_i^{(k)} =(1-\omega)x_i^{(k-1)} + \omega(b_i - \sum_{j=1}^{i-1} a_{ij} \cdot x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} \cdot x_j^{(k-1)})/a_{ii}. \tag{7.4.2}$$

In order to get a good parallelization of the computations (7.4.2), we "color" the matrix A in such a way that no two neighboring nodes are connected so that update information are independent, and then they can be made simultaneously.

Example 7.4.7. Consider a $1D$ discretized problem, with information in the points:

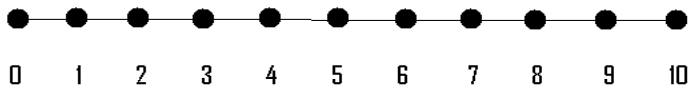


Figure 7.1: The set of points.

Then, the computation of values given by (7.4.2) can be carried on in two stages, according with the red-black type of execution:

Example 7.4.8. Stage 1: Work on even number of node points,

(so only node 2,4,6,8 are updated)

for $i = 1$ to n in parallel do

$$x_i^{(k)} = \left(b_i - \sum_{j-1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} \cdot x_j^{(k-1)} \right) / a_{ii}$$

end

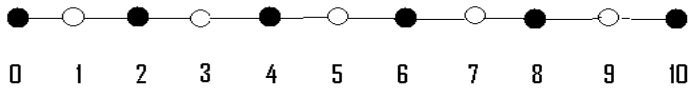


Figure 7.2: The set of even points.

Stage 2: Work on odd number of node points, (so only node 1,3,5,7,9 are updated)for $i = 1$ to n in parallel do

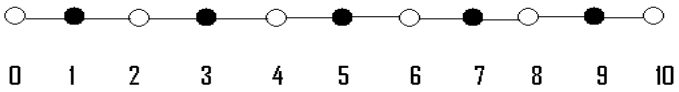


Figure 7.3: The set of odd points.

$$x_i^{(k)} = \left(b_i - \sum_{j=1}^{i-1} a_{ij} \cdot x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} \cdot x_j^{(k-1)} \right) / a_{ii}$$

end for.

7.5 The parallelization of the multigrid method

C.C. Douglas considered that two major classes of methods for parallelization of the multigrid method are used: telescoping and nontelescoping.

7.5.1 Telescoping parallelizations

As we have seen in section 4.3.6, the multigrid method implies the passing of information among several grids. At every step, some values in the grid nodes have to be computed.

One way of doing it in parallel is by means of *domain decomposition techniques*. Then, each processor computes on the block of data per grid that is assigned to it. Of course, with or without overlaps, they have to communicate at the borders, especially if the smoothing procedure is of Gauss-Seidel type.

Another approach to telescoping multigrid involves massively parallel computers (it means at least as many processors as unknowns on all of the grids).

Then, the following method is proposed, known as "*the concurrent mutigrid*" method: the concept is that all operations should be perform simultaneously, on all unknowns, on all grids. An initial approximation of zero is assumed for the solution. Two sets of vectors, q_j and d_j are used to hold information about right hand sides and data on the spectrum of levels in the sense that

$$b_k \simeq \sum_{j=1}^{k} q_1 + d_j.$$

A third set of vectors, x_j, contains the approximations to the solutions to each problem on each level. This information percolates to the finest grid, k, to finally provide the approximate solution to the real problem. Recalling the serial Multigrid Algorithm (MV) given in section 4.3.6, in what follows we denote the restriction operator, I_h^{2h}, by R and the prolongation operator, I_{2h}^h, by P. Also, the level h and $2h$, will be denoted, respectively, by j and $j-1$.

So, the parallel algorithm is the following:

Step 1: Initialize in parallel

$$x_j = q_j = 0, \quad 1 \leq j < k; \quad g_k = b_k, \quad x_k = 0.$$

Step 2: Repeat $(i = 1, ..., \mu)$.

Step 2a: Smoothing in parallel:

$$x_j \leftarrow \ PreSolve\ (x_j, q_j), \quad 1 \leq j \leq k.$$

Step 2b: Compute data corresponding to x_j in parallel:

$$d_j \leftarrow A_j \cdot x_j, \quad 1 \leq j \leq k.$$

Step 2c: Compute residuals in parallel:

$$q_j \leftarrow q_j - d_j, \ 1 \leq j \leq k.$$

Step 2d: Project q onto coarser grids in parallel.

$$q_1 \leftarrow q_1 + R_2 q_2;$$
$$q_k \leftarrow (I - P_{k-1}R_k)q_k;$$
$$q_j \leftarrow (I - P_{j-1}R_j)q_j + R_j q_{j+1},\ 1 < j < k.$$

Step 2e: Inject x into finer grids in parallel:

$$x_1 \leftarrow 0;\ x_k \leftarrow x_k + P_{k-1} \cdot x_{k-1};$$
$$x_j \leftarrow P_{j-1} \cdot x_{j-1},\ 1 < j < k.$$

Step 2f: Inject d into finer grids in parallel:

$$d_1 \leftarrow 0;\ d_k \leftarrow d_k + P_{k-1} \cdot d_{k-1};$$
$$d_j \leftarrow P_{j-1} \cdot d_{j-1},\ 1 < j < k.$$

Step 2g: Put all the data back into q in parallel

$$g_j \leftarrow g_j + d_j,\ 1 \leq j \leq k.$$

Step 3: Return x_k.

Finally, there is another approach to telescoping parallelization: apply a domain decomposition method to the finest grid and then use a serial computer multigrid method on each of the subdomains.

7.5.2 Nontelescoping parallelizations

Nontelescoping methods use multiple coarse subspaces in which the sum of the unknowns on any level equals the sum on any other level.

For a given problem k, it either has a set C_k of coarse space correction problems or it has none at all (i.e. $C_k = \emptyset$). When $C_k \neq \emptyset$, there are restrictions and prolongations operators for each space problem $\ell \in C_k$ such that

$$R_\ell : M_k \to M_\ell \text{ and } P_\ell : M_\ell \to M_k,$$

where by M_ℓ and M_k we denoted the solutions spaces.

A multiple coarse space correction multigrid scheme is defined by the following algorithm, named MCM:

Algorithm MCM$(j, \mu_j, C_j, x_j, b_j)$

Step 1: If $C_j = \emptyset$, then solve $A_j \cdot x_j = b_j$.

Step 2: If $C_j \neq \emptyset$, then repeat $(i = 1, ..., \mu_j)$.

Step 2a: Smoothing: $x_j \leftarrow PreSolve(x_j, b_j)$.

Step 2b: Residual correction:

$$x_j \leftarrow x_i + \sum_{\ell \in C} P_\ell \cdot MCM(\ell, \mu_\ell, C_\ell, O, R_\ell(b_j - A_j x_j))/$$

Step 2c: Smoothing: $x_j \leftarrow PreSolve(x_j, b_j)$.

Step 3: Return x_j.

Remark 7.5.1. In the parallel implementation of MCM algorithm, each processor executes the computation on a C_j coarse space.

This algorithm was introduced by Ta'asan, when standard multigrid failed to converge for a class of problems with highly oscillatory solutions. He used standard interpolation and projection methods for operators (as presented in section 4.3.6) and a relaxation method for the smoothing procedure.

Using a different type of interpolation and projection methods, Hackbusch developed another method, known as *robust multigrid*.

Frederichson and McBryan elaborated the parallel *superconvergent multigrid method*, using a SIMD machine, standard interpolation and projection operators, but an elaborate smoother on every level.

Remark 7.5.2. All these methods reduce the execution time, but are space-wasteful.

In order to correct this inconvenient Douglas proposed the *domain reduction method*, in which the approximate solve step was eliminated. A side benefit of this theory is that the fine grid problem and most of the coarse grid matrices do not need to be generated, so the method can use substantially less memory than a standard iterative algorithm.

Bibliography

[1] O. Agratini, I. Chiorean, Gh. Coman, R.T. Trîmbiţaş, *Analiză Numerică şi Teoria Aproximării,* vol. III, Presa Universitară Clujeană, 2002.

[2] O. Agratini, P. Blaga, Gh. Coman, *Lectures on Wavelets, Numerical Methods and Statistics*, Casa Cărţii de Ştiinţă, Cluj-Napoca, 2005.

[3] R.E. Barnhill, *Representation and Approximation of Surfaces*, Mathematical Software III, (Ed. J.R. Rice), Academic Press, 1977, pp. 69–120.

[4] R.E. Barnhill, *A survey of the representation and design of surfaces,* IEEE Computer Graphics and Applications, **3** (1983), no. 7, pp. 9–16.

[5] R.E. Barnhill, *Surfaces in Computer Aided Geometric Design: A survey with new results,* Computer Aided Geometric Design, **2** (1985), pp. 1–17.

[6] R.E. Barnhill, G. Birkhoff, W.J. Gordon, *Smooth interpolation in triangle*, J. Approx. Theory **8** (1973), pp. 114–128.

[7] R.E. Barnhill, R.P. Dube, F.F Little, *Properties of Shepard's surfaces*, Rocky Mountains J. Math., **13** (1983), no. 2, pp. 365–382.

[8] S. Bernstein, *Demonstration du théorème de Weierstrass fondée sur le calcul des probabilités*, Comm. Soc. Math., Khavkov, **13** (1912), pp. 1-2.

[9] J. Bernoulli, *Ars conjectandi*, Werke, 3, Birkhäser, 1975, pp. 107-286.

[10] P. Blaga, Gh. Coman, *On some bivariate spline operators*, Rev. Anal. Numér. Théor. Approx., **7** (1979), pp. 143–153.

[11] P. Blaga, Gh. Coman, *Multivariate interpolation formulae of Birkhoff type*, Studia Univ. Babeş-Bolyai, Mathematica, **26** (1981), no. 2, pp. 14–22.

[12] P. Blaga, Gh. Coman, S. Pop, R. Trîmbiţaş, D. Văsaru, *Analiză numerică-Lucrări de laborator*, Univ. Babeş-Bolyai, Cluj, 1994.

[13] H. Bohman, *On approximation of continuous and analytic functions*, Ark. Mat. 2, 1952, pp. 43-56.

[14] K. Bohmer, *Spline functionen eine einfuhrung in theorie und anwendungen*, Teubner Valag, Stuttgart, 1974.

[15] K. Böhmer, Gh. Coman, *Smooth interpolation schemes in triagles with error bounds*, Mathematica (Cluj), **18 (41)** (1976), no. 1, pp. 15–27.

[16] K. Böhmer, Gh. Coman, *On some approximation schemes on triangles,* Mathematica (Cluj), **24 (45)** (1980), pp. 231–235.

[17] A. Borodin, I. Munro, *The computational complexity of algebraic and numerical problems*, American Elsevier, 1975.

[18] B. Bylina, *Solving linear systems with vectorized WZ factorization*, Annales UMCS Informatica AI, **1** (2003), pp. 5-13.

[19] T. Cătinaş (Gulea), *On trivariate approximation*, Proceedings of the International Symposium on Numerical Analysis and Approximation Theory, Cluj-Napoca, May 9–11, 2002, pp. 207–230.

[20] T. Cătinaş, *Trivariate approximation operators on cube by parametric extensions*, Acta Universitatis Apulensis, Mathematics-Informatics, no. 4 (2002), pp. 29–36.

[21] T. Cătinaş, *Interpolating on some nodes of a given triangle*, Studia Mathematica "Babeş-Bolyai", **48** (2003), no. 4, pp. 3–8.

[22] T. Cătinaş, *The combined Shepard-Abel-Goncharov univariate operator*, Rev. Anal. Numér. Théor. Approx., **32** (2003), no. 1, pp. 11–20.

[23] T. Cătinaş, *The combined Shepard-Lidstone univariate operator*, "Tiberiu Popoviciu" Itinerant Seminar of Functional Equations, Approximation and Convexity, Cluj-Napoca, May 21–25, 2003, pp. 3–15.

[24] T. Cătinaş, *On the generalized Newton algorithm for some nodes given on tetrahedron*, Seminar on Numerical and Statistical Calculus, "Babeş-Bolyai" University, Cluj-Napoca, 2004, pp. 65–76.

[25] T. Cătinaş, *The combined Shepard-Lidstone bivariate operator*, Trends and Applications in Constructive Approximation, (Eds. M.G. de Bruin, D.H. Mache, J. Szabados), International Series of Numerical Mathematics, Vol. 151, 2005, Springer Group-Birkhäuser Verlag, pp. 77-89.

[26] T. Cătinaş, *Bounds for the remainder in the bivariate Shepard interpolation of Lidstone type*, Rev. Anal. Numér. Théor. Approx., **34** (2005), no. 1, pp. 47-53.

[27] T. Cătinaş, *Bivariate interpolation by combined Shepard operators*, Proceeding of 17^{th}IMACS World Congress, Scientific Computation, Applied Mathematics and Simulation (Eds. P. Borne, M. Benrejeb, N. Dangoumau, L. Lorimier), Paris, July 11-15, 2005, ISBN 2-915913-02-1, pp.1-7.

[28] T. Cătinaş, *Three ways of defining the bivariate Shepard operator of Lidstone type*, Studia Univ. "Babeş-Bolyai", Mathematica, **50** (2005), no. 3, pp. 57–63.

[29] T. Cătinaş, *The Lidstone interpolation on tetrahedron*, J. Appl. Funct. Anal., 2006, no. 4, pp. 425-439.

[30] T. Cătinaş, *A combined method for interpolation of scattered data based on triangulation and Lagrange interpolation*, Studia Univ. Babeş-Bolyai, Mathematica, **51** (2006), no. 4, pp. 55-63.

[31] T. Cătinaş, *Interpolation of scattered data*, "Casa Cărţii de Ştiinţă", 2007.

[32] T. Cătinaş, *The bivariate Shepard operator of Bernoulli type*, Calcolo, 2007, to appear.

[33] T. Cătinaş, *A cubature formula based on Bernoulli interpolation for the rectangle*, to appear.

[34] T. Cătinaş, Gh. Coman, *Optimal quadrature formulas based on φ-function method*, Studia Univ. Babes-Bolyai, Mathematica, **51** (2006), no. 1, pp. 49-64.

[35] E.W. Cheney, *Multivariate Approximation Theory*, Selected Topics, CBMS51, SIAM, Philadelphia, Pennsylvania, 1986.

[36] W. Cheney, W. Light, *A Course in Approximation Theory*, Brooks/Cole Publishing Company, Pacific Grove, 2000.

[37] I. Chiorean, *Parallel Numerical Methods for Solving Partial Differential Equations*, Matarom Rev., Nr. 3, Paris, 1993, pp. 19-38.

[38] I. Chiorean, *On the complexity of multigrid method in solving a system of PDE,* Research seminars, Seminar on Numerical and Statistical Calculus, Preprint nr. 1, 1994, pp. 27-32.

[39] I. Chiorean, *Calcul Paralel,* Editura Microinformatica, 1995.

[40] I. Chiorean, *On the Convergence of the Numerical Solution for a Problem of Convection in Porous Medium,* Research seminars, Seminar on Numerical and Statistical Calculus, Preprint nr. 1, 1996, pp. 23-27.

[41] I. Chiorean, *Parallel algorithm for solving a problem of convection in porous medium,* Advances in Engineering Software, Elsevier Ltd., **28** (1997), no. 8, Great Britain, pp. 463-467.

[42] I. Chiorean, *Free convection in an inclined square enclosure filled with a heat-generating porous-medium,* Studia Universitatis "Babeş-Bolyai" Mathematica, Cluj-Napoca, no. 4, 1997, pp. 35-43.

[43] I. Chiorean, *Convection between the Number of Multigrid Iterations and the Discretized Error in Solving a Problem of Convection in Porous Medium,* Studia Universitatis "Babeş-Bolyai" Mathematica, Cluj-Napoca, no. 1, 1997, pp. 59-65.

[44] I. Chiorean, *Parallel Numerical methods for solving nonlinear equations,* Studia Universitatis "Babeş-Bolyai", Cluj-Napoca, Mathematica, **46** (2001), no. 4, pp. 53-61.

[45] I. Chiorean, *On the complexity of some parallel relaxation schemes,* Bul. Stiinţific Univ. Baia Mare, Ser. B, Matematica-Informatica, **43** (2002), no. 2, pp. 171-176.

[46] I. Chiorean, *On some 2D Parallel Relaxation Schemes,* Proceedings of ROGER, Sibiu, 12-14 iunie 2002, Mathematical Analysis and Approximation Theory, Burg Verlag, 2002, pp. 77-85.

[47] I. Chiorean, *Contributii la studiul unei probleme de convectie in mediu poros,* Presa Universitară Clujeană, 2002.

[48] I. Chiorean, *Parallel numerical methods for PDE,* Kragujevac J. Math. **25** (2003), pp. 5-18.

[49] I. Chiorean, *On some Numerical Methods for Solving the Black-Scholes Formula,* Creative Mathematics Journal, **13** (2004), Pub. by Dep. of Math. and Comp. Science, North Univ. Baia-Mare, pp. 31-36 (conf. ICAM4, Suior, Baia-Mare).

[50] I. Chiorean, *Remarks on the restriction and prolongation operators from the multigrid method,* Mathematical Analysis and Approximation Theory, Mediamira Science Publisher, 2005, pp. 43-49.

[51] I. Chiorean, *Parallel Algorithm for Solving the Black-Scholes Equation,* Kragujevac J. Math, **27** (2005), pp. 39-48.

[52] I. Chiorean, *Parallel LU Method for solving the Black-Scholes Equation,* Proceedings of NAAT, Cluj, 2006, pp. 155-161.

[53] P.L. Clark, *The Stone-Weierstrass Approximation Theory*, www.math.uga.edu.

[54] Gh. Coman, *Two-dimensional monosplines and optimal cubature formulae,* Studia Univ. "Babeş-Bolyai", Ser. Math.-Mech. 1973, no. 1, pp. 41–53.

[55] Gh. Coman, *Multivariate approximation schemes and the approximation of linear functionals*, Mathematica, **16 (39)** (1974), pp. 229–249.

[56] Gh. Coman, *The approximation of multivariate functions*, Technical Summary Report 1254, Mathematics Research Center, University of Wisconsin, Madison, Wisconsin, 1974.

[57] Gh. Coman, *The complexity of the quadrature formulas*, Mathematica (Cluj), **23 (46)** (1981), pp. 183–192.

[58] Gh. Coman, *Homogeneous multivariate approximation formulas with applications in numerical integration*, Research Seminaries, "Babeş-Bolyai" University, Faculty of Mathematics, Preprint No. 4, 1985, pp. 46–63.

[59] Gh. Coman, *The remainder of certain Shepard type interpolation formulas,* Studia Univ. "Babeş-Bolyai", Mathematica, **32** (1987), no. 4, pp. 24–32.

[60] Gh. Coman, *Shepard-Taylor interpolation,* Itinerant Seminar on Functional Equations, Approximation and Convexity, Cluj-Napoca, 1988, pp. 5–14.

[61] Gh. Coman, *Homogeneous cubature formulas*, Studia Univ. "Babeş-Bolyai", Mathematica, **38** (1993), no. 2, pp. 91–101.

[62] Gh. Coman, *Analiză Numerică*, Ed. Libris, Cluj-Napoca, 1995.

[63] Gh. Coman, *Shepard operators of Hermite-type*, Rev. Anal. Numér. Théor. Approx., **26** (1997), no. 1–2, pp. 33–38.

[64] Gh. Coman, *Shepard operators of Birkhoff type,* Calcolo, **35** (1998), pp. 197–203.

[65] Gh. Coman, *Numerical multivariate approximation operator*, "Tiberiu Popoviciu" Itinerant Seminar of Functional Equations, Approximation and Convexity, Cluj-Napoca, 2001, May 22–26.

[66] Gh. Coman, M. Birou, *Bivariate spline-polynomial interpolation,* Studia Mathematica "Babeş-Bolyai", **48** (2003), no. 4, pp. 17–25.

[67] Gh. Coman, K. Böhmer, *Blending interpolation schemes on triangle with error bounds*, Lecture Notes in Mathematics, Springer-Verlag, Berlin, **571** (1977), pp. 14–37.

[68] Gh. Coman, T. Cătinaş, M. Birou, A. Oprişan, C. Oşan, I. Pop, I. Somogyi, I. Todea, *Interpolation operators*, Ed. "Casa Cărţii de Ştiinţă", Cluj-Napoca, 2004.

[69] Gh. Coman, I. Gânscă, *Blending approximation with applications in constructions,* Buletinul Ştiinţific al Institutului Politehnic Cluj-Napoca, **24** (1981), pp. 35–40.

[70] Gh. Coman, I. Gânscă, *Some practical application of blending approximation II,* Itinerant Seminar on Functional Equations, Approximation and Convexity, Cluj-Napoca, 1986.

[71] Gh. Coman, I. Gânscă, L. Ţâmbulea, *Some practical application of blending interpolation,* Proceedings of the Colloquium on Approximation and Optimization, Cluj-Napoca, October 25–27, 1984.

[72] Gh. Coman, I. Gânscă, L. Ţâmbulea, *Some new roof-surfaces generated by blending interpolation technique*, Studia Univ. "Babeş-Bolyai", Mathematica, **36** (1991), no. 1, pp. 119–130.

[73] Gh. Coman, I. Gânscă, L. Ţâmbulea, M*ultivariate approximation*, Seminar on Numerical and Statistical Calculus, Preprint no. 1, 1996, pp. 29–60.

[74] Gh. Coman, D. Johnson, *Complexitatea algoritmilor*, Lito. Univ. "Babeş-Bolyai", Cluj-Napoca, 1987.

[75] Gh. Coman, I. Pop, *Some interpolation schemes on triangle*, Studia Univ. "Babeş-Bolyai", Mathematica, **48** (2003), 3, pp. 57–62.

[76] Gh. Coman, I. Pop, R. Trîmbiţaş, *An adaptive cubature on triangle*, Studia Univ. "Babeş-Bolyai", Mathematica, **47** (2002), 4, pp. 27–35.

[77] Gh. Coman, I. Purdea Pop, *On the remainder term in multivariate approximation*, Studia Univ. "Babeş-Bolyai", Mathematica, **43** (1998), no. 1, pp. 7–14.

[78] Gh. Coman, I. Somogyi, *Homogeneous formulas for approximation of multiple integrales,* Studia Univ. "Babeş-Bolyai", Mathematica, **45** (2000), pp. 11–18.

[79] Gh. Coman, R. Trîmbiţaş, *Lagrange-type Shepard operators*, Studia Univ. "Babeş-Bolyai", Mathematica, **42** (1997), pp. 75–83.

[80] Gh. Coman, R. Trîmbiţaş, *Combined Shepard univariate operators,* East Jurnal on Approximations, **7** (2001), 4, pp. 471–483.

[81] Gh. Coman, R. Trîmbiţaş, *Univariate Shepard-Birkhoff interpolation,* Rev. Anal. Numér. Théor. Approx., **30** (2001), no. 1, pp. 15–24.

[82] Gh. Coman, L. Ţâmbulea, *A Shepard-Taylor approximation formula,* Studia Univ. "Babeş-Bolyai", Mathematica, **33** (1988), no. 3, pp. 65–73.

[83] Gh. Coman, L. Ţâmbulea, *On some interpolation of scattered data*, Studia Univ. "Babeş-Bolyai", **35** (1990), no. 2, pp. 90–98.

[84] Gh. Coman, L. Ţâmbulea, *Bivariate Birkhoff interpolation of scattered data*, Studia Univ. "Babeş-Bolyai", **36** (1991), no. 2, pp. 77–86.

[85] Gh. Coman, L. Ţâmbulea, I. Gânscă, *Multivariate approximation*, Seminar on Numerical and Statistical Calculus, Preprint no. 1, 1996, pp. 29–60.

[86] Ph. J. Davis, *Interpolation and Approximation,* Blaisdell Publishing Company, New York, 1963.

[87] B. Della Vecchia, G. Mastroianni, P. Vertesi,*Direct and converse theorems for Shepard rational approximation*, Numer. Funct. Anal. Optim., **17** (1996), nos. 5–6, pp. 537–561.

[88] F.J. Delvos, *D-variate boolean interpolation*, J. Approx. Theory, **34** (1982), pp. 99–44.

[89] F.J. Delvos, *Boolean methods for double integration*, Math. Comp., **55** (1990), pp. 683–692.

[90] F.J. Delvos, W. Schempp, *Boolean methods in interpolation and approximation,* Longman Scientific & Technical, England, copublished with John Wiley & Sons, New York, 1989.

[91] J.W. Demmel, *Solving the Discrete Poisson Equation using Mutigrid, CS267*, Notes for Lectures, www.cs.berkeley.edu.

[92] C.C. Douglas, *A Review of Numerous Parallel Multigrid Methods*, SIAM News, V. 25, No. 3, 2000.

[93] A.V. Efimov, *Modulus of continuity,* Enciclopedia of Mathematics, 2001.

[94] R. Farwig, *Rate of convergence of Shepard's global interpolation formula*, Math. Comp., **46** (1986), pp. 577–590.

[95] R. Franke, *Scattered data interpolation: tests of some methods,* Math. Comp., **38** (1982), pp. 181–200.

[96] R. Franke, G.M. Nielson, *Smooth interpolation of large sets of scattered data*, International Journal for Numerical Methods in Engineering, **15** (1980), pp. 1691–1704.

[97] P.O. Frederickson, *O.A. McBryan, Parallel superconvergent multigrid*, Lecture Notes in Pure and App. Math., 1988.

[98] M. Gasca, T. Sauer, *Polynomial interpolation in several variables. Multivariate polynomial interpolation,* Adv. Comput. Math., **12** (2000), no. 4, pp. 377–410.

[99] M. Gasca, T. Sauer, *On the history of multivariate polynomial interpolation*, J. Comput. Appl. Math., **122** (2000), pp. 23–35.

[100] W.B. Gearhart, M. Qian, *Peano kernels and the Euler-MacLaurin formula*, Far East J. Math. Sci., **3** (2001), no. 2, pp. 247–271.

[101] W.M. Gentelman, *Least squares computations by Givens transformations without square roots*, Journal of the IMA, 12, 1973.

[102] A.C. Gilbert, *The SOR Method*, www.maths.lancs.ac.uk.

[103] W.J. Gordon, *Distributive lattices and approximation of multivariate functions,* Proc. Symp. Approximation with Special Emphasis on Spline Functions (Madison, Wisc.), (Ed. I.J. Schoenberg), 1969, pp. 223–277.

[104] W.J. Gordon, *Blending-function methods of bivariate and multivariate interpolation and approximation*, SIAM J. Numer. Anal., **8** (1971), pp. 158–177.

[105] W. Hackbusch, *A new approach to robust multigrid solvers*, Proceeding of ICIAM'87.

[106] P.P. Korovkin, *On converge of linear positive operators in the space of continuous functions*, Dobl. Akad. Nauk SSSR 90, 1960, pp. 961-964.

[107] D.V. Ionescu, *Numerical Quadratures,* Bucharest, 1957.

[108] D.V. Ionescu, *Sur une classe de formules de cubature*, C.R. Acad. Sc. Paris, **266** (1968), pp. 1155–1158.

[109] D.V. Ionescu, *Extension de la formule de quadrature de Gauss à une classe de formules de cubature*, C.R. Acad. Sc. Paris, **269** (1969), pp. 655–657.

[110] D.V. Ionescu, *L'extension d'une formule de cubature*, Acad. Royale de Belgique, Bull. de la Classe des Science, **56** (1970), pp. 661–690.

[111] W. Light, W. Cheney, *Approximation Theory in Tensor Product Spaces*, Lecture Notes in Mathematics 1169, (Ed. A. Dold and B. Eckmann), Springer-Verlag, Berlin, 1985.

[112] L.E. Mansfield, *On the optimal approximation of linear functionals in spaces of bivariate functions*, SIAM J. Numer. Anal., **8** (1971), pp. 115–126.

[113] C.A. Micchelli, *Interpolation of scattered data: distance matrices and conditionally positive definite functions*, Constr. Approx., **2** (1986), no. 1, pp. 11–22.

[114] J.J. Modi, *Parallel Algorithm and Matrix Computation*, Clarendon Press, Oxford, 1988.

[115] G.M. Nielson, R. Franke, *A method for construction of surfaces under tension,* Rocky Mtn. J. Math., **14** (1984), no. 1, pp. 203–221.

[116] H.H. Olsen, *The Weierstrass density theorem, TMA4230,* Functional Analysis, 2005.

[117] A. Oprişan, *About convergence order of the iterative methods generated by inverse interpolation*, Seminar on Numerical and Statistical Calculus, 2004, pp. 97–109.

[118] A. Pinkus, *Density Methods and Results in Approximation Theory,* Orliez Centenary volume, Banck Center Publication, vol.64, Polish Acad. of Science, Warszawa, 2004.

[119] T. Popoviciu, *On the proof of Weierstrass theorem using interpolation polynomials,* Lucr. Ses. Gen. Ştiinţific, 2:12 (1950), pp. 1664-1667.

[120] H.J. Quinn, *Designing Efficient Algorithm for Paralell Computers*, McGraw-Hill Int., 1988.

[121] A.H. Sameh, R.P. Brent, *Solving triangular systems on a parallel computer*, SIAM Journal on Numerical Analysis, 14 (6), 1977.

[122] A. Sard, *Linear Approximation*, AMS, Providence, RI, 1963.–334.

[123] I.J. Schoenberg, *On monosplines of least deviation and best quadrature formulae*, J. SIAM Numer. Anal., Ser. B, **2** (1965) no. 1, pp. 145–170.

[124] I.J. Schoenberg, *On monosplines of least deviation and best quadrature formulae* II, J. SIAM Numer. Anal., Ser. B, **3** (1966), no. 2, pp. 321–328.

[125] L.L. Schumaker, *Fitting surfaces to scattered data*, Approximation Theory II, (Eds. G. G. Lorentz, C. K. Chui, L. L. Schumaker), Academic Press, 1976, pp. 203–268.

[126] L.L. Schumaker, *Spline functions: basic theory,* Pure and Applied Mathematics. John Wiley & Sons, Inc., New York, 1981.

[127] B. Sendov, A. Andreev, *Approximation and Interpolation Theory*, Handbook of Numerical Analysis, vol. III, ed. P.G. Ciarlet and J.L. Lions, North Holland, Amsterdam, 1994.

[128] W. Shayne, *Refinements of the Peano kernel theorem*, Numer. Funct. Anal. Optimiz., **20** (1999) nos. 1&2, pp. 147–161.

[129] D. Shepard, *A two-dimensional interpolation function for irregularly spaced points*, Proc. 23rd Nat. Conf. ACM (1968), pp. 517–523.

[130] S.A. Smolyak, *Quadrature and interpolation formulas for tensor products of certain classes of functions*, Soviet Math. Dokl., **4**, 1963, pp. 240–243.

[131] I. Somogyi, *About some cubature formulas*, Proceedings of the International Symposium on Numerical Analysis and Approximation Theory, Cluj-Napoca, May 9–11, 2002, pp. 430–436.

[132] I. Somogyi, *Almost optimal numerical methods*, Studia Univ. "Babeş-Bolyai", Mathematica, 1999, no. 1, pp. 85–93.

[133] D.D. Stancu, *A study of the polynomial interpolation of functions of several variables, with applications to the numerical differentiation and integration; methods for evaluating the remainders*, Doctoral Dissertation, 1956, University of Cluj.

[134] D.D. Stancu, *The generalization of certain interpolation formulae for the functions of many variables,* Buletinul Institutului Politehnic din Iaşi, nos. 1–2, 1957, pp. 31–38.

[135] D.D. Stancu, *On the Hermite interpolation formula and on some of its applications,* Acad. R. P. Romîne. Fil. Cluj. Stud. Cerc. Mat., **8** (1957), pp. 339–355.

[136] D.D. Stancu, *Generalization of some interpolation formulas for multivariate functions and some contributions on the Gauss formula for numerical integration*, Buletin St. Acad. Române, **9**, 1957, pp. 287–313.

[137] D.D. Stancu, *On some Taylor expansions for functions of several variables*, Rev. Roumaine Math. Pures Appl., **4** (1959), pp. 249–265 (in Russian).

[138] D.D. Stancu, *The expression of the remainder in some numerical partial differentiation formulas, Acad. R. P. Romîne Fil. Cluj Stud. Cerc. Mat., **11** (1960), pp. 371–380.*

[139] D.D. Stancu, *On the integral representation of the remainder in Taylor's formula in two variables*, Stud. Cerc. Mat. (Cluj), **13** (1962), no. 1, pp. 175–182.

[140] D.D. Stancu, *The remainder of certain linear approximation formulas in two variables,* SIAM J. Numer. Anal. Ser. B, **1** (1964), pp. 137–163.

[141] D.D. Stancu, *On Hermite's osculatory interpolation formula and on some generalizations of it*, Mathematica (Cluj), **8 (31)** (1966), pp. 373–391.

[142] D.D. Stancu, *A generalization of the Schoenberg approximating spline operator*, Studia Univ. "Babeş-Bolyai", Math., **26** (1981), no. 2, pp. 37–42.

[143] D.D. Stancu, Gh. Coman, O. Agratini, R. Trîmbiţaş, *Analiză Numerică şi Teoria Aproximării*, vol. I, Presa Universitară Clujeană, 2001.

[144] D.D. Stancu, Gh. Coman, P. Blaga, *Analiză Numerică şi Teoria Aproximării*, vol. II, Presa Universitară Clujeană, 2002.

[145] A.H. Stroud, *Optimal quadrature formulas*, Approximation in Theory and Praxis, Bibliographices Institut, Mannheim, 1979.

[146] A.H. Stroud, *Approximate calculation of multiple integrals,* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1971.

[147] J.R. Shewchuk, *An Introduction to the Conjugate Gradient Method without Agonizing Pain, School of Computer Science,* Carnegie Mellon Univ., Pittsburg, 1994.

[148] O. Shisa, B. Mond, *The degree of convergence of sequences of linear positive operators*, Aerospace Research Lab., Ohio, 1968, pp.11-96.

[149] J. Szabados, *Direct and converse approximation theorems for the Shepard operator*, Approx. Theory Appl., **7** (1991), no. 3, pp. 63–76.

[150] S. Ta'asan, *Multigrid Methods for Highly Oscillatory Problems, PhD thesis*, Weizmann Inst. of Science, Rehovot, Israel, 1984.

[151] R. Toomas, *Transformation and LU-Factorization,* www.cs.ut.ee.

[152] J.F. Traub, *Iterative Methods for the Solutions of Equations*, Prentice-Hall, Inc, Englewood Cliffs, 1964.

[153] R.T. Trîmbiţaş, *Univariate Shepard-Lagrange interpolation*, Kragujevac J. Math., **24** (2002), pp. 85–94.

[154] R. Trîmbitaş, *Numerical Analysis*, Presa Universitară Clujeană, 2006.

[155] P. Wong, *Iterative solvers for System of Linear Equations*, www.jics.utk.edu, 2002.

[156] ***www.Wikipedia.org, Weierstrass Approximation Theorem, www.literka.addr.com.