UNIUNEA EUROPEANĂ
GUVERNUL ROMÂNIEI MINISTERUL MUNCII, FAMILIEI ŞI PROTECŢIEI SOCIALE AMPOSDRU
Fondul Social European POS DRU 2007 - 2013
Instrumente Structurale 2007 - 2013
MINISTERUL EDUCAŢIEI CERCETĂRII TINERETULUI ŞI SPORTULUI
OIPOSDRU
Universitatea POLITEHNICA din Bucuresti

Contract POSDRU/86/1.2/S/62485

Universitatea Tehnică din Cluj-Napoca

Ioan Gavrea Mircea Ivan

ML. Numerical Methods

UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA

Universitatea Tehnică "Gheorghe Asachi" din Iaşi

Universitatea din Craiova

Contents

ML.1

Numerical Methods in Linear Algebra

ML.1.1 Special Types of Matrices

Let $\mathcal{M}_{m,n}(\mathbb{R})$ be the set of all $m \times n$ type matrices with real entries, where m, n are positive integers ($\mathcal{M}_n(\mathbb{R}) := \mathcal{M}_{n,n}(\mathbb{R})$).

DEFINITION ML.1.1.1

A matrix $A \in \mathcal{M}_n(\mathbb{R})$ is said to be *strictly diagonally dominant* when its entries satisfy the condition

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|$$

for each $i = 1, \ldots, n$.

THEOREM ML.1.1.2

A strictly diagonally dominant matrix is nonsingular.

Proof. Consider the linear system

$$AX = 0, \qquad A \in \mathcal{M}_n(\mathbb{R}),$$

which has a nonzero solution $X = [x_1, \ldots, x_n]^t \in \mathcal{M}_{n,1}(\mathbb{R})$. Let k be an index such that

$$|x_k| = \max_{1 \leq j \leq n} |x_j|.$$

Since

$$\sum_{j=1}^{n} a_{kj} x_j = 0,$$

we have

$$a_{kk}x_k = -\sum_{\substack{j=1\\ j\neq k}}^{n} a_{kj}x_j.$$

This implies that

$$|a_{kk}| \leq \sum_{\substack{j=1\\ j\neq k}}^{n} |a_{kj}|\frac{|x_j|}{|x_k|} \leq \sum_{\substack{j=1\\ j\neq k}}^{n} |a_{kj}|.$$

This contradicts the strict diagonal dominance of A. Consequently, the only solution to $AX = 0$ is $X = 0$, a condition equivalent to the nonsingularity of A. □

Another special class of matrices is called *positive definite.*

DEFINITION ML.1.1.3

A matrix $A \in \mathcal{M}_n(\mathbb{R})$ is said to be *positive definite* if

$$\det(X^t AX) > 0$$

for every $X \in \mathcal{M}_{n,1}(\mathbb{R})$, $X \neq 0$.

Note that, for $X = [x_1, \ldots, x_n]^t$, we have

$$\det(X^t AX) = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}x_j x_i.$$

Using the definition (ML.1.1.3) to determine whether a matrix is positive definite or not can be difficult. The next result provides some conditions that can be used to eliminate certain matrices from consideration.

THEOREM ML.1.1.4

If the matrix $A \in \mathcal{M}_n(\mathbb{R})$ is symmetric and positive definite, then:
(1) A is nonsingular;
(2) $a_{kk} > 0$, for each $k = 1, \ldots, n$;
(3) $\max\limits_{1\leq k\neq j\leq n} |a_{kj}| < \max\limits_{1\leq i\leq n} |a_{ii}|$;
(4) $(a_{ij})^2 < a_{ii}a_{jj}$, for each $i, j = 1, \ldots, n$, $i \neq j$.

Proof. (1) If $X \neq 0$ is a vector which satisfies $AX = 0$, then

$$\det(X^t AX) = 0.$$

This contradicts the assumption that A is positive definite. Consequently, $AX = 0$ has only the zero solution and A is nonsingular.

(2) For an arbitrary k, let $X = [x_1, \ldots, x_n]^t$ be defined by

$$x_j = \begin{cases} 1, & \text{when } j = k, \\ 0, & \text{when } j \neq k, \end{cases} \qquad j = 1, 2, \ldots, n.$$

Since $X \neq 0$, $a_{kk} = \det(X^t AX) > 0$.

(3) For $k \neq j$ define $X = [x_1, \ldots, x_n]^t$ by

$$x_i = \begin{cases} 0, & \text{when } i \neq j \text{ and } i \neq k, \\ -1, & \text{when } i = j, \\ 1, & \text{when } i = k. \end{cases}$$

Since $X \neq 0$, $a_{jj} + a_{kk} - a_{jk} - a_{kj} = \det(X^t AX) > 0$. But $A^t = A$, so $a_{jk} = a_{kj}$ and $2a_{kj} < a_{jj} + a_{kk}$.

Define $Z = [z_1, \ldots, z_n]^t$ where

$$z_i = \begin{cases} 0, & \text{when } i \neq j \text{ and } i \neq k, \\ 1, & \text{when } i = j \text{ or } i = k, \end{cases}$$

Then $\det(Z^t AZ) > 0$, so $-2a_{kj} < a_{kk} + a_{jj}$. We obtain

$$|a_{kj}| < \frac{a_{kk} + a_{jj}}{2} \leq \max_{1 \leq i \leq n} |a_{ii}|.$$

Hence

$$\max_{1 \leq k \neq j \leq n} |a_{kj}| < \max_{1 \leq i \leq n} |a_{ii}|.$$

(4) For $i \neq j$, define $X = [x_1, \ldots, x_n]^t$ by

$$x_k = \begin{cases} 0, & \text{when } k \neq j \text{ and } k \neq i, \\ \alpha, & \text{when } k = i, \\ 1, & \text{when } k = j. \end{cases}$$

where α represents an arbitrary real number. Since $X \neq 0$,

$$0 < \det(X^t AX) = a_{ii}\alpha^2 + 2a_{ij}\alpha + a_{jj}.$$

As a quadratic polynomial in α with no real roots, the discriminant must be negative. Thus

$$4(a_{ij}^2 - a_{ii}a_{jj}) < 0$$

and

$$a_{ij}^2 < a_{ii}a_{jj}.$$

□

ML.1.2 Norms of Vectors and Matrices

DEFINITION ML.1.2.1

A *vector norm* is a function $\|\cdot\| : \mathcal{M}_{n,1}(\mathbb{R}) \to \mathbb{R}$ satisfying the following conditions:

(1) $\|X\| \geq 0$ for all $X \in \mathcal{M}_{n,1}(\mathbb{R})$,

(2) $\|X\| = 0$ if and only if $X = 0$,

(3) $\|\alpha X\| = |\alpha|\|X\|$ for all $\alpha \in \mathbb{R}$ and $X \in \mathcal{M}_{n,1}(\mathbb{R})$,

(4) $\|X + Y\| \leq \|X\| + \|Y\|$ for all $X, Y \in \mathcal{M}_{n,1}(\mathbb{R})$.

The most common vector norms for n-dimensional column vectors with real number coefficients, $X = [x_1, \ldots, x_n]^t \in \mathcal{M}_{n,1}(\mathbb{R})$, are defined by:

$$\begin{aligned} \|X\|_1 &= \sum_{i=1}^{n} |x_i|, \\ \|X\|_2 &= \sqrt{\sum_{i=1}^{n} x_i^2}, \\ \|X\|_\infty &= \max_{1 \le i \le n} |x_i|, \end{aligned}$$

The norm of a vector gives a measure for the distance between an arbitrary vector and the zero vector. The *distance* between two vectors can be defined as the norm of the difference of the vectors. The concept of distance in $\mathcal{M}_{n,1}(\mathbb{R})$ is also used to define the limit of a sequence of vectors.

DEFINITION ML.1.2.2

A sequence $(X^{(k)})_{k=1}^{\infty}$ of vectors in $\mathcal{M}_{n,1}(\mathbb{R})$ is said to be *convergent* to a vector $X \in \mathcal{M}_{n,1}(\mathbb{R})$, with respect to the norm $\|\cdot\|$, if

$$\lim_{k \to \infty} \|X^{(k)} - X\| = 0.$$

The notion of vector norm will be extended for matrices.

DEFINITION ML.1.2.3

A *matrix norm* on the set $\mathcal{M}_n(\mathbb{R})$ is a function $\|\cdot\| : \mathcal{M}_n(\mathbb{R}) \to \mathbb{R}$ satisfying the conditions:

(1) $\|A\| \ge 0$,

(2) $\|A\| = 0$ **if and only if** $A = 0$,

(3) $\|\alpha A\| = |\alpha| \|A\|$,

(4) $\|A + B\| \le \|A\| + \|B\|$,

(5) $\|AB\| \le \|A\| \cdot \|B\|$,

for all matrices $A, B \in \mathcal{M}_n(\mathbb{R})$ and any real number α.

It is not difficult to show that if $\|\cdot\|$ is a vector norm on $\mathcal{M}_{n,1}(\mathbb{R})$, then

$$\|A\| := \max_{\|X\|=1} \|AX\|$$

is a matrix norm called the *natural norm* or the *induced matrix norm* associated with the vector norm. In this text, all matrix norms will be assumed to be natural matrix norms unless specified otherwise.

The $\|\cdot\|_\infty$ norm of a matrix has an interesting representation with respect to the entries of the matrix.

THEOREM ML.1.2.4

If $A = [a_{ij}] \in \mathcal{M}_n(\mathbb{R})$, then

$$\|A\|_\infty = \max_{1\le i\le n} \sum_{j=1}^{n} |a_{ij}|.$$

Proof. Let $X \in \mathcal{M}_{n,1}(\mathbb{R})$ be an arbitrary column vector with

$$1 = \|X\|_\infty = \max_{1\le i\le n} |x_i|.$$

We have:

$$\begin{aligned} \|AX\|_\infty &= \max_{1\le i\le n} |(AX)_i| \\ &= \max_{1\le i\le n} \left|\sum_{j=1}^{n} a_{ij}x_j\right| \le \max_{1\le i\le n} \sum_{j=1}^{n} |a_{ij}| \cdot \max_{1\le j\le n} |x_j| \\ &= \max_{1\le i\le n} \sum_{j=1}^{n} |a_{ij}| \cdot 1. \end{aligned}$$

Consequently,

$$\|A\|_\infty = \max_{\|X\|=1} \|AX\|_\infty \le \max_{1\le i\le n} \sum_{j=1}^{n} |a_{ij}| \qquad (\circledast)$$

However, if p is an integer such that

$$\sum_{j=1}^{n} |a_{pj}| = \max_{1\le i\le n} \sum_{j=1}^{n} |a_{ij}|$$

and X is chosen with

$$x_j = \begin{cases} 1, & \text{when } a_{pj} \ge 0, \\ -1, & \text{when } a_{pj} < 0, \end{cases}$$

then $\|X\|_\infty = 1$ and $a_{pj}x_j = |a_{pj}|$ for $j = 1, \ldots, n$. So

$$\begin{aligned} \|AX\|_\infty &= \max_{1\le i\le n} \left|\sum_{j=1}^{n} a_{ij}x_j\right| \\ &\ge \left|\sum_{j=1}^{n} a_{pj}x_j\right| = \sum_{j=1}^{n} |a_{pj}| = \max_{1\le i\le n} \sum_{j=1}^{n} |a_{ij}|, \end{aligned}$$

which, together with inequality $(\circledast)$, gives

$$\|A\|_\infty = \max_{1\le i\le n} \sum_{j=1}^{n} |a_{ij}|.$$

□

Similarly, we can prove that

$$\|A\|_1 = \max_{1\leq j\leq n} \sum_{i=1}^{n} |a_{ij}|.$$

An alternative method for finding $\|A\|_2$ will be presented in the next section (see Theorem (ML.1.11.5)).

DEFINITION ML.1.2.5

The *Frobenius norm* (which is not a natural norm) is defined, for a matrix $A = [a_{ij}] \in \mathcal{M}_n(\mathbb{R})$, by

$$\|A\|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} |a_{ij}|^2}.$$

One can easily prove that, for any matrix A,

$$\|A\|_2 \leq \|A\|_F \leq \sqrt{n}\,\|A\|_2.$$

Another matrix norm can be defined by

$$\|A\| = \sum_{i=1}^{n}\sum_{j=1}^{n} |a_{ij}|.$$

Note that the function defined by

$$f(A) = \max_{1\leq i,j\leq n} |a_{ij}|$$

is not a norm.

EXAMPLE ML.1.2.6 MATHEMATICA

```
(*NORMS*)
n = 4;

a = Table [i^2 - j, {i, 1, n}, {j, 1, n}] ;
na = Dimensions[a][[1]];
SequenceForm["A= ", MatrixForm[a]]
```

$$\mathrm{A}= \begin{pmatrix} 0 & -1 & -2 & -3 \\ 3 & 2 & 1 & 0 \\ 8 & 7 & 6 & 5 \\ 15 & 14 & 13 & 12 \end{pmatrix}$$

```
NormInfinity =
Max [Table [Sum_{j=1}^{na} Abs[a[[i, j]]], {i, 1, na}]] ;
SequenceForm ["||A||_inf= ", NormInfinity]
```

$||\mathrm{A}||_{\mathrm{inf}}= 54$

```
Norm1 =
Max [Table [Sum_{i=1}^{na} Abs[a[[i, j]]], {j, 1, na}]] ;
SequenceForm ["||A||_1= ", Norm1]
```

$||\mathrm{A}||_1= 26$

EXAMPLE ML.1.2.7 MATHEMATICA

```
(*Norm_2*)
A = Table[Min[i, j], {i, -1, 1}, {j, -1, 1}];
norm2[A_]:=Sqrt[Max[Abs[ComplexExpand[Eigenvalues[Transpose[A].A]]]]]//
FullSimplify
SequenceForm["A = ", MatrixForm[A]]
SequenceForm ["A^tA = ", MatrixForm[Transpose[A].A]]
SequenceForm ["norm_2[A] = ", norm2[A], " = ", N[norm2[A]], "..."]
```

$$\mathrm{A} = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 0 & 0 \\ -1 & 0 & 1 \end{pmatrix}$$

$$A^t\mathrm{A} = \begin{pmatrix} 3 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 2 \end{pmatrix}$$

$$\mathrm{norm}_2[\mathrm{A}] = \sqrt{2 + \mathrm{Cos}\left[\tfrac{\pi}{9}\right] + \sqrt{3}\mathrm{Sin}\left[\tfrac{\pi}{9}\right]} = 1.87939...$$

ML.1.3 Error Estimation

Consider the linear system

$$AX = B,$$

where $A \in \mathcal{M}_n(\mathbb{R})$ and $B \in \mathcal{M}_{n,1}(\mathbb{R})$. It seems intuitively reasonable that if $\overline{X}$ is an approximation to the solution X of the above-mentioned system and the *residual vector* $R = B - A\overline{X}$ has the property that $\|R\|$ is small, then $\|X - \overline{X}\|$ would be small as well. This is often the case, but certain systems, which occur frequently in practical problems, fail to have this property.

EXAMPLE ML.1.3.1

The linear system $AX = B$ given by

$$\begin{bmatrix} 1 & 2 \\ 1.0001 & 2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 3.0001 \end{bmatrix}$$

has the unique solution $X = [1, 1]^t$. The poor approximation $\overline{X} = [3, 0]^t$ has the residual vector

$$R = B - A\overline{X}$$
$$= \begin{bmatrix} 3 \\ 3.0001 \end{bmatrix} - \begin{bmatrix} 1 & 2 \\ 1.0001 & 2 \end{bmatrix} \begin{bmatrix} 3 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -0.0002 \end{bmatrix},$$

so $\|R\|_\infty = 0.0002$. Although the norm of the residual vector is small, the approximation $\overline{X} = [3, 0]^t$ is obviously quite poor. In fact, $\|X - \overline{X}\|_\infty = 2$.

□

In a general situation we can obtain information of when problems might occur by considering the norm of the matrix A and its inverse.

THEOREM ML.1.3.2

Let $\overline{X}$ be an approximative solution of $AX = B$, where A is a nonsingular matrix, R is the residual vector for $\overline{X}$, and $B \neq 0$. Then for any natural norm,

$$\|X - \overline{X}\| \leq \|R\| \cdot \|A^{-1}\|,$$

and

$$\frac{\|X - \overline{X}\|}{\|X\|} \leq \|A\| \cdot \|A^{-1}\| \cdot \frac{\|R\|}{\|B\|}.$$

Proof. Since A is nonsingular and

$$R = B - A\overline{X} = A(X - \overline{X}),$$

we obtain

$$\|X - \overline{X}\| = \|A^{-1}R\| \leq \|A^{-1}\| \cdot \|R\|.$$

Moreover, since $B = AX$, we have

$$\|B\| \leq \|A\| \cdot \|X\|,$$

and

$$\frac{\|X - \overline{X}\|}{\|X\|} \leq \frac{\|A\| \cdot \|A^{-1}\|}{\|B\|} \cdot \|R\|.$$

□

DEFINITION ML.1.3.3

The *condition number* $K(A)$ of a nonsingular matrix A relative to a natural norm $\|\cdot\|$ is defined by

$$K(A) = \|A\| \cdot \|A^{-1}\|.$$

With this notation, the inequalities in Theorem (ML.1.3.2) become

$$\|X - \overline{X}\| \leq K(A) \cdot \frac{\|R\|}{\|A\|}$$

and

$$\frac{\|X - \overline{X}\|}{\|X\|} \leq K(A) \cdot \frac{\|R\|}{\|B\|}.$$

For any nonsingular matrix A and the natural norm $\|\cdot\|$, we have

$$1 = \|I\| = \|A \cdot A^{-1}\| \leq \|A\| \cdot \|A^{-1}\| = K(A),$$

so

$$K(A) \geq 1.$$

The matrix A is said to be *well-conditioned* if $K(A)$ is close to one and *ill-conditioned* when $K(A)$ is significantly greater than one. The matrix of the system considered in Example (ML.1.3.1) is

$$A = \begin{bmatrix} 1 & 2 \\ 1.0001 & 2 \end{bmatrix}$$

which has $\|A\|_\infty = 3.0001$. But

$$A^{-1} = \begin{bmatrix} -10000 & 10000 \\ 5000.5 & -5000 \end{bmatrix}$$

so $\|A^{-1}\|_\infty = 20000$. The condition number for the infinity norm is $K(A) = 60002$. Its size certainly keeps us from making hasty accuracy decisions based on the residual vector of an approximation.

ML.1.4 Matrix Equations. Pivoting Elimination

Matrix equations are associated with many problems arising in engineering and science, as well as with applications of mathematics to social sciences. For solving a matrix equation, the *partial pivoting elimination* is about as efficient as any other method.

Pivoting is a process of interchanging rows (partial pivoting) or rows and columns (full pivoting), so as to put a particularly desirable element in the diagonal position from which the pivot is about to be selected.

Let us recall the principal row elementary operations used to transform a matrix equation to a more convenient one with the same solution:

1. Multiply the row i by a nonzero constant λ. This operation is denoted by
$$r_i \leftarrow \lambda\, r_i.$$
2. Add the row j multiplied by a constant λ to the row i. This operation is denoted by
$$r_i \leftarrow r_i + \lambda\, r_j.$$
3. Interchange rows i and j. This operation is denoted by
$$r_i \rightleftharpoons r_j.$$

EXAMPLE ML.1.4.1

Consider the matrices:

$$\boldsymbol{A} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{1} \\ \mathbf{2} & \mathbf{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{1} \end{bmatrix}, \qquad \boldsymbol{B} = \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \\ \mathbf{0} \end{bmatrix}.$$

Let us use the partial pivoting method to solving the matrix equation

$$\boldsymbol{AX} = \boldsymbol{B}$$

for the unknown matrix $\boldsymbol{X}$.

By augmenting A with the column matrix B we obtain the *augmented matrix*

$$C_0 = [A; B] = \begin{bmatrix} 1 & 0 & 1 & ; & 1 \\ 2 & -1 & 0 & ; & 1 \\ 0 & 1 & 1 & ; & 0 \end{bmatrix}.$$

Performing row operations, step by step, on the matrix C_0, we will obtain the matrices C_1, C_2, C_3 :

Step 1: From C_0, using the operations:

$$\begin{array}{lcl} r_1 & \rightleftharpoons & r_2 \\ r_1 & \leftarrow & \frac{1}{2}\, r_1 \\ r_2 & \leftarrow & r_2 - r_1 \end{array},$$

we obtain

$$C_1 = \left[\begin{array}{ccc;c} 1 & -\frac{1}{2} & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & 1 & \frac{1}{2} \\ 0 & 1 & 1 & 0 \end{array}\right].$$

Step 2: From C_1, using the operations:

$$\begin{array}{rcl} r_2 & \rightleftharpoons & r_3 \\ r_3 & \leftarrow & r_3 - \frac{1}{2} r_2 \\ r_1 & \leftarrow & r_1 + \frac{1}{2} r_2 \end{array},$$

we get the matrix

$$C_2 = \left[\begin{array}{ccc;c} 1 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 1 & 1 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{array}\right].$$

Step 3: From C_2, using the operations:

$$\begin{array}{rcl} r_1 & \leftarrow & r_1 - r_3 \\ r_3 & \leftarrow & 2\, r_3 \\ r_2 & \leftarrow & r_2 - r_3 \end{array},$$

we obtain the matrix

$$C_3 = \left[\begin{array}{ccc;c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 1 \end{array}\right].$$

The last column in the matrix C_3 is the unknown X, i.e.,

$$X = A^{-1}B = \left[\begin{array}{c} 0 \\ -1 \\ 1 \end{array}\right].$$

The *partial pivoting elimination method* can be described as follows:

Let $A \in \mathcal{M}_n(\mathbb{C})$, and $B \in \mathcal{M}_{n,p}(\mathbb{C})$. The matrix equation

$$AX = B,$$

will be solved for the unknown $X \in \mathcal{M}_{n,p}(\mathbb{C})$. Consider the augmented matrix

$$C^{(0)} = [A; B] \in \mathcal{M}_{n,n+p}(\mathbb{C}).$$

A set of elementary row operations will be performed on the matrix $C^{(0)}$ until the matrix A is reduced to the identity matrix. When this is done, the solution $X = A^{-1}B$ replaces the matrix B. We can arrange these transformations into n steps. In the kth step, a new matrix $C^{(k)}$ is obtained from the existing matrix $C^{(k-1)}$, $k = 1, 2, \ldots, n$. At the beginning of the step k we compare the moduli of the elements $c_{ik}^{(k-1)}$, $i = k, \ldots, n,$. Among these, the element having the

largest modulus, is called *pivot element.* Then, the row containing the pivot element and the kth row are interchanged. If, after last interchange, the modulus of the pivot element $c_{kk}^{(k-1)}$ is less than a given value ε, then the matrix A is considered to be singular and the procedure stops.

Now, the elements of the matrix $C^{(k)}$ are calculated and given by:

$$\begin{cases} c_{kj}^{(k)} = c_{kj}^{(k-1)} / c_{kk}^{(k-1)} \quad (j = k+1, \ldots, n+p) \\ \\ c_{ij}^{(k)} = c_{ij}^{(k-1)} - c_{ik}^{(k-1)} c_{kj}^{(k)} \\ (i = 1, \ldots, n; \quad i \neq k; \quad j = k+1, \ldots, n+p;) \end{cases} \tag{1.4.1}$$

$k = 1, \ldots, n.$

The last p columns of the matrix $C^{(n)}$ is the solution $X = A^{-1}B$.

At the same time, the determinant of the matrix A can be calculated by the formula

$$\det(A) = (-1)^{\sigma} c_{11}^{(0)} c_{22}^{(1)} \cdots c_{nn}^{(n-1)}, \tag{1.4.2}$$

where, $c_{kk}^{(k-1)}$ is the pivot element in the k th step, and σ is the number of row interchanges performed.

Note that, after last interchange, it is not necessary to perform transforms to the columns $1, \ldots, k$.

EXAMPLE ML.1.4.2 MATHEMATICA

```
(* Pivoting elimination *)
c = {{1, 0, 1, 1}, {2, −1, 0, 1}, {0, 1, 1, 0}};
Print[“c= ”, MatrixForm[c]];
det = 1; k = 1; n = 3; p = 1;
While[k ≤ n, If[k ≠ n,
imx = k; cmx = Abs[c[[k, k]]];
For[i = k + 1, i ≤ n, i++,
If[cmx < Abs[c[[i, k]]], imx = i;
cmx = Abs[c[[i, k]]]]];
If[imx ≠ k, For[j = n + p, j ≥ 1, j−,
t = c[[imx, j]];
c[[imx, j]] = c[[k, j]]; c[[k, j]] = t]]; det = − det];
If[Abs[c[[k, k]]] < 0.1, k = n + 1, det = det c[[k, k]];
For [j = n + p, j ≥ 1, j−, c[[k, j]] = c[[k,j]]/c[[k,k]]]];
For[i = 1, i ≤ n, i++,
If[i ≠ k, For[j = n + p, j ≥ 1, j−,
c[[i, j]] = c[[i, j]] − c[[i, k]]c[[k, j]]]]; ];
Pause[2]; Print[“Step ”, k];
Pause[2]; Print[MatrixForm[c]]; k++];
If[k==n + 2, Singular, Print[“det=”, det]]
```

$$\text{c=}\begin{pmatrix} 1 & 0 & 1 & 1 \\ 2 & -1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

Step 1

$$\begin{pmatrix} 1 & -\frac{1}{2} & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & 1 & \frac{1}{2} \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

Step 2

$$\begin{pmatrix} 1 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 1 & 1 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

Step 3

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

det=1

REMARK ML.1.4.3

• If $p = n$ and B is the identity matrix, then the solution of the equation $AX = B$ is A^{-1}.
• If $p = 1$ we obtain the *Gauss-Jordan method* for solving linear system of equations.
• If A is a strictly diagonally dominant matrix, the Gaussian elimination can be performed on any linear system $AX = B$ without row interchange, and the computations are stable with respect to the growth of round-off errors [BF93, p 372].

ML.1.5 Improved Solutions of Matrix Equations

Obviously it is not easy to obtain greater precision for the solution of a matrix equation than the precision of the computer's floating-point word. Unfortunately, for large sets of linear equations, it is not always easy to obtain precision equal to, or even comparable to the computer's limits.

In the direct methods of solution, roundoff errors are accumulated and they magnify to the extend when the matrix is close to singular.

Suppose that the matrix X is the exact solution of the equation

$$AX = B.$$

We don't, however, know X. We only know some slightly wrong solution say $\overline{X}$. Substituting this into $AX = B$ we obtain

$$\overline{B} = A\overline{X}.$$

In order to find a correction matrix $E(X)$ we solve the equation

$$A \cdot E(X) = B - \overline{B} = AX - A\overline{X},$$

where the right-hand side $B - \overline{B}$ is known. We obtain a slightly wrong correction $\overline{E(X)}$. So,

$$\overline{X} + \overline{E(X)}$$

is an improved solution. An extra benefit occurs if we repeat the previous steps.

ML.1.6 Partitioning Methods for Matrix Inversion

Let $A \in \mathcal{M}_n(\mathbb{R})$ be a nonsingular matrix. Consider the partition

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

where $A_{11} \in \mathcal{M}_m(\mathbb{R})$, $A_{12} \in \mathcal{M}_{m,p}(\mathbb{R})$, $A_{21} \in \mathcal{M}_{p,m}(\mathbb{R})$, and $A_{22} \in \mathcal{M}_p(\mathbb{R})$, such that $m+p = n$. In order to compute the inverse of the matrix A, we shall try to find the matrix A^{-1} into the

form

$$A^{-1} = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}.$$

From

$$A \cdot A^{-1} = I_n = \begin{bmatrix} I_m & 0_{m,p} \\ 0_{p,m} & I_p \end{bmatrix},$$

we deduce

$$\begin{cases} A_{11}B_{11} + A_{12}B_{21} &= I_m \\ A_{21}B_{11} + A_{22}B_{21} &= 0_{p,m} \\ A_{21}B_{12} + A_{22}B_{22} &= I_p \end{cases}$$

In what follows, we shall frame the matrices at the moment when they can be effectively calculated. We have:

$$B_{21} = -A_{22}^{-1}A_{21}B_{11},$$
$$(A_{11} - A_{12}A_{22}^{-1}A_{21})B_{11} = I_m,$$

hence

$$\boxed{B_{11}} = (A_{11} - A_{12}A_{22}^{-1}A_{21})^{-1},$$

consequently

$$\boxed{B_{21}} = -A_{22}^{-1}A_{21}B_{11}.$$

From $A^{-1}A = I_n$ we deduce:

$$B_{11}A_{12} + B_{12}A_{22} = 0_{m,p},$$

hence

$$\boxed{B_{12}} = -B_{11}A_{12}A_{22}^{-1},$$

and so

$$\boxed{B_{22}} = A_{22}^{-1} - A_{22}^{-1}A_{21}B_{12}.$$

By choosing $p = 1$, we obtain the following iterative technique for matrix inversion. Let

$$A_1 = [a_{11}], \qquad A_1^{-1} = \left[\frac{1}{a_{11}}\right]$$

and let $A_k \in \mathcal{M}_k(\mathbb{R})$ be the matrix obtained by cancelling the last $n-k$ rows and columns of the matrix A. We have:

$$A_k = \begin{bmatrix} A_{k-1} & u_k \\ v_k & a_{kk} \end{bmatrix}, \qquad A_k^{-1} = \begin{bmatrix} B_{k-1} & x_k \\ y_k & \beta_k \end{bmatrix},$$

$$v_k = [a_{k1}, \dots a_{k,k-1}], \qquad u_k = \begin{bmatrix} a_{1k} \\ \vdots \\ a_{k-1,k} \end{bmatrix},$$

where A_{k-1}^{-1}, v_k, u_k, are known. We deduce:

$$\begin{cases} A_{k-1}B_{k-1} + u_k \cdot y_k &= I_{k-1} \\ A_{k-1}x_k + u_k \cdot \beta_k &= 0_{k-1,1} \\ v_k \cdot x_k + a_{kk}\beta_k &= I_1 \end{cases}$$

hence

$$x_k = -A_{k-1}^{-1} u_k \beta_k,$$

$$(-v_k A_{k-1}^{-1} u_k + a_{kk})\beta_k = I_1,$$

$$\boxed{\beta_k} = \left(-v_k \cdot A_{k-1}^{-1} \cdot u_k + a_{kk}\right)^{-1},$$

$$\boxed{x_k} = -\beta_k A_{k-1}^{-1} u_k.$$

Using the relation

$$y_k A_{k-1} + \beta_k v_k = 0_{1,k-1},$$

we obtain

$$\boxed{y_k} = -\beta_k v_k A_{k-1}^{-1},$$

$$\boxed{B_{k-1}} = A_{k-1}^{-1} - A_{k-1}^{-1} u_k y_k = A_{k-1}^{-1} + x_k \cdot y_k \, {\beta_k}^{-1}.$$

In summa, starting from the matrices:

$$A_k = \begin{bmatrix} A_{k-1} & u_k \\ v_k & a_{kk} \end{bmatrix}, \qquad A_{k-1}^{-1}$$

we obtain

$$A_k^{-1} = \begin{bmatrix} B_{k-1} & x_k \\ y_k & \beta_k \end{bmatrix},$$

where:

$$\begin{cases} \beta_k &= (-v_k A_{k-1}^{-1} u_k + a_{kk})^{-1}, \\ x_k &= -\beta_k A_{k-1}^{-1} u_k, \\ y_k &= -\beta_k v_k A_{k-1}^{-1}, \\ B_{k-1} &= A_{k-1}^{-1} + x_k y_k \beta_k^{-1} \end{cases}$$

Finally, we obtain

$$A_n^{-1} = A^{-1}.$$

EXAMPLE ML.1.6.1 MATHEMATICA

```
(* Inverse by partitioning *)
CheckAbort[A = Table[Min[i, j], {i, 3}, {j, 3}];
If[A[[1, 1]]==0, Abort[]];
inva = {{1/A[[1,1]]}};
k = 2;
While[k ≤ 3, u = Table[{A[[i, k]]}, {i, k − 1}];
v = {Table[A[[k, j]], {j, k − 1}]};
d = −v.inva.u + A[[k, k]];
If[d[[1, 1]]==0, Abort[]];
β = 1/d; x = −β[[1, 1]] inva.u;
y = −bb[[1, 1]]v.inva; B = inva + x.y/β[[1,1]];
inva = Transpose[Join[Transpose[Join[B, y]],
Transpose[Join[x, β]]]]; k++];
SequenceForm[" A = ", MatrixForm[A],
" inva = ", MatrixForm[inva]],
Print["Method fails."]]
```

$$\text{A} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 2 & 3 \end{pmatrix} \quad \text{inva} = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix}$$

```
(************************************************************)
Inverse[A]//MatrixForm
```

$$\begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix}$$

ML.1.7 LU Factorization

Suppose that the matrix $A \in \mathcal{M}_n(\mathbb{R})$ can be written as the product of two matrices,

$$A = L \cdot U,$$

where L is *lower triangular* (has zero entries above the leading diagonal),

$$L = \begin{bmatrix} * & 0 & 0 & 0 \\ * & * & 0 & 0 \\ * & * & * & 0 \\ * & * & * & * \end{bmatrix}$$

and U is *upper triangular* (has zero entries below the leading diagonal),

$$U = \begin{bmatrix} * & * & * & * \\ 0 & * & * & * \\ 0 & 0 & * & * \\ 0 & 0 & 0 & * \end{bmatrix}.$$

The matrix A has an LU factorization or LU decomposition.

The LU factorization of a matrix A can be used to solve the linear system of equations

$$AX = B.$$

Using the LU factorization we can solve the system by using a two-step process. We write the system into the form

$$L(UX) = B.$$

First solve the system

$$LY = B$$

for Y. Since L is triangular, determining Y from this equation requires only $O(n^2)$ operations. Next, the triangular system

$$UX = Y$$

requires an additional $O(n^2)$ operations to determine the solution X. It is to be noted that only $O(n^2)$ number of operations is required to solve the system $AX = B$ compared with $O(n^3)$ needed by the Gaussian elimination. Determining the specific matrices L and U requires $O(n^3)$ operations, but, once the factorization is determined, any system involving the matrix A can be solved in the simplified manner.

EXAMPLE ML.1.7.1

Consider the system

$$\begin{cases} x_1 + x_2 = 4 \\ 2x_1 + x_2 - x_3 = 1 \\ 3x_1 - x_2 = 0 \end{cases}.$$

One can easily verify the following LU factorization:

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 2 & 1 & -1 \\ 3 & -1 & 0 \end{bmatrix} = L \cdot U = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 4 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 0 \\ 0 & -1 & -1 \\ 0 & 0 & 4 \end{bmatrix}.$$

Solving the system

$$LY = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 4 & 1 \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 4 \\ 1 \\ 0 \end{bmatrix} = B,$$

for Y, we find

$$Y = \begin{bmatrix} 4 \\ -7 \\ 16 \end{bmatrix}.$$

Next, solving the system

$$UX = \begin{bmatrix} 1 & 1 & 0 \\ 0 & -1 & -1 \\ 0 & 0 & 4 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 4 \\ -7 \\ 16 \end{bmatrix} = Y,$$

for X, we find

$$X = \begin{bmatrix} 1 \\ 3 \\ 4 \end{bmatrix}.$$

Note that there exist matrices which have no LU factorization, e.g.,

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Now, a method for performing matrix LU factorization is presented.

THEOREM ML.1.7.2

Let $A \in \mathcal{M}_n(\mathbb{R})$ and $A_k \in \mathcal{M}_k(\mathbb{R})$ be obtained by cancelling the last $n-k$ rows and columns of the matrix A $(k = 1, \ldots, n-1)$. If A_k are nonsingular matrices $(k = 1, \ldots, n-1)$, then there exists a lower triangular matrix L whose entries of the leading diagonal are 1, and an upper triangular matrix U such that

$$A = LU;$$

furthermore, the LU factorization is unique.

Proof. We shall use the mathematical induction method. Firstly, find

$$L_2 = \begin{bmatrix} 1 & 0 \\ l_{21} & 1 \end{bmatrix}, \qquad U_2 = \begin{bmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{bmatrix},$$

such that

$$A_2 = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ l_{21} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{bmatrix}.$$

We obtain:

$$\begin{aligned} \boxed{u_{11}} &= a_{11} \neq 0, \\ \boxed{u_{12}} &= a_{12}, \\ l_{21}u_{11} &= a_{21}, \\ \boxed{l_{21}} &= \frac{a_{21}}{a_{11}}, \\ l_{21}u_{12} + u_{22} &= a_{22}, \\ \boxed{u_{22}} &= a_{22} - a_{21}a_{12}/a_{11} = \frac{\det(A_2)}{a_{11}}. \end{aligned}$$

The factorization is unique. Next, suppose that the matrix A_{k-1} has an LU factorization

$$A_{k-1} = L_{k-1}U_{k-1}.$$

Since matrix A_{k-1} is nonsingular the matrices L_{k-1} and U_{k-1} are nonsingular. Try to find the matrix A_k into the form

$$A_k = L_kU_k,$$

$$\begin{bmatrix} A_{k-1} & b_k \\ c_k & a_{kk} \end{bmatrix}$$

$$= \begin{bmatrix} P_k & 0 \\ l_k & 1 \end{bmatrix} \begin{bmatrix} Q_k & u_k \\ 0 & u_{kk} \end{bmatrix} = \begin{bmatrix} P_kQ_k & P_ku_k \\ l_kQ_k & l_ku_k + u_{kk} \end{bmatrix}.$$

From $A_{k-1} = P_k Q_k$, and using the uniqueness property of the factorization, we obtain:

$$\begin{aligned}
\boxed{P_k} &= L_{k-1}, \\
\boxed{Q_k} &= U_{k-1}, \\
L_{k-1}u_k &= b_k, \\
\boxed{u_k} &= L_{k-1}^{-1} b_k, \\
l_k U_{k-1} &= c_k \\
\boxed{l_k} &= c_k U_{k-1}^{-1} \\
l_k u_k + u_{kk} &= a_{kk}, \\
\boxed{u_{kk}} &= a_{kk} - l_k u_k.
\end{aligned}$$

For $k = n$ the existence and the uniqueness of the LU factorization are proved. Following the proof, we obtain a recurrent method for finding the matrices L and U.

□

EXAMPLE ML.1.7.3 MATHEMATICA

```
(* LUDecomposition *)
A:= {{a11, a12}, {a21, a22}};
n = Dimensions[A][[1]];
SequenceForm["A =", MatrixForm[A]]
```

$$\text{A} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$$

```
Information["LUDecomposition"]
```

LUDecomposition[m] generates a representation of the LU decomposition of a matrix m.More...
Attributes[LUDecomposition] = {Protected}

Options[LUDecomposition] = {Modulus → 0}

```
MatrixForm[LUDecomposition[A][[1]]//Factor]
```

$$\begin{pmatrix} a_{11} & a_{12} \\ \frac{a_{21}}{a_{11}} & \frac{-a_{12}a_{21}+a_{11}a_{22}}{a_{11}} \end{pmatrix}$$

```
L = Table[If[i > j, LUDecomposition[A][[1]][[i]][[j]],
 If[i == j, 1, 0]], {i, n}, {j, n}]//Factor;
U = Table[If[i ≤ j, LUDecomposition[A][[1]][[i]][[j]],
 0], {i, n}, {j, n}]//Factor;
SequenceForm["L =", MatrixForm[L], " U = ", MatrixForm[U]]
```

$$\text{L} = \begin{pmatrix} 1 & 0 \\ \frac{a_{21}}{a_{11}} & 1 \end{pmatrix} \quad \text{U} = \begin{pmatrix} a_{11} & a_{12} \\ 0 & \frac{-a_{12}a_{21}+a_{11}a_{22}}{a_{11}} \end{pmatrix}$$

```
(********** * TEST **********)
MatrixForm[L.U]//Simplify
```

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$$

REMARK ML.1.7.4

The determinant of an LU decomposed matrix is the product of the diagonal entries of U,

$$\det(A) = \det(U) = \prod_{i=1}^{n} u_{ii}.$$

There are several factorization methods:

- **Doolittle's** method, which requires that $l_{ii} = 1, \quad i = 1, \ldots, n$;
- **Crout's** method, which requires the diagonal elements of U to be one;
- **Choleski's** method, which requires that $l_{ii} = u_{ii}, \quad i = 1, \ldots, n$.

ML.1.8 Doolittle's Factorization

We present an algorithm which produces a Doolittle type factorization. Consider the matrices:

$$L = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 & 0 \\ l_{21} & 1 & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \ldots & l_{n,n-1} & 1 \end{bmatrix},$$

$$U = \begin{bmatrix} u_{11} & u_{12} & u_{13} & \ldots & u_{1,n-1} & u_{1,n} \\ 0 & u_{22} & u_{23} & \ldots & u_{2,n-1} & u_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 0 & u_{nn} \end{bmatrix},$$

$$A = \begin{bmatrix} a_{11} & \ldots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \ldots & a_{nn} \end{bmatrix},$$

such that

$$L \cdot U = A.$$

This gives the following set of equations:

$$\sum_{k=1}^{\min(i,j)} l_{ik} u_{kj} = a_{ij}, \quad i, j = 1, \ldots, n,$$

which can be solved by arranging them in a certain order. The order is as follows:
$u_{11} := a_{11}$
For $j = 2, \ldots, n$ set
$\smile\smile\smile\smile \quad u_{1j} = a_{1j}, \quad l_{j1} = a_{j1}/u_{11}$
For $i = 2, \ldots, n-1$

$$u_{ii} = a_{ii} - \sum_{k=1}^{i-1} l_{ik}u_{ki}$$

For $j = i+1, \ldots, n$

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik}u_{kj}$$

$$l_{ji} = \left(a_{ji} - \sum_{k=1}^{i-1} l_{jk}u_{ki} \right) / u_{ii}$$

$$u_{nn} = a_{nn} - \sum_{k=1}^{n-1} l_{nk}u_{kn}$$

It is clear that each a_{ij} is used only once and never again. This means that the corresponding l_{ij} or u_{ij} can be stored in the location that a_{ij} used to occupy; the decomposition is "in place" (the diagonal unit elements l_{ii} are not stored at all). The following example presents the order of finding l_{ij} and u_{ij} for $n = 4$. The calculations are performed in the order shown in brackets.

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} {}^{[1]}u_{11} & {}^{[2]}u_{12} & {}^{[4]}u_{13} & {}^{[6]}u_{14} \\ {}^{[3]}l_{21} & a_{22} & a_{23} & a_{24} \\ {}^{[5]}l_{31} & a_{32} & a_{33} & a_{34} \\ {}^{[7]}l_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \rightarrow \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ l_{21} & {}^{[8]}u_{22} & {}^{[9]}u_{23} & {}^{[11]}u_{24} \\ l_{31} & {}^{[10]}l_{32} & a_{33} & a_{34} \\ l_{41} & {}^{[12]}l_{42} & a_{43} & a_{44} \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ l_{21} & u_{22} & u_{23} & u_{24} \\ l_{31} & l_{32} & {}^{[13]}u_{33} & {}^{[14]}u_{34} \\ l_{41} & l_{42} & {}^{[15]}l_{43} & a_{44} \end{bmatrix} \rightarrow \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ l_{21} & u_{22} & u_{23} & u_{24} \\ l_{31} & l_{32} & u_{33} & u_{34} \\ l_{41} & l_{42} & l_{43} & {}^{[16]}u_{44} \end{bmatrix}.$$

EXAMPLE ML.1.8.1 MATHEMATICA

```
(* Doolittle Factorization Method *)
a = {{1, 1, 0}, {2, 1, -1}, {3, -1, 0}};
n = Dimensions[a][[1]];
l = Table[0, {i, n}, {j, n}];
u = Table[0, {i, n}, {j, n}];
CheckAbort[
For[i = 1, i ≤ n, i++, l[[i, i]] = 1];
If[a[[1, 1]]==0, Abort[]]; u[[1, 1]] = a[[1, 1]];
For[j = 2, j ≤ n, j++, u[[1, j]] = a[[1, j]];
l[[j, 1]] = a[[j,1]]/a[[1,1]] ];
For[i = 2, i ≤ n - 1, i++,
u[[i, i]] = a[[i, i]] - Σ_{k=1}^{i-1} l[[i, k]]u[[k, i]];
If[u[[i, i]]==0, Abort[]];
For[j = i + 1, j ≤ n, j++,
u[[i, j]] = a[[i, j]] - Σ_{k=1}^{i-1} l[[i, k]]u[[k, j]];
l[[j, i]] = (a[[j,i]] - Σ_{k=1}^{i-1} l[[j,k]]u[[k,i]]) / u[[i,i]] ] ; ] ;
u[[n, n]] = a[[n, n]] - Σ_{k=1}^{n-1} l[[n, k]]u[[k, n]]; ,
Print["Factorization impossible"]]
SequenceForm["L= "MatrixForm[l], " U = "MatrixForm[u]]
SequenceForm["A= "MatrixForm[a], " L.U ="MatrixForm[l.u]]
```

$$\text{L}= \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 4 & 1 \end{pmatrix} \quad \text{U} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & -1 & -1 \\ 0 & 0 & 4 \end{pmatrix}$$

$$\text{A}= \begin{pmatrix} 1 & 1 & 0 \\ 2 & 1 & -1 \\ 3 & -1 & 0 \end{pmatrix} \quad \text{L.U} = \begin{pmatrix} 1 & 1 & 0 \\ 2 & 1 & -1 \\ 3 & -1 & 0 \end{pmatrix}$$

EXAMPLE ML.1.8.2 MATHEMATICA

```
(*DoolittleFactorizationMethod – "in place"*)
a = {{1, 1, 0}, {2, 1, −1}, {3, −1, 0}}; olda = a;
n = Dimensions[a][[1]];
CheckAbort[If[a[[1, 1]]==0, Abort[]];
For[j = 2, j ≤ n, j++, a[[j, 1]] = a[[j,1]]/a[[1,1]]];
For[i = 2, i ≤ n − 1, i++, a[[i, i]] = a[[i, i]] − Sum[a[[i, k]]a[[k, i]], {k, 1, i − 1}];
If[a[[i, i]]==0, Abort[]];
For[j = i + 1, j ≤ n, j++, a[[i, j]] = a[[i, j]] − Sum[a[[i, k]]a[[k, j]], {k, 1, i − 1}];
a[[j, i]] = (a[[j,i]] − Sum[a[[j,k]]a[[k,i]], {k, 1, i − 1}])/a[[i,i]]];];
a[[n, n]] = a[[n, n]] − Sum[a[[n, k]]a[[k, n]], {k, 1, n − 1}];
, Print["Factorization impossible"]]
l = Table[0, {i, n}, {j, n}];
u = Table[0, {i, n}, {j, n}];
For[i = 1, i ≤ n, i++, l[[i, i]] = 1;
For[j = 1, j ≤ i − 1, j++, l[[i, j]] = a[[i, j]]]];
For[i = 1, i ≤ n, i++,
For[j = i, j ≤ n, j++, u[[i, j]] = a[[i, j]]]];
SequenceForm["A = "MatrixForm[olda], " New A = L\\\U = "MatrixForm[a]]
SequenceForm["L = "MatrixForm[l], " U = "MatrixForm[u]]
SequenceForm["Test: L.U = "MatrixForm[l.u]]
```

$$\text{A} = \begin{pmatrix} 1 & 1 & 0 \\ 2 & 1 & -1 \\ 3 & -1 & 0 \end{pmatrix} \quad \text{New A = L\\U} = \begin{pmatrix} 1 & 1 & 0 \\ 2 & -1 & -1 \\ 3 & 4 & 4 \end{pmatrix}$$

$$\text{L} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 4 & 1 \end{pmatrix} \quad \text{U} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & -1 & -1 \\ 0 & 0 & 4 \end{pmatrix}$$

$$\text{Test: L.U} = \begin{pmatrix} 1 & 1 & 0 \\ 2 & 1 & -1 \\ 3 & -1 & 0 \end{pmatrix}$$

ML.1.9 Choleski's Factorization Method

If a square matrix A is symmetric and positive definite, then it has a special triangular decomposition.

One can prove that a matrix A is symmetric and positive definite if and only if it can be factored in the form

$$A = L \cdot L^t \tag{1.9.1}$$

where L is lower triangular with nonzero diagonal entries.

The factorization ((1.9.1)) is sometimes referred to as "taking the square root" of A. Writing out Equation (1.9.1) in components, one readily obtains

$$l_{ii} = \left(a_{ii} - \sum_{1 \le k \le i-1} l_{ik}^2 \right)^{1/2}$$

$$l_{ji} = \frac{1}{l_{ii}} \left(a_{ij} - \sum_{1 \le k \le i-1} l_{ik} l_{jk} \right)$$

$1 \le i \le n, \quad i+1 \le j \le n.$

See Examples (ML.1.9.1) and (ML.1.9.2).

EXAMPLE ML.1.9.1 MATHEMATICA

```
(*CholeskyDecompositionMethod–1*)
BeginPackage["LinearAlgebra`Cholesky`"]
LinearAlgebra`Cholesky`
CholeskyDecomposition::"usage"
For a symmetric positive definite matrix A, CholeskyDecomposition[A]
returns an upper-triangular matrix U such that A = Transpose[U] . U.
<< "LinearAlgebra`Cholesky`"
A:={{4,0,0},{0,9,1},{0,1,2}}; MatrixForm[A]
```

$$\begin{pmatrix} 4 & 0 & 0 \\ 0 & 9 & 1 \\ 0 & 1 & 2 \end{pmatrix}$$

```
U:=CholeskyDecomposition[A]; MatrixForm[U]
```

$$\begin{pmatrix} 2 & 0 & 0 \\ 0 & 3 & \frac{1}{3} \\ 0 & 0 & \frac{\sqrt{17}}{3} \end{pmatrix}$$

```
Transpose[U].U == A
True
```

EXAMPLE ML.1.9.2 MATHEMATICA

```
(*CholeskyDecompositionMethod–2*)
(*A = L.Transpose[L]*)
n = 3; A = {{4, 0, 0}, {0, 9, 1}, {0, 1, 2}};
CheckAbort[L = A; For[i = 2, i ≤ n, i++,
For[j = i, j ≤ n, j++, L[[i, j]] = 0]];
For[i = 1, i ≤ n, i++, tmp = A[[i, i]] - ∑_{k=1}^{i-1} L[[i, k]]^2;
If[tmp > 0, L[[i, i]] = √tmp, Abort[]];
For[j = i + 1, j ≤ n, j++, L[[j, i]]
= (A[[j,i]] - ∑_{k=1}^{i-1} L[[j,k]] L[[i,k]]) / L[[i,i]]]];
Print["A = ", MatrixForm[A], " L = ", MatrixForm[L]],
Print["With rounding errors, A is not positive definite"]]
```

$$\text{A} = \begin{pmatrix} 4 & 0 & 0 \\ 0 & 9 & 1 \\ 0 & 1 & 2 \end{pmatrix} \quad \text{L} = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & \frac{1}{3} & \frac{\sqrt{17}}{3} \end{pmatrix}$$

```
A == L.Transpose[L]
True
```

ML.1.10 Iterative Techniques for Solving Linear Systems

Consider the linear system

$$AX = B,$$

where $A \in \mathcal{M}_n(\mathbb{R})$ and $B \in \mathcal{M}_{n,1}(\mathbb{R})$.

An iterative technique to solve the above linear system begins with an initial approximation $X^{(0)}$ to the solution X, and generates a sequence of approximations $(X^{(k)})_{k\geq 0}$, that converges to X.

Writing the system $AX = B$ into an equivalent form

$$PX = QX + B,$$

where $A = P - Q$, after the initial approximation $X^{(0)}$ is selected, the sequence of approximate solutions is generated by computing

$$PX^{(k+1)} = QX^{(k)} + B,$$

$k = 0, 1, \ldots$

One can prove that, for $\|P^{-1}Q\| < 1$, the sequence $(X^{(k)})_{k\geq 0}$ is convergent.

In order to present some iterative techniques, we consider a standard decomposition of a matrix A,

$$A = L + D + U,$$

where:

- L has zero entries above and on the leading diagonal;
- D has zero entries above and below the leading diagonal;
- U has zero entries below and on the leading diagonal,

$$L = \begin{bmatrix} 0 & 0 & 0 \\ * & 0 & 0 \\ * & * & 0 \end{bmatrix}, \qquad D = \begin{bmatrix} * & 0 & 0 \\ 0 & * & 0 \\ 0 & 0 & * \end{bmatrix}, \qquad U = \begin{bmatrix} 0 & * & * \\ 0 & 0 & * \\ 0 & 0 & 0 \end{bmatrix}.$$

- **Jacobi's Iterative Method**

Write the system $AX = B$ into the equivalent form

$$DX = -(L+U)X + B.$$

From the recurrence formula

$$DX^{(k+1)} = -(L+U)X^{(k)} + B,$$

$k = 0, 1, \ldots,$ where

$$X^{(k)} = \begin{bmatrix} x_1^{(k)} \\ \vdots \\ x_n^{(k)} \end{bmatrix},$$

we obtain:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij} x_j^{(k)} \right), \tag{1.10.1}$$

$i = 1, \ldots, n;\ k = 0, 1, \ldots$ If $\| - D^{-1}(L+U)\|_\infty < 1,$ i.e.,

$$\max_{1 \leq i \leq n} \sum_{\substack{j=1 \\ j \neq i}}^{n} \left| \frac{a_{ij}}{a_{ii}} \right| < 1,$$

(the matrix A is strictly diagonally dominant), the Jacobi iterative method is convergent.

- **Gauss-Seidel Iterative Method**

Write the system $AX = B$ into the equivalent form

$$(L+D)X = -UX + B.$$

From

$$(L+D)X^{(k+1)} = -UX^{(k)} + B,$$

$k = 0, 1, \ldots,$ we obtain:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right), \tag{1.10.2}$$

$i = 1, \ldots, n$. If the matrix A is strictly diagonally dominant, then the Gauss-Seidel method is convergent.

Note that, there exist linear systems for which the Jacobi method is convergent but not the Gauss-Seidel method, and vice versa.

- **Relaxation Methods**

Let $\omega > 0$. Modifying the Gauss-Seidel procedure ((1.10.2)) to

$$x_i^{(k+1)} = (1-\omega)x_i^{(k)} + \frac{\omega}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right) \tag{1.10.3}$$

$i = 1, \ldots, n$, certain choice of positive ω will lead to significant faster convergence.

Methods involving ((1.10.3)) are called *relaxation methods.* For values of ω in $(0, 1)$, the procedure is called *under-relaxation method* and can be used to obtain convergence of some systems that are not convergent by the Gauss-Seidel method. For choice of $\omega > 1$, the procedure is called *over-relaxation method,* which is used to accelerate the convergence for systems that are convergent by Gauss-Seidel technique. These methods are abbreviated by *SOR* for *Successive Over-Relaxation.*

EXAMPLE ML.1.10.1 MATHEMATICA

```
(* Succesive Over Relaxation *)
n = 3; A = Table[Min[i, j], {i, n}, {j, n}]; B = {6, 11, 14};
CheckAbort[Print["A = ", MatrixForm[A], " B = ", MatrixForm[B]];
X = Table[0, {i, n}];
X0 = Table[0, {i, n}];
tol = 0.0001;
omega = 1.2;
numbiter = 20;
k = 1;
While[k ≤ numbiter,
For[i = 1, i ≤ n, i++,
X[[i]] = (1 − omega)X0[[i]]+
omega/A[[i,i]] (B[[i]] − Σ_{j=1}^{i−1} A[[i, j]]X[[j]] − Σ_{j=i+1}^{n} A[[i, j]]X0[[j]])];
If[Max[Abs[X − X0]] < tol, Print["X = ", MatrixForm[X]];
Abort[]]; X0 = X;
k++];
Print["Maximum number of iterartions exceeded"],
Null]
```

$$\text{A} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 2 & 3 \end{pmatrix} \quad \text{B} = \begin{pmatrix} 6 \\ 11 \\ 14 \end{pmatrix}$$

$$\text{X} = \begin{pmatrix} 0.999998 \\ 2.00004 \\ 2.99998 \end{pmatrix}$$

```
Max[Abs[X − X0]]
```

0.0000610555

```
Max[Abs[B − A.X]]
```

0.0000309092

```
(* Exact Solution *)
MatrixForm[{1, 2, 3}]
```

$$\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

ML.1.11 Eigenvalues and Eigenvectors

Let $A \in \mathcal{M}_n(\mathbb{R})$ and $I \in \mathcal{M}_n(\mathbb{R})$ be the identity matrix.

DEFINITION ML.1.11.1

The polynomial P defined by

$$P(\lambda) = \det(A - \lambda I)$$

is called the *characteristic polynomial* of the matrix A.

P is a polynomial of degree n.

DEFINITION ML.1.11.2

If P is the characteristic polynomial of a matrix A, the roots of P are called *eigenvalues* or *characteristic values* of A.

Let λ be an eigenvalue of the matrix A.

DEFINITION ML.1.11.3

If $X \in \mathcal{M}_{n,1}(\mathbb{R})$, $X \neq 0$, satisfies the equation

$$(A - \lambda I)X = 0,$$

then X is called an *eigenvector* or *characteristic vector* of A corresponding to the eigenvalue λ.

DEFINITION ML.1.11.4

The *spectral radius* $\rho(A)$ of a matrix A is defined by

$$\rho(A) = \max\left\{|\lambda| \;\middle|\; \lambda \text{ is an eigenvalue of } A\right\}.$$

The spectral radius is closely related to the norm of a matrix, as shown in the following theorem:

THEOREM ML.1.11.5

If $A \in \mathcal{M}_n(\mathbb{R})$, then

(1) $\sqrt{|\rho(A^t A)|} = \|A\|_2$;

(2) $\rho(A) \leq \|A\|$, **for any natural norm** $\|\cdot\|$.

EXAMPLE ML.1.11.6 MATHEMATICA

Eigenvalues[m] gives a list of the eigenvalues of the square matrix m.

Eigenvectors[m] gives a list of the eigenvectors of the square matrix m.

Eigensystem[m] gives a list $\{values, vectors\}$ of the eigenvalues and eigenvectors of the square matrix m.

```
(* Eigenvalues,Eigenvectors,Eigensystem *)
A = {{1, 2}, {3, 6}};
SequenceForm["MatrixForm[A] = ", MatrixForm[A]]
```

$$\text{MatrixForm[A]} = \begin{pmatrix} 1 & 2 \\ 3 & 6 \end{pmatrix}$$

```
SequenceForm["MatrixForm[Eigenvalues[A]] = ", MatrixForm[Eigenvalues[A]]]
```

$$\text{MatrixForm[Eigenvalues[A]]} = \begin{pmatrix} 7 \\ 0 \end{pmatrix}$$

```
MatrixForm[Solve[Det[A - λ * IdentityMatrix[2]] == 0, λ]]
```

$$\begin{pmatrix} \lambda \to 0 \\ \lambda \to 7 \end{pmatrix}$$

```
SequenceForm["MatrixForm[Eigenvectors[A]] = ", MatrixForm[Eigenvectors[A]]]
```

$$\text{MatrixForm[Eigenvectors[A]]} = \begin{pmatrix} 1 & 3 \\ -2 & 1 \end{pmatrix}$$

```
SequenceForm["Eigensystem[A] = ", Eigensystem[A]]
```

$$\text{Eigensystem[A]} = \{\{7, 0\}, \{\{1, 3\}, \{-2, 1\}\}\}$$

```
SequenceForm["MatrixForm[Eigensystem[A]] = ", MatrixForm[Eigensystem[A]]]
```

$$\text{MatrixForm[Eigensystem[A]]} = \begin{pmatrix} 7 & 0 \\ \{1, 3\} & \{-2, 1\} \end{pmatrix}$$

ML.1.12 Characteristic Polynomial: Le Verrier Method

This algorithm has been rediscovered and modified several times. In 1840, Urbain Jean Joseph Le Verrier provided the basic connection with Newton's identities.

Let $A \in \mathcal{M}_n(\mathbb{R})$. The *trace* of a matrix A denoted by $\operatorname{tr}(A)$ is defined by

$$\operatorname{tr}(A) = a_{11} + a_{22} + \cdots + a_{nn}.$$

Write the characteristic polynomial P of a matrix A in the form

$$P(\lambda) = \lambda^n - c_1\lambda^{n-1} - \cdots - c_{n-1}\lambda - c_n.$$

It is known that, if $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of A, then:

$$c_1 = \lambda_1 + \lambda_2 + \cdots + \lambda_n = \operatorname{tr}(A),$$

$$s_k = \lambda_1^k + \lambda_2^k + \cdots + \lambda_n^k = \operatorname{tr}(A^k),$$

$k = 1, \ldots, n$. Using the Newton formula

$$c_k = \frac{1}{k}\left(s_k - c_1 s_{k-1} - c_2 s_{k-2} - \cdots - c_{k-1} s_1\right),$$

$k = 2, \ldots, n$, we obtain:

$$\begin{aligned}
&c_1 = \mathrm{tr}(A),\\
&c_2 = \tfrac{1}{2}(\mathrm{tr}(A^2) - c_1\mathrm{tr}(A)),\\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\\
&c_n = \tfrac{1}{n}(\mathrm{tr}(A^n) - c_1\mathrm{tr}(A^{n-1}) - \cdots - c_{n-1}\mathrm{tr}(A)).
\end{aligned}$$

ML.1.13 Characteristic Polynomial: Fadeev-Frame Method

J. M. Souriau, also from France, and J. S. Frame, from Michigan State University, independently modified the algorithm to its present form. Souriau's formulation was published in France in 1948, and Frame's method appeared in the United States in 1949. Paul Horst (USA, 1935) along with Faddeev and Sominskii (USSR, 1949) are also credited with rediscovering the technique. Although the algorithm is intriguingly beautiful, it is not practical for floating-point computations. The Fadeev-Frame algorithm is closely related to the Le Verrier method.

Let

$$\det(A - \lambda I_n) = (-1)^n P(\lambda),$$

where

$$P(\lambda) = \lambda^n - c_1\lambda^{n-1} - \cdots - c_{n-1}\lambda - c_n.$$

Using the notations:

$$\begin{array}{lll}
A_1 = A, & c_1 = \mathrm{tr}(A_1), & B_1 = A_1 - c_1 I_n,\\
A_2 = AB_1, & c_2 = \frac{1}{2}\mathrm{tr}(A_2), & B_2 = A_2 - c_2 I_n,\\
\vdots & \vdots & \vdots\\
A_n = AB_{n-1}, & c_n = \frac{1}{n}\mathrm{tr}(A_n), & B_n = A_n - c_n I_n,
\end{array}$$

and taking into account the fact that the matrix A is a solution to its characteristic equation [1] (Cayley -Hamilton theorem), we obtain

$$A^n - c_1 A^{n-1} - \cdots - c_{n-1}A - c_n I_n = P(A) = 0.$$

The relation $B_n = A_n - c_n I_n$ is a control test,

$$B_n = 0.$$

Furthermore, from the relation

$$\det(A) = (-1)^n P(0) = (-1)^{n+1} c_n, \tag{1.13.1}$$

[1] In 1896 Frobenius became aware of Cayley's 1858 *Memoir on the theory of matrices* and after this started to use the term *matrix*. Despite the fact that Cayley only proved the Cayley-Hamilton theorem for 2×2 and 3×3 matrices, Frobenius generously attributed the result to Cayley (Frobenius, in 1878, had been the first to prove the general theorem). Hamilton proved a special case of the theorem, the 4×4 case, in the course of his investigations into quaternions.

we can determine the determinant of the matrix A.

If $\det(A) \neq 0$ then, from the relations

$$AB_{n-1} = A_n = c_n I_n,$$

using the formula

$$A^{-1} = \frac{1}{c_n} B_{n-1}, \tag{1.13.2}$$

we obtain the inverse of the matrix A.

Also, note that $(-1)^{n+1}B_{n-1}$ is the adjoint of the matrix A.

EXAMPLE ML.1.13.1 MATHEMATICA

```
(* Faddev - Frame *)
n = 3; A = Table[Min[i, j], {i, n}, {j, n}]; c = Table[0, {i, n}];
tr[x_]:= Sum_{i=1}^{Length[x]} x[[i, i]]; B = IdentityMatrix[n];
For[k = 1, k <= n - 1, k++, c[[k]] = tr[A.B]/k; B = A.B - c[[k]]IdentityMatrix[n]];
c[[n]] = tr[A.B]/n; detA = (-1)^(n+1) c[[n]]; adjA = (-1)^(n+1) B;
SequenceForm["A = ", MatrixForm[A]]
SequenceForm["adj(A) = ", MatrixForm[adjA]]
SequenceForm["CharPol(x) = ", x^n - Sum_{i=1}^{n} c[[i]] x^(n-i)]
SequenceForm["Det(A) = ", detA]
CheckAbort[If[detA==0, Abort[], invA = adjA/detA];
SequenceForm["A^(-1) = ", MatrixForm[invA]], Print["Singular matrix"]]
```

$$\text{A} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 2 & 3 \end{pmatrix}$$

$$\text{adj(A)} = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix}$$

$$\text{CharPol(x)} = -1 + 5x - 6x^2 + x^3$$

$$\text{Det(A)} = 1$$

$$\text{A\^{}(-1)} = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix}$$

ML.2

Solutions of Nonlinear Equations

ML.2.1 Introduction

In this chapter we consider one of the most basic problems in numerical approximation, the root-finding problem.

A few type of nonlinear equations can be solved by using direct algebraic methods. In general, algebraic equations of fifth and of higher orders cannot be solved by means of radicals. Moreover, there exists no explicit formula for finding the roots of nonalgebraic (transcendental) equations such as, $e^x + x = 0$, $x \in \mathbb{R}$. Therefore, root-finding invariably proceeds by iteration, and this is equally true in one or in many dimensions.

Let x^* be a real number and $(x_k)_{k\geq 0}$ be a sequence of real numbers that converges to x^*.

DEFINITION ML.2.1.1

If positive constants p and λ exist such that

$$\lim_{k\to\infty} \frac{|x_{k+1} - x^*|}{|x_k - x^*|^p} = \lambda,$$

then the sequence $(x_k)_{k\geq 0}$ is said to *converge to* **x^*** *of order* **p,** **with** *asymptotic constant* **λ.**

An iterative technique of the form $x_{k+1} = f(x_k)$, $k \in \mathbb{N}$, is said to be of order p if the sequence $(x_k)_{k\geq 0}$ converges to the solution $x^* = f(x^*)$ of order p.

In general a sequence of a high order of convergence converges more rapidly than a sequence with a lower order. Two cases of order are given special attention, namely the linear ($p = 1$) and the quadratic ($p = 2$).

Consider that ε is a bound of the maximum size of the error. We have to stop the calculations when the following condition is satisfied

$$|x_k - x^*| < \varepsilon.$$

But such a criterion cannot be used because the root x^* is not known. A typical *stopping criterion* is to stop when the difference between two successive iterates satisfy the inequality

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

We can also use as stopping criterion

$$|x_k - x_{k+1}| \leq \varepsilon |x_k|,$$

etc.

Unfortunately none of these criteria guarantees the required precision.

The following section deals with several traditional iterative methods.

ML.2.2 Method of Successive Approximation

Let $F : [a, b] \to \mathbb{R}$ be a function. A number x is said to be a *fixed point* of the function F if

$$F(x) = x.$$

Root-finding problems and fixed-point problems are equivalent classes in the following sense:

A point x is a root of the equation

$$f(x) = 0$$

if and only if it is a fixed point of the function

$$F(x) = x - f(x).$$

Although the problem we wish to solve is in the root-finding form, the fixed-point form is easier to analyze and certain fixed-point choices lead to very powerful root-finding techniques.

THEOREM ML.2.2.1

If $F : [a, b] \to [a, b]$ has derivative such that $|F'(x)| \leq \alpha < 1$, for all $x \in (a, b)$, then the function F has a unique fixed point x^*, and the sequence $(x_k)_{k>0}$, defined by:

$$x_{k+1} = F(x_k), \quad k \in \mathbb{N}, \quad x_0 \in [a, b].$$

converges to x^*.

Proof. Using Lagrange's mean value theorem it follows that for all $x, y \in [a, b]$ there exists $c \in (a, b)$ such that

$$|F(x) - F(y)| = |F'(c)| \cdot |x - y| \leq \alpha |x - y|.$$

The function F is a contraction. The proof is concluded by using the Banach fixed point theorem.

THEOREM ML.2.2.2

Let $F \in C^p[a,b]$, $p \geq 2$, and $x^* \in (a,b)$ be a fixed point of F. If

$$F'(x^*) = F''(x^*) = \cdots = F^{(p-1)}(x^*) = 0, \quad F^{(p)}(x^*) \neq 0,$$

then, for x_0 sufficiently close to x^*, the sequence $(x_k)_{k>0}$, where

$$x_{k+1} = F(x_k), \quad k \in \mathbb{N}, \quad x_0 \in [a,b],$$

converges to x^*, of order p.

Proof. Since $|F'(x^*)| = 0 < 1$, there exists a ball centered at x^* on which F be a contraction. By virtue of Banach's fixed point theorem it follows that the sequence $(x_k)_{k\geq 0}$ converges to x^*. Using Taylor's formula it follows that there exists $c_k \in (a,b)$ such that

$$\begin{aligned} x_{k+1} &= F(x_k) \\ &= F(x^*) + \sum_{i=1}^{p-1} F^{(i)}(x^*)\frac{(x_k - x^*)^i}{i!} + F^{(p)}(c_k)\frac{(x_k - x^*)^p}{p!} \\ &= x^* + 0 + F^{(p)}(c_k)\frac{(x_k - x^*)^p}{p!}, \end{aligned}$$

hence

$$\lim_{k\to\infty} \frac{|x_{k+1} - x^*|}{|x_k - x^*|^p} = \lim_{k\to\infty} \frac{|F^{(p)}(c_k)|}{p!} = \frac{|F^{(p)}(x^*)|}{p!} > 0.$$

Consequently, the method has order p.

ML.2.3 The Bisection Method

THEOREM ML.2.3.1

If $f \in C[a,b]$ and $f(a)f(b) < 0$ then the sequence $(x_k)_{k\geq 0}$, defined by:
$x_0 = a$, $y_0 = b$;
x_{k+1}, $y_{k+1} \in \left\{x_k, \dfrac{x_k + y_k}{2}, y_k\right\}$, such that:
$y_{k+1} - x_{k+1} = \dfrac{y_k - x_k}{2}$,
$f(x_{k+1})f(y_{k+1}) \leq 0$, $k \in \mathbb{N}$,
converges to a root x^* of the equation $f(x) = 0$. Moreover, we have

$$|x_k - x^*| \leq \frac{b-a}{2^k}, \quad k \in \mathbb{N}.$$

Proof. The sequence $(x_k)_{k\geq 0}$ is increasing, the sequence $(y_k)_{k\geq 0}$ is decreasing such that $y_k - x_k \to 0$. It follows that they converge to the same limit x^*. From the conditions $f(x_k)f(y_k) \leq$

0, $k \to \infty$, we deduce

$$f^2(x^*) \leq 0,$$

that is

$$f(x^*) = 0.$$

Since

$$y_k - x_k = \frac{b-a}{2^k},$$

it follows that

$$|x_k - x^*| \leq \frac{b-a}{2^k}.$$

This is only a bound for the arising error in approximation process. The actual error can be much smaller.

□

The bisection method, is also called the *binary-search method.* It is very slow in converging. However, the method guarantees convergence to a solution and, for this reason, it is used as a "starter" for more efficient methods.

ML.2.4 The Newton-Raphson Method

The *Newton-Raphson method* (or simply Newton's method) is one of the most powerful and well-known techniques used for a root-finding problem.

THEOREM ML.2.4.1

If $x_0 \in [a,b]$ and $f \in C^2[a,b]$ satisfies the conditions:
(1) **$f(a)f(b) < 0$;**
(2) **f' and f'' have no roots in $[a,b]$,**
(3) **$f(x_0)f''(x_0) > 0$,**
then the equation $f(x) = 0$ has a unique solution $x^* \in (a,b)$, and the sequence $(x_k)_{k\geq 0}$ defined by

$$x_{k+1} = x_k - f(x_k)/f'(x_k), \qquad k \in \mathbb{N},$$

converges to x^*.

We shall prove that the Newton method is of order 2. Indeed, using Taylor's formula, we have

$$0 = f(x^*) = f(x_k) + (x^* - x_k)f'(x_k) + (x^* - x_k)^2 \frac{f''(c_k)}{2},$$

where $|x^* - c_k| \leq |x^* - x_k|$, which we rewrite in the form

$$x_{k+1} - x^* = (x^* - x_k)^2 \frac{f''(c_k)}{2f'(x_k)},$$

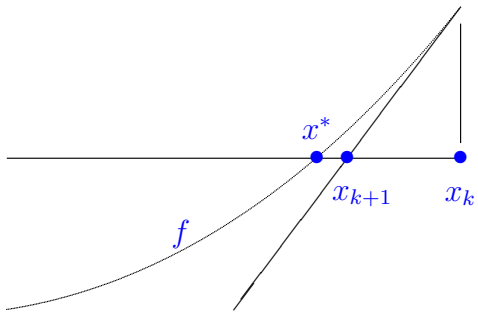


Figure ML.2.1: Newton's Method

hence

$$\lim_{k\to\infty} \frac{|x_{k+1} - x^*|}{|x_k - x^*|^2} = \frac{|f''(x^*)|}{2|f'(x^*)|} > 0,$$

that is, the Newton method has the order 2.

We can also define a sequence of approximations by the formula

$$x_{k+1} = x_k - f(x_k)/f'(x_0),$$

$k \in \mathbb{N}$.

This formula needs to know the value of the derivative only at the starting point x_0. It is known as the *simplified Newton's formula.*

ML.2.5 The Secant Method

Newton's method is an extremely powerful technique, but it has a major difficulty: it requires the value of the derivative of the function at each approximation. To circumvent the problem of the derivative evaluation in Newton's method, we derive a slight variation.

Using the approximation

$$\frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$$

for $f'(x_k)$ in Newton's formula, gives

$$x_{k+1} = \frac{x_{k-1}f(x_k) - x_k f(x_{k-1})}{f(x_k) - f(x_{k-1})}, \qquad k \in \mathbb{N}^*.$$

The technique using the above formula is called the *secant method.* Starting with two initial approximations x_0 and x_1, the approximation x_{k+1} is the x-intercept of the line joining $(x_{k-1}, f(x_{k-1}))$ and $(x_k, f(x_k))$, $k = 0, 1, \ldots$

ML.2.6 False Position Method

The method of *False Position* (also called *Regula Falsi*) generates approximation in a similar way to Secant method, but it provides a test to ensure that the root lies between two successive iterations. The method can be described as follows:

Choose x_0, x_1 such that $f(x_0)f(x_1) < 0$. Let x_2 be the x-intercept of the line joining $(x_0, f(x_0))$ and $(x_1, f(x_1))$. If $f(x_0)f(x_2) < 0$ then let x_3 be the x-intercept of the line joining $(x_0, f(x_0))$ and $(x_2, f(x_2))$. If $f(x_1)f(x_2) < 0$ then let x_3 be x-intercept of the line joining $(x_1, f(x_1))$ and $(x_2, f(x_2))$, etc.

THEOREM ML.2.6.1

Let $f \in C^2[a,b]$, $x_0, x_1 \in [a,b]$, $x_0 \neq x_1$. If the following conditions are satisfied:

(1) $f''(x) \neq 0, \quad \forall x \in [a,b]$;

(2) $f(x_0)f''(x_0) > 0$ **(Fourier condition);**

(3) $f(x_0)f(x_1) < 0$,

then the equation $f(x) = 0$ has a unique root x^* between x_0 and x_1, and the sequence $(x_k)_{k\geq 0}$ defined by

$$x_{k+1} = \frac{x_0 f(x_k) - x_k f(x_0)}{f(x_k) - f(x_0)}, \qquad k \in \mathbb{N},$$

converges to x^*.

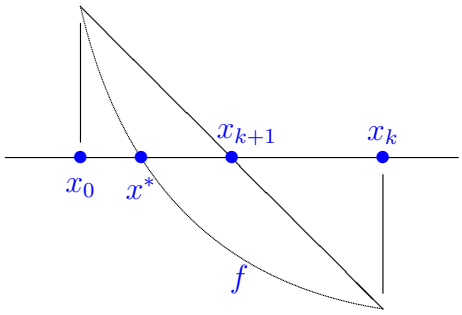


Figure ML.2.2: False Position Method

ML.2.7 The Chebyshev Method

Let $f \in C^{p+1}[a,b]$ and x^* be a solution of the equation $f(x) = 0$. Assume that f' has no roots in $[a,b]$, and let h be the inverse of the function f. By virtue of Taylor's formula we obtain:

$$h(t) = h(y) + \sum_{i=1}^{p} \frac{h^{(i)}(y)}{i!}(t-y)^i + \frac{h^{(p+1)}(c)}{(p+1)!}(t-y)^{p+1},$$

where $|c - y| < |t - y|$.

For $t = 0$ and $y = f(x)$ we obtain

$$x^* = x + \sum_{i=1}^{p} \frac{h^{(i)}(f(x))}{i!}(-1)^i f^i(x) + \frac{h^{(p+1)}(c)}{(p+1)!}(-1)^{p+1} f^{p+1}(x),$$

where $|c - f(x)| \leq |f(x)|$. Using the notation

$$F(x) = x + \sum_{i=1}^{p} \frac{(-1)^i}{i!} a_i(x) f^i(x),$$

where $a_i(x) = h^{(i)}(f(x))$.

The *Chebyshev method* can be obtained by defining the sequence $(x_k)_{k\geq 0}$ such that

$$x_{k+1} = F(x_k), \qquad k \in \mathbb{N}.$$

If $h^{(p+1)}(0) \neq 0$, then the Chebyshev method has the order of convergence $p+1$.

In order to calculate the coefficients $a_i(x),\ i = 1, \ldots, p$, we differentiate p-times, the identity

$$h(f(x)) = x, \quad \forall x \in [a, b].$$

We obtain:

$$h'(f(x)) \cdot f'(x) = 1,$$

i.e.,

$$a_1(x) f'(x) = 1, \qquad a_1(x) = \frac{1}{f'(x)},$$

and

$$a_i = \frac{1}{f'(x)} \frac{\mathrm{d}}{\mathrm{d}x} a_{i-1}(x), \qquad i = 1, \ldots p,$$

where $a_0(x) = x$.

For $p = 1$ we obtain

$$x_{k+1} = x_k - f(x_k)/f'(x_k),$$

$k \in \mathbb{N}$, i.e., the Newton method.

For $p = 2$ we obtain a Chebyshev method of order 3:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} - \frac{f''(x_k) f^2(x_k)}{2(f'(x_k))^3},$$

$k \in \mathbb{N}$.

The Chebyshev method needs more calculations, but it converges more rapidly.

EXAMPLE ML.2.7.1 MATHEMATICA

```
(*Chebyshev*)
p:=3; Array[a, p];
For[a[1] = 1/f'[t]; i = 2, i <= p, i++,
a[i] = Simplify[D[a[i-1],t]/f'[t]];
F[j_, x_]:=t + Sum[(-1)^i/i! a[i] f[t]^i /. t -> x, {i, 1, j}]
TableForm[Table[a[i], {i, 1, 3}]]
1/f'[t]
-f''[t]/f'[t]^3
(3 f''[t]^2 - f'[t] f^(3)[t])/f'[t]^5
TableForm[Table[F[i, x], {i, 1, 3}]]
x - f[x]/f'[x]
x - f[x]/f'[x] - f[x]^2 f''[x]/(2 f'[x]^3)
x - f[x]/f'[x] - f[x]^2 f''[x]/(2 f'[x]^3) - f[x]^3 (3 f''[x]^2 - f'[x] f^(3)[x])/(6 f'[x]^5)
```

ML.2.8 Numerical Solutions of Nonlinear Systems of Equations

Consider the functions $f_i : \mathcal{M}_{n,1}(\mathbb{R}) \to \mathbb{R}$, $i = 1, \ldots, n$, and $F : \mathcal{M}_{n,1}(\mathbb{R}) \to \mathcal{M}_{n,1}(\mathbb{R})$, where $F = [f_1, \ldots, f_n]^t$. The functions $f_1, f_2, \ldots, f_n$ are called the coordinate functions of F. Consider the system of equations

$$\begin{cases} f_1([x_1, \ldots, x_n]^t) &= 0 \\ f_2([x_1, \ldots, x_n]^t) &= 0 \\ \vdots & \vdots \ \vdots \\ f_n([x_1, \ldots, x_n]^t) &= 0 \end{cases} \tag{2.8.1}$$

in unknown $x = [x_1, x_2, \ldots, x_n]^t \in \mathcal{M}_{n,1}(\mathbb{R})$. The system (2.8.1) has the form

$$F(x) = 0. \tag{2.8.2}$$

By denoting

$$g(x) = F(x) + x,$$

one can see that the solutions of equation (2.8.2) are the fixed points of $g = [g_1, \ldots, g_n]^t$.

Let

$$\|x\|_\infty := \max_{1 \le i \le n} |x_i|$$

and x^* be a solution of equation (2.8.2). Define the ball

$$D := \{x \in \mathcal{M}_{n,1}(\mathbb{R}) \mid \|x - x^*\|_\infty \leq r\},$$

$r > 0$.

THEOREM ML.2.8.1

If the functions $g_i : \mathcal{M}_{n,1}(\mathbb{R}) \to \mathbb{R}$, $(i = 1, 2, \ldots, n)$, satisfy the conditions:

$$\left|\frac{\partial g_i}{\partial x_j}(x)\right| \leq \frac{\lambda}{n}$$

$(\lambda < 1, \quad i, j = 1, \ldots, n,)$ for all $x \in D$, then equation (2.8.2) has a unique solution $x^* \in D$. Moreover for all point $x^{(0)} \in D$ the sequence $(x^{(k)})_{k\geq 0}$,

$$x^{(k+1)} = g(x^{(k)}), \qquad k \in \mathbb{N},$$

converges to the solution x^*.

Proof. We prove that the function g is a λ-contraction on D. Let $x, y \in D$. We have:

$$|g_i(x) - g_i(y)| \leq \sum_{j=1}^{n} \left|\frac{\partial g_i}{\partial x_j}(\xi_i)\right| |x_j - y_j| \leq \frac{\lambda}{n} \sum_{j=1}^{n} |x_j - y_j| \leq \lambda \|x - y\|_\infty,$$

$i = 1, 2, \ldots, n$. Hence,

$$\|g(x) - g(y)\|_\infty \leq \lambda \|x - y\|_\infty.$$

Now, let us prove that $g(D) \subset D$. Let $x \in D$, then

$$\|g(x) - x^*\|_\infty = \|g(x) - g(x^*)\|_\infty \leq \lambda \|x - x^*\|_\infty < \lambda r < r,$$

hence $g(x) \in D$. Using the Banach fixed point theorem, the sequence $(x^{(k)})_{k\geq 0}$ converges to the unique fixed point of g in D.

An estimate of the error is given by

$$\|x^{(k)} - x^*\|_\infty < \frac{\lambda^k}{1 - \lambda} \|x^{(1)} - x^{(0)}\|_\infty, \quad k \in \mathbb{N}.$$

ML.2.9 Newton's Method for Systems of Nonlinear Equations

Let $f_1, f_2, \ldots, f_n \in C^1(\mathcal{M}_{n,1}(\mathbb{R}))$. In addition to the notations used in Section (ML.2.8) we need the following definition

$$F'(x) \overset{def.}{=} \left[\frac{\partial f_i}{\partial x_j}(x)\right]_{1\leq i,j\leq n}.$$

Assume that at the fixed point x^* there exists the matrix $(F'(x^*))^{-1}$. Taking

$$g(x) = x - (F'(x^*))^{-1} \cdot F(x),$$

we can see that

$$\frac{\partial g_i}{\partial x_j}(x^*) = 0,$$

$i, j = 1, 2, \ldots, n$. Since F is continuous, there exists a ball D centered at x^* such that the conditions of Theorem (ML.2.8.1) are satisfied for

$$g(x) = x - (F'(x))^{-1} \cdot F(x).$$

For a certain value x_0, sufficiently close to x^*, the sequence $(x^{(k)})_{k\geq 0}$, defined by

$$x^{(k+1)} = x^{(k)} - (F'(x^{(k)}))^{-1} \cdot F(x^{(k)}), \qquad k \in \mathbb{N},$$

converges to x^*.

In the case $n = 2$, we have:

$$F(x, y) = [f(x, y), g(x, y)]^t,$$

$$F'(x, y) = \begin{bmatrix} f'_x & f'_y \\ g'_x & g'_y \end{bmatrix},$$

$$(F'(x, y))^{-1} = \frac{1}{f'_x g'_y - f'_y g'_x} \begin{bmatrix} g'_y & -f'_y \\ -g'_x & f'_x \end{bmatrix},$$

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)}, y^{(k)}) g'_y(x^{(k)}, y^{(k)}) - g(x^{(k)}, y^{(k)}) f'_y(x^{(k)}, y^{(k)})}{f'_x(x^{(k)}, y^{(k)}) g'_y(x^{(k)}, y^{(k)}) - g'_x(x^{(k)}, y^{(k)}) f'_y(x^{(k)}, y^{(k)})},$$

$$y^{(k+1)} = y^{(k)} + \frac{f(x^{(k)}, y^{(k)}) g'_x(x^{(k)}, y^{(k)}) - g(x^{(k)}, y^{(k)}) f'_x(x^{(k)}, y^{(k)})}{f'_x(x^{(k)}, y^{(k)}) g'_y(x^{(k)}, y^{(k)}) - g'_x(x^{(k)}, y^{(k)}) f'_y(x^{(k)}, y^{(k)})},$$

$k \in \mathbb{N}$.

The weakness of Newton's method arises from the need to compute and invert the matrix $F'(x^{(k)})$ at each step. In practice, explicit calculation of $(F'(x^{(k)}))^{-1}$ is avoided by performing the operation in a two-step manner:

– Step (1) A vector $y^{(k)} = [y_1^{(k)}, \ldots, y_n^{(k)}]^T$, is found which satisfies

$$F'(x^{(k)}) y^{(k)} = -F(x^{(k)});$$

– Step (2) The vector $x^{(k+1)}$ is calculated by the formula

$$x^{(k+1)} = x^{(k)} + y^{(k)}.$$

ML.2.10 Steepest Descent Method

The *steepest descent* method (also called the *gradient method*) determines a local minimum for a function $\Psi : \mathbb{R}^n \to \mathbb{R}$. Note that the system (2.8.1) has a solution precisely when the function Ψ defined by

$$\Psi = f_1^2 + \cdots + f_n^2$$

has the minimal value zero.

The method of Steepest Descent for finding a local minimum for an arbitrary function $\Psi : \mathbb{R}^n \to \mathbb{R}$ can be intuitively described as follows:

(1) Evaluate Ψ at an initial approximation $x^{(0)}$;

(2) Determine a direction from $x^{(0)}$ that results in a decrease in the value of Ψ;

(3) Move an appropriate distance in this direction and call the new vector $x^{(1)}$;

Repeat steps (1) to (3).

More precisely:

Consider a starting value $x^{(0)}$. Find the minimum of the function

$$t \mapsto \Psi(x^{(0)} - t \operatorname{grad} \Psi(x^{(0)})).$$

Let t_0 be the smallest positive root of the equation

$$\frac{d}{dt}(\Psi(x^{(0)} - t \operatorname{grad} \Psi(x^{(0)}))) = 0.$$

Define

$$x^{(1)} = x^{(0)} - t_0 \operatorname{grad} \Psi(x^{(0)}),$$

and, step by step, denoting by t_k the smallest nonnegative root of the equation

$$\frac{d}{dt}(\Psi(x^{(k)} - t \operatorname{grad} \Psi(x^{(k)}))) = 0,$$

take

$$x^{(k+1)} = x^{(k)} - t_k \operatorname{grad} \Psi(x^{(k)}), \quad k \in \mathbb{N}.$$

The Steepest Descent converges only linearly. As a consequence, the method is used to find sufficiently accurate starting approximation for the Newton based techniques.

ML.3

Elements of Interpolation Theory

Sometimes we know the values of a function f at a set of numbers $x_0 < \cdots < x_n$, but we do not have an analytic expression for f. For example, the values $f(x_i)$ might be obtained from some physical measurement. The task now is to estimate $f(x)$ for an arbitrary x.

If the required x lies in the interval (x_0, x_n) the problem is called *interpolation;* if x is outside the interval $[x_0, x_n]$, it is called *extrapolation.* Interpolation must model the function by some known function called *interpolator.* By far most common among the functions used as interpolators are polynomials. Rational functions and trigonometric polynomials also turn out to be extremely useful in interpolation.

We would like to emphasize the contribution given to the Interpolation Theory by the members of the school of thought in the Theory of Allure, Convexity and Interpolation founded by Elena Popoviciu.

ML.3.1 Lagrange Interpolation

Let $x_0 < \cdots < x_n$ be a set of distinct real numbers and f be a function whose values are given at these numbers.

DEFINITION ML.3.1.1

The polynomial P of degree at most n such that

$$P(x_i) = f(x_i), \qquad i = 0, \ldots, n,$$

is called the *Lagrange Interpolating Polynomial* (or simply, Lagrange Polynomial) of the function f with respect to $x_0, \ldots, x_n$.

The numbers x_i are called *mesh points* or *interpolation points.*

Consider the polynomial $P : \mathbb{R} \to \mathbb{R}$, $P(x) = a_0 + a_1 x + \cdots + a_n x^n$. The Lagrange interpolating conditions

$$P(x_i) = f(x_i), \qquad i = 0, \ldots, n,$$

are equivalent to the system

$$\begin{cases} a_0 + a_1x_0 + \cdots + a_nx_0^n = f(x_0) \\ a_0 + a_1x_1 + \cdots + a_nx_1^n = f(x_1) \\ \vdots \qquad \vdots \qquad\qquad \vdots \qquad \vdots \\ a_0 + a_1x_n + \cdots + a_nx_n^n = f(x_n) \end{cases}$$

in unknowns $a_0, \ldots, a_n$.

Since the *Vandermonde determinant*

$$V(x_0, \ldots, x_n) = \begin{vmatrix} 1 & x_0 & \cdots & x_0^n \\ 1 & x_1 & \cdots & x_1^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \cdots & x_n^n \end{vmatrix}$$

is different from zero (the points x_i are distinct) the above system has a unique solution. The uniqueness of the Lagrange Polynomial allows us to denote it by the symbol

$$\boldsymbol{L}(x_0, \ldots, x_n; f).$$

DEFINITION ML.3.1.2

The *divided difference* of the function f with respect to the points $x_0, \ldots, x_n$, is defined to be the coefficient at x^n in the Lagrange interpolating polynomial $L(x_0, \ldots, x_n; f)$ and is denoted by

$$[x_0, \ldots, x_n; f].$$

For example,

$$[x_0; f] = f(x_0), \quad [x_0, x_1; f] = \frac{f(x_1) - f(x_0)}{x_1 - x_0}.$$

REMARK ML.3.1.3

In the case when the points x_i are not distinct, see (ML.3.6.2). For instance,

$$[x_0, x_0, x_0;\ f] = \frac{f''(x_0)}{2}.$$

ML.3.2 Some Forms of the Lagrange Polynomial

From the interpolating conditions:

$$\boldsymbol{L}(x_0, \ldots, x_n; f)(x_i) = f(x_i), \qquad i = 0, \ldots, n,$$

we obtain the system

$$\begin{cases} a_0 + a_1x_0 + \cdots + a_nx_0^n = f(x_0) \\ a_0 + a_1x_1 + \cdots + a_nx_1^n = f(x_1) \\ \vdots \quad\quad \vdots \quad\quad\quad \vdots \quad\quad \vdots \\ a_0 + a_1x_n + \cdots + a_nx_n^n = f(x_n) \\ a_0 + a_1x + \cdots + a_nx^n = \boldsymbol{L}(x_0, \ldots, x_n; f)(x) \end{cases}$$

in unknowns $a_0, \ldots, a_n$. The system has a solution if

$$\begin{vmatrix} 1 & x_0 & \cdots & x_0^n & f(x_0) \\ 1 & x_1 & \cdots & x_1^n & f(x_1) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & x_n & \cdots & x_n^n & f(x_n) \\ 1 & x & \cdots & x^n & \boldsymbol{L}(x_0, \ldots, x_n; f)(x) \end{vmatrix} = 0,$$

hence, we obtain

$$\boldsymbol{L}(x_0, \ldots, x_n; f)(x) = -\frac{\begin{vmatrix} 1 & x_0 & \cdots & x_0^n & f(x_0) \\ 1 & x_1 & \cdots & x_1^n & f(x_1) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & x_n & \cdots & x_n^n & f(x_n) \\ 1 & x & \cdots & x^n & 0 \end{vmatrix}}{V(x_0, \ldots, x_n)} \tag{3.2.1}$$

□

Consider the polynomials $\ell_i \in \mathbb{R}[X], \quad i = 0, \ldots, n,$

$$\ell_i(x) = \frac{(x - x_0) \ldots (x - x_{i-1})(x - x_{i+1}) \ldots (x - x_n)}{(x_i - x_0) \ldots (x_i - x_{i-1})(x_i - x_{i+1}) \ldots (x_i - x_n)}.$$

It is clear that:

$$\ell_i(x_k) = \delta_{ik}, \qquad i, k = 0, \ldots n,$$

and hence the polynomial

$$P = \sum_{i=0}^{n} f(x_i)\ell_i$$

satisfies the relations:

$$P(x_k) = f(x_k), \qquad k = 0, \ldots, n.$$

Since the Lagrange Polynomial has a unique representation we obtain $P = \boldsymbol{L}(x_0, \ldots, x_n; f)$. Hence

$$\boldsymbol{L}(x_0, \ldots, x_n; f) = \sum_{i=0}^{n} f(x_i)\ell_i. \tag{3.2.2}$$

The polynomials ℓ_i are called the *fundamental polynomials of the Lagrange interpolation.* By letting

$$\ell(x) = (x - x_0)(x - x_1) \ldots (x - x_n),$$

we obtain

$$\ell_i(x) = \frac{\ell(x)}{(x - x_i)\ell'(x_i)},$$

and consequently

$$\boldsymbol{L}(x_0, \ldots, x_n; f)(x) = \sum_{i=0}^{n} f(x_i) \frac{\ell(x)}{(x - x_i)\ell'(x_i)}. \tag{3.2.3}$$

□

Consider the polynomial

$$Q = \boldsymbol{L}(x_0, \ldots, x_n; f) - (X - x_0) \ldots (X - x_{n-1})[x_0, \ldots, x_n; f].$$

Note that the polynomial Q is of degree at most $n - 1$ and it satisfies the relations:

$$Q(x_i) = f(x_i), \qquad i = 0, \ldots, n - 1.$$

It follows that

$$Q = \boldsymbol{L}(x_0, \ldots, x_{n-1}; f),$$

and hence

$$\begin{gathered}\mathbf{L}(x_0, \ldots, x_n; f) \\ = \mathbf{L}(x_0, \ldots, x_{n-1}; f) + (X - x_0) \ldots (X - x_{n-1})[x_0, \ldots, x_n; f].\end{gathered}$$

Adding the equalities:

$$\begin{array}{lcl}
\mathbf{L}(x_0, x_1; f) & = & f(x_0) + (X - x_0)[x_0, x_1; f] \\
\mathbf{L}(x_0, x_1, x_2; f) & = & \mathbf{L}(x_0, x_1; f) + (X - x_0)(X - x_1)[x_0, x_1, x_2; f] \\
\ldots\ldots\ldots\ldots & \ldots & \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
\mathbf{L}(x_0, \ldots, x_n; f) & = & \mathbf{L}(x_0, \ldots, x_{n-1}; f) \\
 & & +(X - x_0) \ldots (X - x_{n-1})[x_0, \ldots, x_n; f],
\end{array}$$

we obtain the *Newton Interpolatory Divided Difference Formula:*

$$\begin{array}{rl}
 & \mathbf{L}(x_0, \ldots, x_n; f) \\
= & f(x_0) + (X - x_0)[x_0, x_1; f] + \cdots \\
 & +(X - x_0) \ldots (X - x_{n-1})[x_0, \ldots, x_n; f].
\end{array} \tag{3.2.4}$$

EXAMPLE ML.3.2.1

$$\begin{aligned}
\mathbf{L}(x_0, x_1; f) &= -\begin{vmatrix} 1 & x_0 & f(x_0) \\ 1 & x_1 & f(x_1) \\ 1 & x & 0 \end{vmatrix} \Big/ \begin{vmatrix} 1 & x_0 \\ 1 & x_1 \end{vmatrix} \\
&= \frac{x - x_1}{x_0 - x_1} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1) \\
&= f(x_0) + (x - x_0)[x_0, x_1; f].
\end{aligned}$$

□

Let x_i, x_j be distinct points, $i, j \in \{0, \ldots, n\}$. Let

$$L_k := \mathbf{L}(x_0, \ldots, x_{k-1}, x_{k+1}, \ldots, x_n; f), \qquad k = 0, \ldots, n.$$

Consider the polynomial

$$Q = \frac{(X - x_i)L_i - (X - x_j)L_j}{x_j - x_i}.$$

Note that:

$$Q(x_i) = -\frac{x_i - x_j}{x_j - x_i} L_j(x_i) = f(x_i),$$

$$Q(x_j) = \frac{x_j - x_i}{x_j - x_i} L_i(x_j) = f(x_j),$$

$$Q(x_k) = \frac{(x_k - x_i)f(x_k) - (x_k - x_j)f(x_k)}{x_j - x_i} = f(x_k),$$

$k \in \{0, \ldots, n\} \setminus \{i, j\}$, that is,

$$Q(x_k) = f(x_k), \qquad k = 0, \ldots, n,$$

and grad $(Q) \leq n$. Hence,

$$Q = \mathbf{L}(x_0, \ldots, x_n; f).$$

Consequently we obtain the *Aitken-Neville formula:*

$$\begin{aligned} \mathbf{L}(x_0, \ldots, x_n; f) = \quad & \frac{X - x_i}{x_j - x_i} \mathbf{L}(x_0, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n; f) \\ & - \frac{X - x_j}{x_j - x_i} \mathbf{L}(x_0, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n; f) \\ & (i, j \in \{0, \ldots, n\}, \quad i \neq j). \end{aligned} \tag{3.2.5}$$

□

Consider the notations:

$$Q(i, j) = \mathbf{L}(x_{i-j}, \ldots, x_i; f)(x), \qquad 0 \leq j \leq i \leq n.$$

Using formula (3.2.5) we obtain the following algorithm

$$\begin{gathered} Q(i, j) = \frac{(x - x_{i-j})Q(i, j-1) - (x - x_i)Q(i-1, j-1)}{x_i - x_{i-j}} \\ (i = 1, \ldots, n; \quad j = 1, \ldots, i). \end{gathered} \tag{3.2.6}$$

used to generate recursively the Lagrange Polynomials.

For example, with $n = 3$, the polynomials can be obtained by proceeding as shown in the table below, where each row is completed before the succeeding rows are begun.

$$\begin{array}{llllll} Q(0,0) = f(x_0) & & & & & \\ & \downarrow & & & & \\ Q(1,0) = f(x_1) & Q(1,1) & & & & \\ & \downarrow & & & & \\ Q(2,0) = f(x_2) & Q(2,1) & \rightarrow & Q(2,2) & & \\ & & \swarrow & & & \\ Q(3,0) = f(x_3) & Q(3,1) & \rightarrow & Q(3,2) & \rightarrow & Q(3,3) \end{array}$$

□

Using the notations

$$\mathbf{L}(i,j) = \mathbf{L}(x_i, \ldots, x_j; f)(x) \qquad (0 \le i \le j \le n)$$

from (3.2.5), we obtain the *Neville's algorithm*

$$\mathbf{L}(i-m,i) = \frac{(x - x_{i-m})\mathbf{L}(i-m+1,i) - (x - x_i)\mathbf{L}(i-m,i-1)}{x_i - x_{i-m}} \qquad (3.2.7)$$
$$(m = 1, \ldots, n; \quad i = n, n-1, \ldots, m).$$

The various L's form a "tableau" with "ancestors" on the left leading to a single "descendent" at the extreme right. The *Neville's algorithm* is a recursive way of filling in the numbers in the "tableau", a column at the time, from left to right.

For example, with $n = 3$, we have

$$\begin{array}{llll}
\mathbf{L}(0,0) = f(x_0) & & & \\
\mathbf{L}(1,1) = f(x_1) & \mathbf{L}(0,1) & & \\
\mathbf{L}(2,2) = f(x_2) & \mathbf{L}(1,2) & \mathbf{L}(0,2) & \\
\mathbf{L}(3,3) = f(x_3) & \mathbf{L}(2,3) & \mathbf{L}(1,3) & \mathbf{L}(0,3)
\end{array}$$

EXAMPLE ML.3.2.2 MATHEMATICA

```
(* Neville Method 1*)
n:=3; Clear[x, f, L];
Array[x, n, 0];
Array[L, {n, n}, 0];
f[t_]:=t^3;
For [i = 0, i ≤ n, i++, x[i] = i/n; L[i, i] = f[x[i]]] ;
For[j = 1, j ≤ n, j++,
For[i = n, i ≥ j, i--,
L[i - j, i] =
((x - x[i - j])L[i - j + 1, i] - (x - x[i])L[i - j, i - 1])/
(x[i] - x[i - j])]
]
Simplify[TableForm[Table[L[i, j], {i, 0, n}, {j, 0, n}]]]
```

0	$\frac{x}{9}$	$-\frac{2x}{9}+x^2$	x^3
$L[1,0]$	$\frac{1}{27}$	$-\frac{2}{9}+\frac{7x}{9}$	$\frac{2}{9}-\frac{11x}{9}+2x^2$
$L[2,0]$	$L[2,1]$	$\frac{8}{27}$	$-\frac{10}{9}+\frac{19x}{9}$
$L[3,0]$	$L[3,1]$	$L[3,2]$	1

```
Clear[L]; step = 1;
For[j = 1, j ≤ n, j++,
For[i = n, i ≥ j, i--,
L[i - j, i] = "step"(N[step]); step = step + 1]
];
Print[
"=========================================="]
Print["The way of filling in the numbers
in the table"]
Print["=========================================="]
Do[{Print[Simplify[L[k, 0]], " * ", Simplify[L[k, 1]],
" * ", Simplify[L[k, 2]], " * ", Simplify[L[k, 3]]]},
{k, 0, n}];
Print["=========================================="];
```

```
==========================================
The way of filling in the numbers
in the table
==========================================
L[0, 0] * 3.step * 5.step * 6.step
L[1, 0] * L[1, 1] * 2.step * 4.step
L[2, 0] * L[2, 1] * L[2, 2] * 1.step
L[3, 0] * L[3, 1] * L[3, 2] * L[3, 3]
==========================================
```

EXAMPLE ML.3.2.3 MATHEMATICA

```
(* Aitken Method *)
<< Graphics`Arrow`
n:=3; Clear[x, f, Q, points]; points = {};Array[x, n, 0]; Array[Q, {n, n}, 0];
f[t_]:=t^3;
For [i = 0, i ≤ n, i++, x[i] = i/n; Q[i, 0] = f[x[i]]] ;
For[i = 1, i ≤ n, i++,
For[j = 1, j ≤ i, j++,
AppendTo[points, {j, -i}];
Q[i, j] = ((x-x[i-j])Q[i,j-1]-(x-x[i])Q[i-1,j-1])/(x[i]-x[i-j]) ;]
Simplify[TableForm[Table[Q[i, j], {i, 0, n}, {j, 0, i}]]]
```

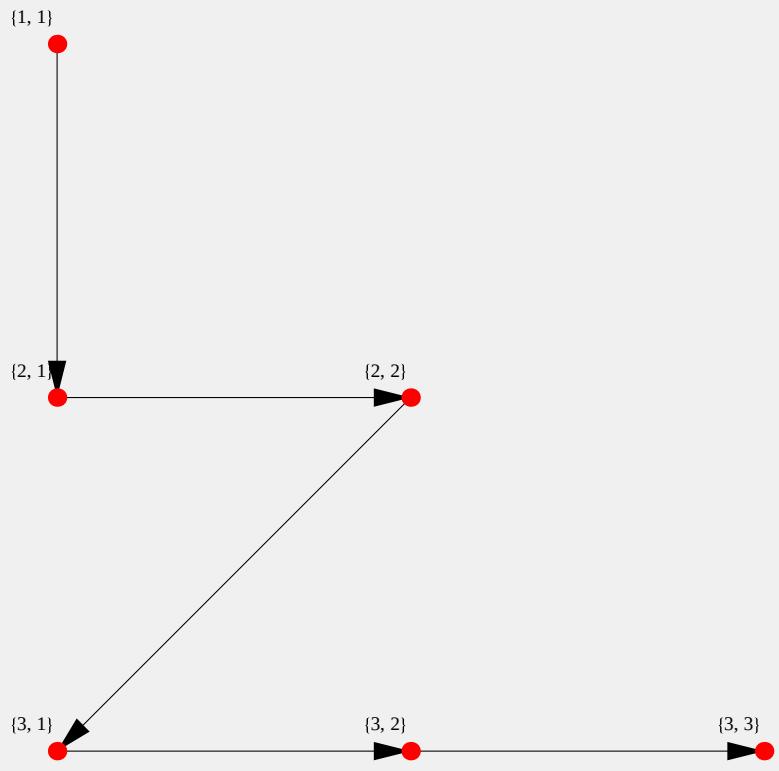
$$\begin{array}{lll} 0 & & \\ \frac{1}{27} & \frac{x}{9} & \\ \frac{8}{27} & -\frac{2}{9}+\frac{7x}{9} & -\frac{2x}{9}+x^2 \\ 1 & -\frac{10}{9}+\frac{19x}{9} & \frac{2}{9}-\frac{11x}{9}+2x^2 \quad x^3 \end{array}$$

```
Print["Lagrange(x) = ", Simplify[Q[n, n]]]
```

Lagrange(x) = x^3

```
txt = Table[Text[{-points[[i]][[2]], points[[i]][[1]]}, points[[i]], {1.2, -2}],
{i, 1, n(n+1)/2}];

arrows = Table [Arrow[points[[i - 1]], points[[i]]], {i, 2, n(n+1)/2}] ;
Show[Graphics[txt], Graphics[arrows], Graphics[{PointSize[.025],
RGBColor[1, 0, 0], Point/@points}],
AspectRatio → Automatic, PlotRange → All]
```



ML.3.3 Some Properties of the Divided Difference

Some forms of the divided difference can be obtained from the formulas which were derived in the previous section.

From (3.2.1) we deduce

$$[x_0,\dots,x_n;f]=\frac{\begin{vmatrix} 1 & x_0 & \cdots & x_0^{n-1} & f(x_0)\\ 1 & x_1 & \cdots & x_1^{n-1} & f(x_1)\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ 1 & x_n & \cdots & x_n^{n-1} & f(x_n)\end{vmatrix}}{\begin{vmatrix} 1 & x_0 & \cdots & x_0^{n-1} & x_0^{n}\\ 1 & x_1 & \cdots & x_1^{n-1} & x_1^{n}\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ 1 & x_n & \cdots & x_n^{n-1} & x_n^{n}\end{vmatrix}}. \tag{3.3.1}$$

Eq. (3.3.1) implies

$$[x_0,\dots,x_n;\ x^i]=\begin{cases} 0, & i=0,\dots,n-1;\\ 1, & i=n.\end{cases} \tag{3.3.2}$$

From (3.2.2) we obtain

$$\begin{aligned}&[x_0,\dots,x_n;f]\\ &=\sum_{i=0}^{n}\frac{f(x_i)}{(x_i-x_0)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_n)}.\end{aligned} \tag{3.3.3}$$

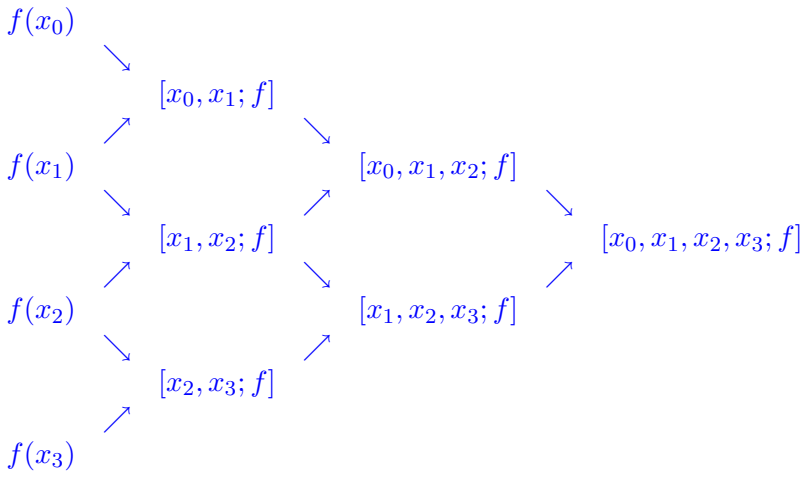
From here we easily get

$$[x_0,\dots,x_i,y_0,\dots,y_j;\ (t-x_0)\dots(t-x_i)\,f(t)]_t=[y_0,\dots,y_j;\ f]. \tag{3.3.4}$$

By identifying the coefficients at x^n in (3.2.5), we obtain the following recurrence formula for divided differences

$$[x_0,\dots,x_n;f]=\frac{[x_1,\dots,x_n;f]-[x_0,\dots,x_{n-1};f]}{x_n-x_0}. \tag{3.3.5}$$

For instance, with $n=3$, the determination of the divided difference from tabulated data points is outlined in the table below:



□

By applying the functional $[x_0, \ldots, x_n; \cdot]$ to the identity

$$(t - x_i)\, f(t) = \frac{x_i - x_0}{x_n - x_0}\,(t - x_n)\, f(t) + \frac{x_n - x_i}{x_n - x_0}\,(t - x_0)\, f(t),$$

$t \in [t_0, t_n]$, $i = 0, \ldots, n$, using (3.3.4), we obtain another recurrence formula

$$\begin{aligned} &[x_0, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n;\ f] \\ &= \frac{x_i - x_0}{x_n - x_0}[x_0, \ldots, x_{n-1};\ f] + \frac{x_n - x_i}{x_n - x_0}[x_1, \ldots, x_n;\ f], \end{aligned} \tag{3.3.6}$$

$i = 0, \ldots, n$.

The following is the Leibniz formula for divided differences

$$[x_0, \ldots, x_n;\ fg] = \sum_{k=0}^{n} [x_0, \ldots, x_k;\ f] \cdot [x_k, \ldots, x_n;\ g]. \tag{3.3.7}$$

We give bellow a simple proof of this formula (Ivan, 1998)[Iva98a]:

$$\begin{aligned} & [x_0, \ldots, x_n;\ fg] \\ = \quad & [x_0, \ldots, x_n; \mathbf{L}(x_0, \ldots, x_n;\ f) \cdot g] \\ = \quad & [x_0, \ldots, x_n; \sum_{k=0}^{n} (t - x_0) \ldots (t - x_{k-1})[x_0, \ldots, x_k;\ f] \cdot g(t)]_t \\ = \quad & \sum_{k=0}^{n} [x_0, \ldots, x_k;\ f] \cdot [x_0, \ldots, x_n;\ (t - x_0) \ldots (t - x_{k-1})\, g(t)]_t \\ \overset{(3.3.4)}{=} \quad & \sum_{k=0}^{n} [x_0, \ldots, x_k;\ f] \cdot [x_k, \ldots, x_n;\ g]. \end{aligned}$$

□

Another useful formula is the integral representation

$$[x_0, \dots, x_k;\, f] = \int_0^1 dt_1 \int_0^{t_1} dt_2 \cdots \int_0^{t_{n-1}} f^{(n)}(x_0 + (x_1 - x_0)t_1 + \cdots + (x_n - x_{n-1})t_n)\, dt_n \tag{3.3.8}$$

which is valid if $f^{(n-1)}$ is absolutely continuous. This can be proved by mathematical induction on n.

□

If $A : C[a,b] \to \mathbb{R}$ is a continuous linear functional which is orthogonal to all polynomials P of degree $\leq n-1$, i.e., $A(P) = 0$, then the following Peano-type representation is valid

$$A(f) = \int_a^b f^{(n)}(t)\, A\left(\frac{(\cdot - t)_+^{n-1}}{(n-1)!}\right) dt, \tag{3.3.9}$$

for all $f \in C^n[a,b]$, where $u_+ := 0$ if $u < 0$ and $u_+ := u$ if $u \geq 0$. This can be proved by applying the functional A to both sides of the Taylor formula

$$f(x) = f(a) + \frac{x-a}{1!} f'(a) + \cdots + \frac{(x-a)^{n-1}}{(n-1)!} f^{(n-1)}(a)$$

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

The functional $[x_0, \dots, x_n;\, \cdot]$ is continuous on $C[a,b]$. Hence it has a Peano representation

$$[x_0, \dots, x_n;\, f] = \frac{1}{n!} \int_a^b n \cdot [x_0, \dots, x_n;\, (\cdot - t)_+^{n-1}]\, f^{(n)}(t)\, dt, \tag{3.3.10}$$

$k = 1, \dots, n$. We would like to emphasize that the Peano kernel $n \cdot [x_0, \dots, x_n;\, (\cdot - t)_+^{n-1}]$, a B-spline function, is non-negative (see Remark (ML.3.11.1)).

□

If the points $x_0, \dots, x_n$ are inside the simple closed curve C and f is analytic on and inside C, then

$$[x_0, \dots, x_n;\, f] = \frac{1}{2\pi i} \int_C \frac{f(z)\, dz}{(z - x_0) \dots (z - x_n)}. \tag{3.3.11}$$

ML.3.4 Mean Value Properties in Lagrange Interpolation

Let $x_0, \dots, x_n$ be a set of distinct points in the interval $[a,b]$. The following mean value theorem gives a bound for the error arising in the Lagrange interpolating-approximating process.

THEOREM ML.3.4.1

(Lagrange Error Formula) If the function $f : [a, b] \to \mathbb{R}$ is continuous and possesses an $(n+1)^{th}$ derivative on the interval (a, b), then for each $x \in [a, b]$ there exists a point $\xi(x) \in (a, b)$, such that

$$f(x) - \mathbf{L}(x_0, \ldots, x_n; f)(x) = (x - x_0) \ldots (x - x_n) \frac{f^{(n+1)}(\xi(x))}{(n+1)!}.$$

Proof. If $x \in \{x_0, \ldots, x_n\}$ then the formula is satisfied for all $\xi \in (a, b)$. Assume that $x \in [a, b] \setminus \{x_0, \ldots, x_n\}$. Define an auxiliary function $H : [a, b] \to \mathbb{R}$, such that

$$H(t) = (f(t) - \mathbf{L}(x_0, \ldots, x_n; f)(t))(x - x_0) \ldots (x - x_n)$$
$$- (f(x) - \mathbf{L}(x_0, \ldots, x_n; f)(x))(t - x_0) \ldots (t - x_n).$$

Note that the function H has $n + 2$ roots, namely $x, x_0, \ldots, x_n$. By the Generalized Rolle Theorem there exists a point $\xi(x) \in (a, b)$ with

$$H^{(n+1)}(\xi(x)) = 0,$$

i.e.,

$$f(x) - \mathbf{L}(x_0, \ldots, x_n; f)(x) = (x - x_0)(x - x_1) \ldots (x - x_n) \frac{f^{(n+1)}(\xi(x))}{(n+1)!}.$$

□

The error formula given in Theorem (ML.3.4.1) is an important theoretical result because Lagrange polynomials are extremely used for deriving numerical differentiation and integration methods.

□

The well known Lagrange's Mean Value Theorem states that if $f \in C[x_0, x_1]$ and f' exists on (x_0, x_1), then $[x_0, x_1; f] = f'(\xi)$ for some $\xi \in (x_0, x_1)$. This result can be generalized by the following theorem.

THEOREM ML.3.4.2

If the function $f : [a, b] \to \mathbb{R}$ is continuous and has an n^{th} derivative on (a, b), then there exists a point $\xi \in (a, b)$ such that

$$[x_0, \ldots, x_n; f] = \frac{f^{(n)}(\xi)}{n!}.$$

Proof. Define an auxiliary function $F : [a, b] \to \mathbb{R}$, such that

$$F(x) = \begin{vmatrix} 1 & x_0 & \cdots & x_0^n & f(x_0) \\ 1 & x_1 & \cdots & x_1^n & f(x_1) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & x_n & \cdots & x_n^n & f(x_n) \\ 1 & x & \cdots & x^n & f(x) \end{vmatrix}.$$

Note that the function F has the roots $x_0, \ldots, x_n$. By the Generalized Rolle Theorem, there exists a point $\xi \in (a, b)$ with

$$F^{(n)}(\xi) = 0,$$

i.e.,

$$\begin{vmatrix} 1 & x_0 & \cdots & x_0^n & f(x_0) \\ 1 & x_1 & \cdots & x_1^n & f(x_1) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & x_n & \cdots & x_n^n & f(x_n) \\ 0 & 0 & \cdots & n! & f^{(n)}(\xi) \end{vmatrix} = 0.$$

By expanding the determinant in terms of the last row we obtain

$$f^{(n)}(\xi)V(x_0, \ldots, x_n) - n! \begin{vmatrix} 1 & x_0 & \cdots & x_0^{n-1} & f(x_0) \\ 1 & x_1 & \cdots & x_1^{n-1} & f(x_1) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & x_n & \cdots & x_n^{n-1} & f(x_n) \end{vmatrix} = 0$$

and the theorem is proved.

ML.3.5 Approximation by Interpolation

Let $f \in C^\infty[a, b]$ be a function possessing uniform bounded derivatives.

THEOREM ML.3.5.1

If $(x_n)_{n\geq 0}$ is a sequence of distinct points in the interval $[a, b]$, then the sequence of Lagrange Polynomials $(\mathbf{L}(x_0, \ldots, x_n; f))_{n\geq 0}$ converges uniformly to f on $[a, b]$.

Proof. Since f has uniform bounded derivatives, then there exists a constant $M > 0$ such that $|f^{(n)}(x)| \leq M$, for all $x \in [a, b]$ and $n \in \mathbb{N}$. Let $x \in [a, b]$. By Theorem (ML.3.4.1) there exists a point $\xi \in (a, b)$, such that

$$\begin{aligned} |f(x) - \mathbf{L}(x_0, \ldots, x_n; f)(x)| &= |(x - x_0) \ldots (x - x_n)| \frac{|f^{(n+1)}(\xi)|}{(n+1)!} \\ &\leq \frac{(b-a)^{n+1}}{(n+1)!} M. \end{aligned}$$

By using

$$\lim_{n\to\infty} \frac{(b-a)^n}{n!} = 0,$$

it follows that the sequence $(\mathbf{L}(x_0, \ldots, x_n; f))_{n\in\mathbb{N}}$ converges uniformly to f on $[a, b]$.

ML.3.6 Hermite-Lagrange Interpolating Polynomial

Let $m, n \in \mathbb{N}$, and $\alpha_0, \ldots, \alpha_n \in \mathbb{N}^*$ such that

$$\alpha_0 + \cdots + \alpha_n = m + 1.$$

Consider the numbers y_i^j, $i = 0, \ldots, n$; $j = 0, \ldots, \alpha_i - 1$, and the distinct points $x_0, \ldots, x_n$. Then there exists a unique polynomial H_m of degree at most m, called the *Hermite-Lagrange interpolating polynomial,* such that

$$H_m^{(j)}(x_i) = y_i^j,$$

for $i = 0, \ldots, n; \quad j = 0, \ldots, \alpha_i - 1$. Define

$$\ell(x) := (x - x_0)^{\alpha_0} \ldots (x - x_n)^{\alpha_n}.$$

The Hermite-Lagrange interpolating polynomial can be written in the form

$$H_m(x) = \sum_{i=0}^{n} \sum_{j=0}^{\alpha_i - 1} \sum_{k=0}^{\alpha_i - j - 1} y_i^j \frac{1}{k!j!} \left(\frac{(x - x_i)^{\alpha_i}}{\ell(x)} \right)^{(k)}_{x = x_i} \frac{\ell(x)}{(x - x_i)^{\alpha_i - j - k}}. \tag{3.6.1}$$

□

Let l_j denotes the j^{th} Lagrange fundamental polynomial,

$$l_j(x) = \prod_{\substack{i=1 \\ i \neq j}}^{n} \frac{x - x_i}{x_j - x_i}.$$

THEOREM ML.3.6.1

If $f \in C^1[a, b]$, then the unique polynomial of least degree agreeing with f and f' at $x_0, \ldots, x_n$ is the polynomial of degree at most $2n + 1$ given by

$$H_{2n+1}(x) = \sum_{j=0}^{n} f(x_j) h_j(x) + \sum_{j=0}^{n} f'(x_j) g_j(x),$$

where

$$\begin{aligned} h_j(x) &= 1 - 2(x - x_j)\, l_j'(x_j)\, l_j^2(x), \\ g_j(x) &= (x - x_j)\, l_j^2(x). \end{aligned}$$

Proof. Since $l_j(x_i) = \delta_{ij}, \quad 0 \leq i, j \leq n$, one can easily show that

$$\begin{aligned} h_j(x_j) &= \delta_{ij}, \\ g_j(x_i) &= 0, \quad 0 \leq i, j \leq n \end{aligned}$$

and

$$\begin{aligned} h_j'(x_i) &= 0, \\ g_j'(x_i) &= \delta_{ij}, \quad 0 \leq i, j \leq n \end{aligned}$$

Consequently

$$\begin{aligned} H_{2n+1}(x_i) &= f(x_i), \\ H_{2n+1}'(x_i) &= f'(x_i), \quad i = 0, \ldots, n. \end{aligned}$$

□

REMARK ML.3.6.2

The divided difference for coalescing points

$$[\underbrace{x_0,\ldots,x_0}_{\alpha_0-\text{times}},\ldots,\underbrace{x_n,\ldots,x_n}_{\alpha_n-\text{times}};f]$$

is defined by Tiberiu Popoviciu [Pop59] to be the coefficient at x^m of the Hermite-Lagrange polynomial satisfying

$$H_m^{(j)}(x_i)=f^{(j)}(x_i),$$

for $i=0,\ldots,n;\ \ j=0,\ldots,\alpha_i-1.$

EXAMPLE ML.3.6.3 MATHEMATICA

```
(* Hermite - Lagrange *)
(* InterpolatingPolynomial[data, var]
gives a polynomial in the variable var which
provides an exact fit to a list of data. *)
(* The number of knots = n + 1 *)
n = 0; Print["Number of knots = ", n + 1]
Number of knots = 1
(* Multiplicity of knots *)
α0 = 2; (*α1 = 1; *)
Table [αi, {i, 0, n}]
{2}
(*The degree of the Hermite − Lagrange polynomial = m *)
m = ∑_{j=0}^{n} αj − 1
1
data =
Table[
{xi, Table [ D[f[t], {t, j}], {j, 0, αi − 1}] /. {t → xi}} , {i, 0, n}]
{{x0, {f [x0] , f′ [x0]}}}
%//TableForm
 x0   f [x0]
      f′ [x0]
H[x_] = InterpolatingPolynomial[data, x]//FullSimplify
f [x0] + (x − x0) f′ [x0]
(*The divided difference on multiple knots [x0, ..., xn; f] *)
Coefficient [H[x], x^m] //Simplify
f′ [x0]
```

ML.3.7 Finite Differences

Let $\mathcal{F} = \{f \mid f : \mathbb{R} \to \mathbb{R}\}$ and $h > 0$. Define the following operators:

- the *identity operator,* $\mathbf{1} : \mathcal{F} \to \mathcal{F}$,
$$\mathbf{1}\, f(x) = f(x);$$
- the *translation, displacement,* or *shift operator,* $\mathbf{T} : \mathcal{F} \to \mathcal{F}$,
$$\mathbf{T} f(x) = f(x + h),$$
introduced by George Boole [Boo60];
- the *backward difference operator,* $\nabla : \mathcal{F} \to \mathcal{F}$,
$$\nabla f(x) = f(x) - f(x - h);$$
- the *forward difference operator,* $\mathbf{\Delta} : \mathcal{F} \to \mathcal{F}$,
$$\mathbf{\Delta} f(x) = f(x + h) - f(x);$$
- the *centered difference operator,* $\delta : \mathcal{F} \to \mathcal{F}$,
$$\delta f(x) = f\left(x + \frac{h}{2}\right) - f\left(x - \frac{h}{2}\right).$$

Some properties of the forward difference operator $\mathbf{\Delta}$ will be presented. Since
$$\mathbf{\Delta} = \mathbf{T} - \mathbf{1},$$
the following relation can be derived
$$\mathbf{\Delta}^n = (\mathbf{T} - \mathbf{1})^n,$$
and hence
$$\mathbf{\Delta}^n f(x) = \sum_{k=0}^{n} (-1)^k \binom{n}{k} \mathbf{T}^{n-k} f(x).$$
Moreover, using the relation $\mathbf{T}^i f(x) = f(x + ih)$, we obtain
$$\mathbf{\Delta}^n f(x) = \sum_{k=0}^{n} (-1)^k \binom{n}{k} f(x + (n - k)h). \tag{3.7.1}$$
We also have
$$\sum_{k=0}^{n} \binom{n}{k} \mathbf{\Delta}^k f(x) = (\mathbf{1} + \mathbf{\Delta})^n f(x) = \mathbf{T}^n f(x) = f(x + nh). \tag{3.7.2}$$

EXAMPLE ML.3.7.1

For $h = 1$, since $\Delta^k \dfrac{1}{x} = \dfrac{(-1)^k k!}{x(x+1)\dots(x+k)} = \dfrac{(-1)^k k!}{x^{\overline{k+1}}}$, by (3.7.1), we obtain

$$\sum_{k=0}^{n} \frac{(-1)^k n^{\underline{k}}}{x(x+1)\dots(x+k)} = \frac{1}{x+n}. \tag{3.7.3}$$

□

By denoting $y_i = f(x+ih)$, $i = 0, \dots, n$, the forward finite differences can be calculated as in the following table:

y_0	Δy_0	$\Delta^2 y_0$	$\Delta^3 y_0$	$\cdots$	$\Delta^{n-1} y_0$	$\Delta^n y_0$
y_1	Δy_1	$\Delta^2 y_1$	$\Delta^3 y_1$	$\cdots$	$\Delta^{n-1} y_1$	
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$		
y_{n-3}	Δy_{n-3}	$\Delta^2 y_{n-3}$	$\Delta^3 y_{n-3}$			
y_{n-2}	Δy_{n-2}	$\Delta^2 y_{n-2}$				
y_{n-1}	Δy_{n-1}					
y_n						

□

It should be noted that the finite difference is a particular case of the divided difference. In fact, with $x_k = x_0 + k\,h$,

$$\begin{aligned}
& [x, x+h, \dots, x+nh; f] \\
= & \sum_{k=0}^{n} \frac{f(x_k)}{(x_k - x_0)\dots(x_k - x_{k-1})(x_k - x_{k+1})\dots(x_k - x_n)} \\
= & \sum_{k=0}^{n} \frac{(-1)^{n-k} f(x+kh)}{k!(n-k)!h^n} \\
= & \frac{1}{n!h^n} \sum_{k=0}^{n} (-1)^{n-k} \binom{n}{k} f(x+kh) \\
= & \frac{\Delta^n f(x)}{n!h^n},
\end{aligned}$$

hence

$$[x, x+h, \dots, x+nh; f] = \frac{\Delta^n f(x)}{n!h^n}. \tag{3.7.4}$$

□

The relationship between the derivative of a function and the forward finite difference is presented in the following theorem.

THEOREM ML.3.7.2

If the function $f : \mathbb{R} \to \mathbb{R}$ has an n^{th} derivative, then for each $x \in \mathbb{R}$ there exists $\xi \in (x, x + nh)$, such that

$$\Delta^n f(x) = h^n f^{(n)}(\xi).$$

This is a direct consequence of (ML.3.4.2) and (3.7.4).

□

Let $\mathcal{C}$ be the set of all functions possessing a Taylor expansion at all points $x \in \mathbb{R}$. In what follows, Δ is taken as the restriction of the forward finite difference operator to the set $\mathcal{C}$. Consider the differentiating operator $D : \mathcal{C} \to \mathcal{C}$,

$$Df = f'.$$

From

$$\begin{aligned} f(x+h) - f(x) &= \frac{h}{1!} f'(x) + \frac{h^2}{2!} f''(x) + \cdots + \frac{h^n}{n!} f^{(n)}(x) + \cdots \\ &= \left(\frac{hD}{1!} + \frac{(hD)^2}{2!} + \cdots + \frac{(hD)^n}{n!} + \cdots \right) f(x), \end{aligned}$$

we deduce

$$\Delta = e^{hD} - 1. \tag{3.7.5}$$

Likewise, we get:

$$\nabla = 1 - e^{-hD},$$

$$\delta = 2 \operatorname{sh} \frac{hD}{2}.$$

In order to represent the operator D by the operator Δ, we use the following result

$$e^{hD} = 1 + \Delta,$$

to derive

$$hD = \ln(1 + \Delta) = \Delta - \frac{\Delta^2}{2} + \frac{\Delta^3}{3} - \frac{\Delta^4}{4} + \cdots$$

Furthermore, we obtain:

$$h^2 D^2 = \Delta^2 - \Delta^3 + \frac{11}{12} \Delta^4 - \frac{5}{6} \Delta^5 + \cdots$$

$$h^3 D^3 = \Delta^3 - \frac{3}{2} \Delta^4 + \frac{7}{4} \Delta^5 - \cdots$$

□

Taylor's Formula can be written in the form

$$f(a + h) = e^{hD} f(a).$$

Therefore

$$f(a+xh) = e^{xhD}f(a) = (e^{hD})^x f(a) = (1+\boldsymbol{\Delta})^x f(a),$$

which known as the

Gregory-Newton Interpolating Formula,

$$f(a+xh) = (1 + x\boldsymbol{\Delta} + \frac{x(x-1)}{2!}\boldsymbol{\Delta}^2 + \frac{x(x-1)(x-2)}{3!}\boldsymbol{\Delta}^3 + \cdots)f(a),$$

i.e.,

$$f(a+xh) = \sum_{n=0}^{\infty} \binom{x}{n} \boldsymbol{\Delta}^n f(a), \tag{3.7.6}$$

$x \in \mathbb{R}$.

From (3.7.6) it follows that, for any polynomial P of degree at most n, we have

$$P(x+a) = \sum_{k=0}^{n} \binom{x}{k} \boldsymbol{\Delta}^k P(a),$$

hence, for all $x \in \mathbb{R}$ and $m \in \mathbb{Z}$, we obtain

$$P(x+m) = \sum_{k=0}^{n} \sum_{i=0}^{k} \binom{x}{k} \binom{k}{i} (-1)^{k-i} P(m+i). \tag{3.7.7}$$

PROBLEM ML.3.7.3

Let P be a polynomial of degree at most n with real coefficients. Suppose that P takes integer values at some $n+1$ consecutive integers. Prove that P takes integer values at any integer number.

We simply use representation (3.7.7) for P.

ML.3.8 Interpolation of Functions of Several Variables

Let $a \le x_0 < \cdots < x_n \le b$ and $c \le y_0 < \cdots < y_m \le d$. Define

$$\mathcal{F} = \{f \mid f : [a,b] \times [c,d] \to \mathbb{R}\},$$

and

$$u_i : [a,b] \to \mathbb{R}, \quad v_j : [c,d] \to \mathbb{R}$$

be functions satisfying the conditions:

$$u_i(x_k) = \delta_{ik}, \qquad i,k = 0,\ldots,n;$$

$$v_j(y_h) = \delta_{jh}, \qquad j,h = 0,\ldots,m.$$

Define the interpolating operators $U, V : \mathcal{F} \to \mathcal{F}$, by

$$Uf(x,y) = \sum_{i=0}^{n} f(x_i, y) u_i(x),$$

$$Vf(x,y) = \sum_{j=0}^{m} f(x, y_j) v_j(y).$$

Note that:

$$Uf(x_k, y) = f(x_k, y) \quad \text{and} \quad Vf(x, y_h) = f(x, y_h),$$

$k = 0, \ldots, n$; $h = 0, \ldots, m$; $x \in [a, b]$, $y \in [c, d]$.

DEFINITION ML.3.8.1

The operator UV is called the *tensor product* of the operators U and V.

Note that we have

$$UVf(x,y) = \sum_{i=0}^{n} \sum_{j=0}^{m} f(x_i, y_j)\, u_i(x)\, v_j(y)$$

and

$$UVf(x_k, y_h) = f(x_k, y_h),$$

$k = 0, \ldots, n$; $h = 0, \ldots, m$.

DEFINITION ML.3.8.2

The operator $U \oplus V := U + V - UV$ is called the *Boolean sum* of the operators U and V.

Note that:

$$U \oplus Vf(x_k, y) = f(x_k, y), \qquad U \oplus Vf(x, y_h) = f(x, y_h),$$

$k = 0, \ldots, n$; $h = 0, \ldots, m$; $x \in [a, b]$, $y \in [c, d]$.

ML.3.9 Scattered Data Interpolation. Shepard's Method

In many fields such as geology, meteorology and cartography, we frequently encounter nonuniformly spaced data, also called *scattered data,* which have to be interpolated.

The interpolation problem can be stated as follows. Suppose that a set of n distinct abscissae $X_i = (x_i, y_i) \in \mathbb{R}^2$, and the associated ordinates z_i, $i = 1, \ldots, n$, are given. Then we seek a function $f : \mathbb{R}^2 \to \mathbb{R}$ such that $f(X_i) = z_i$, $i = 1, \ldots, n$.

In a *global method,* the interpolant depends on all the data points at once in the sense that adding, changing, or removing one data point implies that we have to solve the problem again.

In a *local method,* adding, changing or removing a data point affects the interpolant only locally, i.e., in a certain subset of the domain.

The *Shepard interpolant* to the data $(X_i; z_i) \in \mathbb{R}^3$, $i = 1, \ldots, n$, is a weighted mean of the ordinates z_i :

$$S(X) := \sum_{i=1}^{n} \omega_i(X)\, z_i, \quad X \in \mathbb{R}^3, \tag{3.9.1}$$

with weight functions

$$\omega_i(X) := \frac{\displaystyle\prod_{\substack{j=1 \\ j\neq i}}^{n} \rho^{\mu_i}(X, X_j)}{\displaystyle\sum_{k=1}^{n} \prod_{\substack{j=1 \\ j\neq k}}^{n} \rho^{\mu_i}(X, X_j)}$$

where ρ is a metric on $\mathbb{R}^2$ and $\mu_i > 0$. For $X \neq X_i, i = 1, \ldots, n$, we can write

$$\omega_i(X) = \frac{\dfrac{1}{\rho^{\mu_i}(X, X_i)}}{\displaystyle\sum_{k=1}^{n} \frac{1}{\rho^{\mu_i}(X, X_k)}}, \quad i = 1, \ldots, n.$$

This is a kind of inverse distance weighted method, i.e., the larger the distance of X to X_i, the less influence z_i has on the value of S at the point X.

Now, we assume that ρ is the Euclidean metric on $\mathbb{R}^2$. The functions ω_i have the following properties:

- $\omega_i \in C^0$, continuity;
- $\omega_i(X) \geq 0$, positivity;
- $\omega_i(X_j) = \delta_{ij}$, interpolation property $(S(X_i) = z_i)$;
- $\displaystyle\sum_{i=1}^{n} \omega_i(X) = 1$, normalization.

It can be shown that the Shepard interpolant (3.9.1) has

- cusps at X_i if $0 < \mu_i < 1$,
- corners at X_i if $\mu_i = 1$,
- flat spots at X_i if $\mu_i > 1$, i.e., the tangent planes at such points are parallel to the (x, y) plane.

Besides the obvious weaknesses of Shepard's method, clearly all of the functions $\omega_i(X)$ have to be recomputed as soon as just one data point is changed, or a data point is added or removed. This shows that Shepard's method is a global method.

There are several ways to improve Shepard's method, for example by removing the discontinuities in the derivatives and the flat spots at the interpolation points. Even the global character can be made local using the following procedures:

1. multiply the weight functions ω_i by a mollifying function λ_i :

   $\lambda_i(X_i) = 1$,

   $\lambda_i(X) \geq 0$, inside a given neighborhood B_i of X_i,

   $\lambda_i(X) = 0$, outside of B_i,

2. use weight functions which are locally supported like B-splines,

3. use a recurrence formula. Consider the weight functions

$$e_k(X) := \frac{\prod_{j=1}^{k-1} \rho^{\mu}(X, X_j)}{\sum_{j=1}^{k} \prod_{\substack{i=1 \\ i \neq j}}^{k} \rho^{\mu}(X, X_i)}, \quad k = 2, \ldots n,$$

   and the recurrence formula

$$\begin{array}{rcl} s_1(X) & = & z_1, \\ s_k(X) & = & s_{k-1}(X) + e_k(X)(z_k - s_{k-1}(X_k)), \quad k = 2, \ldots, n. \end{array} \tag{3.9.2}$$

We have obtained a recursive formula for the Shepard-type interpolant s_n.

ML.3.10 Splines

Splines are piecewise polynomials with $\mathbb{R}$ as their natural domain of definition. They often appear as solutions to extremal problems. Splines also appears as kernels of interpolations formulas. The systematic study of splines has begun with the work of Isaac Jacob Schoenberg in the forties.

We would like to emphasize that it is Tiberiu Popoviciu who defined the so called *"elementary functions of order n"*, later referred as *splines*. Results of Popoviciu's work were also known by I. J. Schoenberg (see M. Mardens and I. J. Schoenberg, *On the variation diminishing spline approximation methods,* Mathematica **8**(31),1, 1966, pp. 61–82 (dedicated to Tiberiu Popoviciu on his 60^{th} birthday)).

At present, splines are widely used in numerical computation and are indispensable for many theoretical questions.

A typical spline is the truncated power

$$(x-a)_+^k := \begin{cases} 0, & \text{if } x < a, \\ (x-a)^k, & \text{if } x \geq a, \end{cases}$$

where $k \in \mathbb{N}^*$. With $k = 0$, we obtain the Heaviside function

$$(x-a)_+^0 := \begin{cases} 0, & \text{if } x < a, \\ 1, & \text{if } x \geq a, \end{cases}$$

Let $n, p \in \mathbb{N}^*$ and $(a = x_0 < \cdots < x_n = b)$ be a division of the interval $[a, b]$.

DEFINITION ML.3.10.1

A function $S \in C^{p-2}[a,b]$ is called a *spline of order* p $(p = 2, 3, \ldots)$, equivalently of degree $m = p-1$, with breakpoints x_i, if on each interval $[x_{i-1}, x_i]$ it is a polynomial of degree $\leq m$, and on one of them is of degree exactly m.

Thus, the splines of order two are broken lines.

One can prove that the linear space of splines of order p on the interval $[a,b]$ has the following basis:

$$\{1, x, \ldots, x^{p-1}, (x-x_0)_+^{p-1}, \ldots, (x-x_n)_+^{p-1}\}.$$

ML.3.11 B-splines

The representation of a spline as a sum of a truncated powers is generally not very useful because the truncated power functions have large support.

A basis which is more local in character can be defined by using *B-splines* (*basic splines*) which are splines with the smallest possible support. The B-splines are defined by means of the divided differences

$$B(x) := B(x_0, \ldots, x_n;\, x) := n\,[x_0, \ldots, x_n;\, (\cdot - x)_+^{n-1}]. \tag{3.11.1}$$

Important properties of B-splines can be derived from properties of divided differences. As an example, we derive the following recurrence formula which is valid when $n \geq 2$

$$B(x_0, \ldots, x_n;\, x) \tag{3.11.2}$$

$$= \frac{n}{n-1}\left(\frac{x-x_0}{x_n-x_0}\,B(x_0, \ldots, x_{n-1};\, x) + \frac{x_n-x}{x_n-x_0}\,B(x_1, \ldots, x_n;\, x)\right),$$

$$B(x_i, x_{i+1}; x) = \begin{cases} \dfrac{1}{x_{i+1}-x_i}, & \text{if } x \in [x_i, x_{i+1}) \\ 0, & \text{otherwise} \end{cases}, \quad i = 0, 1, \ldots, n-1.$$

In fact, if x is not a knot, by using (3.3.4) and (3.3.6), we obtain

$$B(x_0, \ldots, x_n;\, x) = n\,[x_0, \ldots, x_n;\, (t-x)_+^{n-1}]_t$$

$$= \frac{n(x-x_0)}{x_n-x_0}[x_0, \ldots, x_{n-1}, x;\, (t-x)(t-x)_+^{n-2}]_t$$

$$+ \frac{n(x_n-x)}{x_n-x_0}[x_1, \ldots, x_n, x;\, (t-x)(t-x)_+^{n-2}]_t$$

$$= \frac{n(x-x_0)}{x_n-x_0}[x_0, \ldots, x_{n-1};\, (t-x)_+^{n-2}]_t + \frac{n(x_n-x)}{x_n-x_0}[x_1, \ldots, x_n;\, (t-x)_+^{n-2}]_t$$

$$= \frac{n}{n-1}\Big(\frac{x-x_0}{x_n-x_0}\,B(x_0,\dots,x_{n-1};\,x) + \frac{x_n-x}{x_n-x_0}\,B(x_1,\dots,x_n;\,x)\Big).$$

If x is a knot (for instance $x = x_0$), by using (3.3.4), we obtain

$$B(x_0,\dots,x_n;\,x_0) = n\,[x_0,\dots,x_n;\,(t-x_0)_+^{n-1}]_t$$

$$= n\,[x_0,\dots,x_n;\,(t-x_0)(t-x_0)_+^{n-2}]_t = n\,[x_1,\dots,x_n;\,(t-x_0)_+^{n-2}]_t.$$

REMARK ML.3.11.1

From Equation (3.11.2) one can easily prove that B-splines are non-negative functions.

It is also easy to differentiate a B-spline. We have, for any x which is not a knot,

$$B'(x_0,\dots,x_n;\,x) = n\,[x_0,\dots,x_n;\,\frac{d}{dx}(t-x)_+^{n-1}]_t$$

$$= n(n-1)[x_0,\dots,x_n;\,(t-x)_+^{n-2}]_t$$

$$= \frac{n(n-1)}{x_n-x_0}\Big([x_1,\dots,x_n;\,(t-x)_+^{n-2}]_t - [x_0,\dots,x_{n-1};\,(t-x)_+^{n-2}]_t\Big)$$

$$\overset{(3.3.5)}{=} \frac{n}{x_n-x_0}\Big(B(x_1,\dots,x_n;\,x) - B(x_0,\dots,x_{n-1};\,x)\Big).$$

Next, we present some properties of the spline functions of degree 3.

Let us consider the points $x_i = a + ih$, $h = (b-a)/n$, $i = -3, -2, \dots, n+3$. We define the functions:

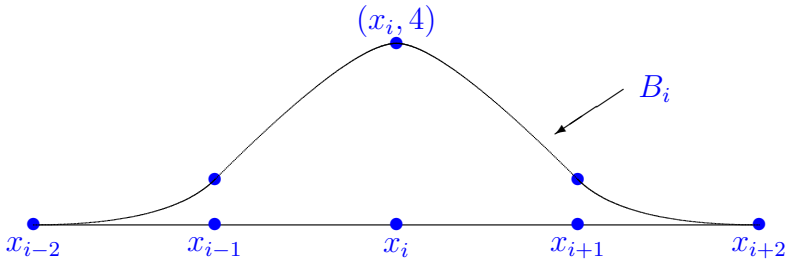
$$B_i(x) :=$$

$$\frac{1}{h^3}\begin{cases} 0, & x \in (-\infty, x_{i-2}] \\ (x-x_{i-2})^3 & x \in (x_{i-2}, x_{i-1}] \\ h^3 + 3h^2(x-x_{i-1}) + 3h(x-x_{i-1})^2 - 3(x-x_{i-1})^3, & x \in (x_{i-1}, x_i] \\ h^3 + 3h^2(x_{i+1}-x) + 3h(x_{i+1}-x)^2 - 3(x_{i+1}-x)^3, & x \in (x_i, x_{i+1}] \\ (x_{i+2}-x)^3, & x \in (x_{i+1}, x_{i+2}] \\ 0, & x \in (x_{i+2}, \infty). \end{cases}$$

One can prove that the functions B_i belong to $C^2[a,b]$, for $i = -1, 0, 1, \dots, n+1$.

Some of their properties can be found in the table below.

x	x_{i-2}	x_{i-1}	x_i	x_{i+1}	x_{i+2}
$B_i(x)$	0	1	4	1	0
$B_i'(x)$	0	$\frac{3}{h}$	0	$-\frac{3}{h}$	0
$B_i''(x)$	0	$\frac{6}{h^2}$	$-\frac{12}{h^2}$	$\frac{6}{h^2}$	0



Let $f \in C^1[a,b]$. We try to find a spline function $S \in C^2[a,b]$ satisfying the following interpolating conditions:

$$\begin{cases} S'(x_0) &= f'(x_0), \\ S(x_i) &= f(x_i), \quad i = 0, \ldots, n, \\ S'(x_n) &= f'(x_n). \end{cases}$$

We try to find such a function S in the form

$$S = a_{-1}B_{-1} + a_0 B_0 + \cdots + a_n B_n + a_{n+1}B_{n+1}.$$

We obtain the system

$$\begin{cases} a_{-1}B'_{-1}(x_0) + a_0 B'_0(x_0) + \cdots + a_{n+1}B'_{n+1}(x_0) &= f'(x_0) \\ a_{-1}B_{-1}(x_i) + a_0 B_0(x_i) + \cdots + a_{n+1}B_{n+1}(x_i) &= f(x_i) \\ & \quad (i = 0, \ldots, n) \\ a_{-1}B'_{-1}(x_n) + a_0 B'_0(x_n) + \cdots + a_{n+1}B'_{n+1}(x_n) &= f'(x_n) \end{cases}$$

The matrix of the system is

$$\begin{bmatrix} -\frac{3}{h} & 0 & \frac{3}{h} & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 & 0 & \cdots & -\frac{3}{h} & 0 & \frac{3}{h} \end{bmatrix}$$

and its determinant is equal to the determinant of the matrix

$$\begin{bmatrix} -\frac{3}{h} & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1 & 4 & 2 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 2 & 4 & 1 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \frac{3}{h} \end{bmatrix}.$$

It is clear that the above matrix is strictly diagonally dominant, hence the system has a unique solution.

We shall write the functions B_i in terms of the divided differences. By direct calculation, we can show that

$$B_i(x) = 4!\, h\, [x_{i-2}, x_{i-1}, x_i, x_{i+1}, x_{i+2};\ (\cdot - x)^3_+],$$

$i = -1, 0, \ldots, n+1$.

ML.3.12 Problems

Let $x_0, \dots, x_n$ be distinct points on the real axis, f be a function defined on these points and $\ell(x) = (x - x_0) \dots (x - x_n)$.

PROBLEM ML.3.12.1

Let P be a polynomial. Prove that the Lagrange Polynomial $\mathbf{L}(x_0, \dots, x_n; P)$ is the remainder obtained by dividing the polynomial P by the polynomial $(X - x_0) \dots (X - x_n)$.

Let $P(x) = Q(x)\,(x - x_0) \dots (x - x_n) + R(x)$. We have $R(x_i) = P(x_i)$ for $i = 0, \dots, n$. Since the degree of the polynomial R is at most n, we get $R = \mathbf{L}(x_0, \dots, x_n; P)$.

PROBLEM ML.3.12.2

Calculate $\mathbf{L}(x_0, \dots, x_n;\, X^{n+1})$.

We have

$$\begin{aligned} X^{n+1} \;=\; & 1 \cdot (X - x_0)(X - x_1) \dots (X - x_n) \\ & + X^{n+1} - (X - x_0)(X - x_1) \dots (X - x_n). \end{aligned}$$

By P. (ML.3.12.1) we obtain

$$\mathbf{L}(x_0, \dots, x_n;\, X^{n+1}) = X^{n+1} - (X - x_0)(X - x_1) \dots (X - x_n).$$

PROBLEM ML.3.12.3

Let $P_n : \mathbb{R} \to \mathbb{R}$ be a polynomial of degree at most n, $n \in \mathbb{N}$. Prove the decomposition in partial fractions

$$\frac{P_n(x)}{(x - x_0) \dots (x - x_n)} = \sum_{k=0}^{n} \frac{1}{x - x_i} \cdot \frac{P(x_i)}{\ell'(x_i)}.$$

With $\ell(x) = (x - x_0) \dots (x - x_n)$, we have

$$\frac{P_n(x)}{\ell(x)} = \frac{\mathbf{L}(x_0, \dots, x_n;\, P_n)(x)}{\ell(x)} = \sum_{k=0}^{n} \frac{1}{x - x_i} \cdot \frac{P(x_i)}{\ell'(x_i)}.$$

PROBLEM ML.3.12.4

Calculate the sum

$$\sum_{i=0}^{n} \frac{x_i^k}{(x_i - x_0) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n)},$$

$k = 0, \dots, n$.

PROBLEM ML.3.12.5

Calculate

$$[x_0, \dots, x_n; x^k],$$

for $k = n, n+1$.

PROBLEM ML.3.12.6

Calculate

$$[x_0, \dots, x_n; (X - x_0) \dots (X - x_k)],$$

$k \in \mathbb{N}$.

PROBLEM ML.3.12.7

Prove that

$$\mathbf{L}(x_0, \dots, x_n; \mathbf{L}(x_1, \dots, x_n; f)) = \mathbf{L}(x_1, \dots, x_n; \mathbf{L}(x_0, \dots, x_n; f)).$$

PROBLEM ML.3.12.8

Prove that, for $x \notin \{x_0, \dots, x_n\}$,

$$\left[x_0, \dots, x_n; \frac{f(t)}{x - t}\right]_t = \frac{\mathbf{L}(x_0, \dots, x_n;\ f)(x)}{(x - x_0) \dots (x - x_n)}.$$

With $\ell(x) = (x - x_0) \dots (x - x_n)$, we have

$$\mathbf{L}(x_0, \dots, x_n;\ f)(x) = \sum_{k=0}^{n} \frac{\ell(x)}{x - x_i} \cdot \frac{f(x_i)}{\ell'(x_i)},$$

hence

$$\frac{\mathbf{L}(x_0, \dots, x_n;\ f)(x)}{\ell(x)} = \sum_{k=0}^{n} \frac{\frac{f(x_i)}{x - x_i}}{\ell'(x_i)} = \left[x_0, \dots, x_n; \frac{f(t)}{x - t}\right]_t.$$

PROBLEM ML.3.12.9

If $f(x) = \frac{1}{x}$ and $0 < x_0 < x_1 < \dots < x_n$, then

$$[x_0, \dots, x_n;\ f] = \frac{(-1)^n}{x_0 \dots x_n}.$$

By using problem (ML.3.12.8), we have

$$\left[x_0, \ldots, x_n; \frac{1}{t}\right]_t = -\frac{\mathbf{L}(x_0, \ldots, x_n; 1)(0)}{(0 - x_0) \ldots (0 - x_n)} = \frac{(-1)^n}{x_0 \ldots x_n}.$$

Solution 2. We write the obvious equality

$$\left[x_0, \ldots, x_n; \frac{(t - t_0) \ldots (t - x_n)}{t}\right] = 0$$

in the form

$$\left[x_0, \ldots, x_n; t^n - (t_0 + \cdots + t_n)t^{n-1} + \cdots + (-1)^{n+1}\frac{t_0 \ldots x_n}{t}\right] = 0,$$

i.e.,

$$1 - (t_0 + \cdots + t_n)0 + \cdots + 0 + (-1)^{n+1} t_0 \ldots x_n \left[x_0, \ldots, x_n; \frac{1}{t}\right] = 0.$$

ML.4

Elements of Numerical Integration

ML.4.1 Richardson's Extrapolation

Extrapolation is applied when it is known that the approximation technique has an error term with a predictable form, one that depends on a parameter, usually the step size h.

Richardson's extrapolation is a technique frequently employed to generate results of high accuracy by using low-order formulas.

Let $1 < p_1 < p_2 < \dots$ be a sequence of integers and F be a formula that approximates an unknown value M. Suppose that, in a neighborhood of the point zero, the following relation holds

$$M = F(h) + a_1 h^{p_1} + a_2 h^{p_2} + \cdots$$

where a_1, a_2, $\dots$ are constants. The order of approximation for $F(h)$ is $O(h^{p_1})$. By denoting

$$F_1(h) := F(h), \quad a_{11} := a_1, \quad a_{12} := a_2, \dots$$

we obtain

$$M = F_1(h) + a_{11} h^{p_1} + a_{12} h^{p_2} + \cdots .$$

For $q > 1$, we deduce

$$M = F_1\left(\frac{h}{q}\right) + a_{11}\left(\frac{h}{q}\right)^{p_1} + a_{12}\left(\frac{h}{q}\right)^{p_2} + \cdots ,$$

hence, by letting

$$F_2(h) = \frac{q^{p_1} F_1\left(\frac{h}{q}\right) - F_1(h)}{q^{p_1} - 1},$$

we obtain

$$M = F_2(h) + 0 \cdot h^{p_1} + a_{22} h^{p_2} + \cdots ,$$

i.e., the formula $F_2(h)$ has order of approximation $O(h^{p_2})$.

Step by step, we deduce approximation formulas of order $O(h^{p_j})$:

$$F_j(h) = \frac{q^{p_{j-1}} F_{j-1}\left(\frac{h}{q}\right) - F_{j-1}(h)}{q^{p_{j-1}} - 1}, \tag{4.1.1}$$

$j = 2, 3, \ldots$.

In the case of the formula

$$M = F(h) + a_1 h + a_2 h^2 + \cdots ,$$

with $q = 2$, we deduce

$$F_1(h) := F(h),$$
$$F_j(h) := \frac{2^{j-1} F_{j-1}\left(\frac{h}{2}\right) - F_{j-1}(h)}{2^{j-1} - 1},$$

$j = 2, 3, \ldots, n$, which are approximations of order $O(h^j)$ for M.

For example, with $n = 3$, the approximations are generated by rows, in the order indicated by the framed entries in the table below.

$O(h)$	$O(h^2)$	$O(h^3)$	$O(h^4)$
$\boxed{1}$ $F_1(h)$			
$\boxed{2}$ $F_1(\frac{h}{2})$	$\boxed{3}$ $F_2(h)$		
$\boxed{4}$ $F_1(\frac{h}{4})$	$\boxed{5}$ $F_2(\frac{h}{2})$	$\boxed{6}$ $F_3(h)$	
$\boxed{7}$ $F_1(\frac{h}{8})$	$\boxed{8}$ $F_2(\frac{h}{4})$	$\boxed{9}$ $F_3(\frac{h}{2})$	$\boxed{10}$ $F_4(h)$

Each column beyond the first in the extrapolation table is obtained by a simple averaging process.

If $F_1(h)$ is an approximation formula of order $O(h^2)$, then we obtain the following approximations of order $O(h^{2j})$ for M :

$$F_j(h) := \frac{4^{j-1} F_{j-1}\left(\frac{h}{2}\right) - F_{j-1}(h)}{4^{j-1} - 1}, \tag{4.1.2}$$

$j = 2, 3, \ldots$.

ML.4.2 Numerical Quadrature

Let $f : [a, b] \to \mathbb{R}$ be an integrable function.

In general, the problem of evaluating the definite integral involves the following difficulties:

- the values of the function f are known only at the given points, say $x_0, \ldots, x_n$;
- the function f has no antiderivatives;
- the antiderivatives of f cannot be determined analytically.

Therefore, the basic method involved in approximating the definite integral of f uses a sum of the form

$$\sum_{i=0}^{n} A_i f(x_i),$$

where $A_0, \ldots, A_n$ are constants not depending on f.

DEFINITION ML.4.2.1

A relation of the form

$$\int_a^b f(x)\,dx = \sum_{i=0}^{n} A_i f(x_i) + R_n(f),$$

is called a *quadrature formula.*

The number $R_n(f)$ is the *remainder* of the quadrature formula (ML.4.2.1). The previous equality is sometimes written in the form

$$\int_a^b f(x)\,dx \approx \sum_{i=0}^{n} A_i f(x_i).$$

DEFINITION ML.4.2.2

The *degree of accuracy,* **or** *degree of precision* **of the quadrature formula (ML.4.2.1) is the positive integer m such that**

$$R_n(X^k) = 0, \quad k = 0, \ldots, m, \quad R_n(X^{m+1}) \neq 0.$$

The quadrature formulas can be classified as follows:

- **Newton-Cotes type,** when the nodes $x_0, \ldots, x_n$ are given and the coefficients $A_0, \ldots, A_n$ are determined such that the degree of precision is at least n;
- **Chebyshev type,** when the coefficients $A_0, \ldots, A_n$ are equal one to the other and the nodes $x_0, \ldots, x_n$ are determined such that the degree of precision is at least n;
- **Gauss type,** when coefficients $A_0, \ldots, A_n$ and the nodes $x_0, \ldots, x_n$, are determined such that the degree of precision is maximal.

ML.4.3 Error Bounds in the Quadrature Methods

Let $f \in C^{n+1}[a, b]$. Consider a quadrature formula

$$\int_a^b f(x)\,dx = \sum_{i=0}^{n} A_i f(x_i) + R_n(f)$$

with order of precision n. Using Taylor's formula with integral form of the remainder

$$f(x) = \sum_{j=0}^{n} \frac{f^{(j)}(a)}{j!}(x-a)^j + \frac{1}{n!}\int_a^x (x-t)^n f^{(n+1)}(t)dt,$$

$x \in [a, b]$, we obtain

$$R_n(f)$$

$$= R_n\left(\sum_{j=0}^{n}\frac{f^{(j)}(a)}{j!}(X-a)^j\right)+\frac{1}{n!}R_n\left(\int_a^X (X-t)^n f^{(n+1)}(t)dt\right)$$

$$= \frac{1}{n!}R_n\left(\int_a^X (X-t)^n f^{(n+1)}(t)dt\right)$$

$$= \int_a^b \frac{1}{n!}\int_a^x (x-t)^n f^{(n+1)}(t)dt\,dx - \frac{1}{n!}\sum_{i=0}^{n} A_i \int_a^{x_i}(x_i-t)^n f^{(n+1)}(t)dt$$

$$= \int_a^b \frac{1}{n!}\int_t^b (x-t)^n f^{(n+1)}(t)dx\,dt - \frac{1}{n!}\sum_{i=0}^{n} A_i \int_a^{b}(x_i-t)_+^n f^{(n+1)}(t)dt$$

$$= \int_a^b \left(\frac{(b-t)^{n+1}}{(n+1)!} - \frac{1}{n!}\sum_{i=0}^{n} A_i(x_i-t)_+^n\right) f^{(n+1)}(t)dt.$$

By denoting

$$K_n = \int_a^b \left|\frac{(b-t)^{n+1}}{(n+1)!} - \frac{1}{n!}\sum_{i=0}^{n} A_i(x_i-t)_+^n\right| dt,$$

which is a constant not depending on f and using the notation

$$\|f^{(n+1)}\| = \max_{a\le x\le b}|f^{(n+1)}(x)|,$$

we obtain an error bound in quadrature formula:

$$|R_n(f)| \le K_n\|f^{(n+1)}\|.$$

ML.4.4 Trapezoidal Rule

The idea used to produce *trapezoidal rule* is to approximate the function $f:[a,b]\to\mathbb{R}$ by the Lagrange interpolating polynomial $L(a,b;f)$.

THEOREM ML.4.4.1

If $f\in C^2[a,b]$, then there exists $\xi\in(a,b)$ such that the following quadrature formula holds

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

Proof. We have

$$\int_a^b L(a,b;f)(x)dx$$

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

By using the weighted mean value theorem, we obtain

$$R_1(f) = \int_a^b (f(x) - L(a,b;f)(x))dx$$

$$= \int_a^b (x-a)(x-b)[a,b,x;f]dx = [a,b,c;f]\int_a^b (x-a)(x-b)dx$$

$$= \frac{f''(\xi)}{2}\int_a^b \left((x-a)^2 - (x-a)(b-a)\right)dx$$

$$= \frac{f''(\xi)}{2}\left(\frac{(x-a)^3}{3} - \frac{(x-a)^2}{2}(b-a)\right)\Bigg|_a^b$$

$$= -\frac{(b-a)^3}{12}f''(\xi).$$

□

We have proved that the trapezoidal quadrature formula has degree of precision 1. To generalize this procedure, choose a positive integer n. Subdivide the interval $[a,b]$ into n subintervals $[x_{i-1},x_i]$, $x_i = a + i\frac{b-a}{n}$, $i = 0,\ldots,n$, and apply trapezoidal rule on each subinterval. We obtain

$$\sum_{i=1}^{n}\frac{b-a}{2n}(f(x_{i-1}) + f(x_i)) = \frac{b-a}{n}\left(\frac{f(a)+f(b)}{2} + \sum_{i=1}^{n-1} f(x_i)\right)$$

and

$$R(f) = -\sum_{i=1}^{n}\frac{(b-a)^3}{12n^3}f''(\xi_i) = -\frac{(b-a)^3}{12n^2}\frac{1}{n}\sum_{i=1}^{n} f''(\xi_i)$$

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

where $h = \dfrac{b-a}{n}$. Consequently, there exists $\xi \in (a,b)$ such that

$$\int_a^b f(x)dx = \frac{b-a}{n}\left(\frac{f(a)+f(b)}{2} + \sum_{i=1}^{n-1} f(x_i)\right) - \frac{(b-a)^3}{12n^2}f''(\xi). \tag{4.4.1}$$

This formula is known as the *composite trapezoidal rule.* It is one of the simplest formulas for numerical integration.

ML.4.5 Richardson's Deferred Approach to the Limit

Let $T(h)$ be the result from the trapezoidal rule using step size h and $T(l)$ be the result with step size l, such that $h < l$.

If the second derivative f'' is *"reasonably constant"*, then:

$$\int_a^b f(x)\,dx \approx T(h) + Ch^2,$$

$$\int_a^b f(x)\,dx \approx T(l) + Cl^2.$$

We deduce

$$C \approx \frac{T(h) - T(l)}{l^2 - h^2},$$

hence

$$\int_a^b f(x)\,dx \approx T(h) + h^2\frac{T(h) - T(l)}{l^2 - h^2} =$$

$$= \frac{(\frac{l}{h})^2 T(h) - T(l)}{(\frac{l}{h})^2 - 1}.$$

This gives a better approximation to $\int_a^b f(x)\,dx$ than $T(h)$ and $T(l)$. In fact, if the second derivative is actually a constant, then the truncation error is zero.

ML.4.6 Romberg Integration

Romberg Integration uses the Composite Trapezoidal Rule to give preliminary approximations and then applies the Richardson Extrapolation process to improve the approximations.

Denoting by $T(h)$ the approximation of the integral $I = \int_a^b f(x)dx$ given by the Composite Trapezoidal rule,

$$T(h) = \frac{h}{2}\left(f(a) + f(b) + 2\sum_{i=1}^{m-1} f(a+ih)\right),$$

where $h = (b-a)/m$, it can be shown that if $f \in C^\infty[a,b]$, the Composite Trapezoidal rule can be written in the form

$$\int_a^b f(x)dx = T(h) + a_1h^2 + a_2h^4 + \cdots,$$

therefore, we can apply Richardson Extrapolation.

Let $n \in \mathbb{N}^*$,

$$h_k = \frac{b-a}{2^{k-1}}, \qquad k = 1, \ldots, n.$$

The following Composite Trapezoidal approximations with $m = 1, 2, \ldots, 2^{n-1}$, can be derived:

$$R_{1,1} = h_1\frac{f(a) + f(b)}{2},$$

$$R_{k,1} = \frac{h_k}{2}\left(f(a) + f(b) + 2\sum_{i=1}^{2^{k-1}-1} f(a+ih_k)\right),$$

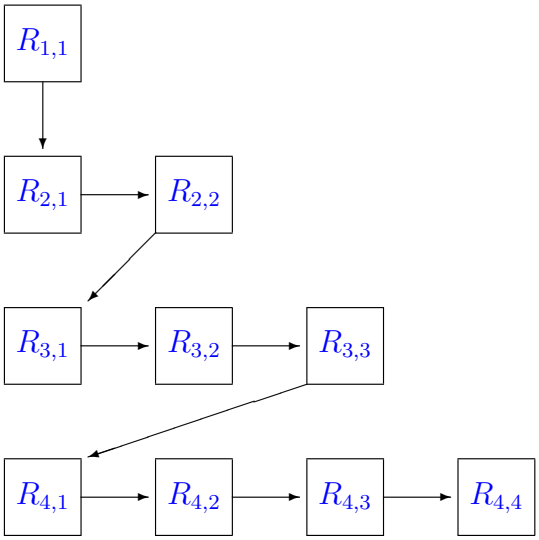
$k = 2, \ldots, n$. We can rewrite $R_{k,1}$ in the form:

$$R_{k,1} = \frac{1}{2}\left(R_{k-1,1} + h_{k-1}\sum_{i=1}^{2^{k-2}} f(a + (2i-1)h_k)\right), \qquad k = 2, \ldots, n.$$

Applying the Richardson Extrapolation to these values we obtain an $O(h_k^{2j})$ approximation formula defined by

$$R_{k,j} = \frac{4^{j-1}R_{k,j-1} - R_{k-1,j-1}}{4^{j-1}-1},$$
$$k = 2, \ldots, n; \quad j = 2, \ldots, k.$$

In the case of $n = 4$, the results generated by these formulas are shown in the following table.



ML.4.7 Newton-Cotes Formulas

Let $x_i = a + ih$, $i = 0, \ldots, m$ and $h = (b-a)/m$. The idea used to produce *Newton-Cotes formulas* is to approximate the function $f : [a, b] \to \mathbb{R}$ by the Lagrange Interpolating Polynomial $L(x_0, \ldots, x_m; f)$. We have

$$\int_a^b f(x)dx = \int_a^b L(x_0, \ldots, x_m; f)(x)dx + R(f),$$

$$\int_a^b f(x)dx = \sum_{i=0}^{m} f(x_i)A_i^{(m)} + R(f),$$

and denoting

$$\ell(x) = (x - x_0)\ldots(x - x_m),$$

we obtain

$$A_i^{(m)} = \int_a^b \frac{\ell(x)}{(x - x_i)\ell'(x_i)}dx,$$

$i = 0, \ldots, m$.

For some particular cases the values of the coefficients $A_i^{(m)}$ are given in the following table.

m	$A_i^{(m)}$	h	x_i
1	$\frac{1}{2}, \frac{1}{2}$	$b-a$	$a,\ b$
2	$\frac{1}{6}, \frac{4}{6}, \frac{1}{6}$	$\frac{b-a}{2}$	$a,\ \frac{a+b}{2},\ b$
3	$\frac{1}{8}, \frac{3}{8}, \frac{3}{8}, \frac{1}{8}$	$\frac{b-a}{3}$	$a,\ \frac{a+2b}{3}, \frac{2a+b}{3},\ b$

ML.4.8 Simpson's Rule

Simpson's Rule is a Newton-Cotes formula in the case $m = 3$.

THEOREM ML.4.8.1

If $f \in C^4[a,b]$ then there exists $\xi \in (a,b)$ such that

$$\int_a^b f(x)dx = \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right) - \frac{(b-a)^5}{2880} f^{(4)}(\xi).$$

The Composite Simpson's Rule can be written with its error term as

$$\int_a^b f(x)dx = \frac{b-a}{3n}\left(\frac{f(a)+f(b)}{2} + \sum_{i=1}^{n-1} f(x_i) + 2\sum_{i=0}^{n-1} f\left(\frac{x_i + x_{i+1}}{2}\right)\right) - \frac{f^{(4)}(\xi)(b-a)^5}{2880n^5},$$

where $\xi \in (a,b)$.

ML.4.9 Gaussian Quadrature

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

The problem is to find the nodes $x_0, \ldots, x_n \in [a,b]$ and the coefficients $A_0, \ldots, A_n$ such that the quadrature formula

$$\int_a^b f(x)\rho(x)dx \approx \sum_{i=0}^{n} A_i f(x_i)$$

have a maximal degree of precision.

Let $(\varphi_n)_{n\in\mathbb{N}}$ be the sequence of orthogonal polynomials associated with the weight function ρ, where degree$(\varphi_n) = n$. Let $x_0, \ldots, x_n$ be the roots of the polynomial φ_{n+1} and $L(x_0, \ldots, x_n; f)$ be the Lagrange interpolating polynomial.

THEOREM ML.4.9.1

The quadrature formula

$$\int_a^b f(x)\rho(x)dx \approx \sum_{i=0}^{n} A_i f(x_i) = \int_a^b L(x_0, \ldots, x_n; f)(x)\rho(x)dx$$

has degree of precision $2n + 1$.

Proof. Let P a polynomial of degree at most $2n + 1$. We divide the polynomial P to the polynomial φ_{n+1}:

$$P = Q\varphi_{n+1} + r,$$

where degree$(Q) \leq n$ and degree$(r) \leq n$. We have:

$$L(x_0, \ldots, x_n; P) = \underbrace{L(x_0, \ldots, x_n; Q\varphi_{n+1})}_{0} + L(x_0, \ldots, x_n; r) = r,$$

$$\begin{aligned} R(P) &= \int_a^b (P(x) - L(x_0, \ldots, x_n; P)(x))\rho(x)dx \\ &= \int_a^b (P(x) - r(x))\rho(x)dx \\ &= \int_a^b Q(x)\varphi_{n+1}(x)\rho(x)dx = 0. \end{aligned}$$

It follows that the degree of precision is at least $2n + 1$.

On the other hand, from the equalities

$$\begin{aligned} R(\varphi_{n+1}^2) &= \int_a^b (\varphi_{n+1}^2(x) - \underbrace{L(x_0, \ldots, x_n; \varphi_{n+1}^2)(x))}_{0}\rho(x)dx \\ &= \int_a^b \varphi_{n+1}^2(x)\rho(x)dx > 0, \end{aligned}$$

we deduce that it cannot be greater than $2n + 1$.

□

The following result is concerning the coefficients of the Gaussian quadrature formula.

THEOREM ML.4.9.2

The coefficients of the Gaussian quadrature formula are positive.

Proof. Let

$$\ell_i(x) = \frac{(x - x_0)(x - x_1) \cdots (x - x_n)}{x - x_i},$$

$i = 0, \ldots, n$. Taking into account the fact that the degree of the polynomials ℓ_i^2 is $2n$, and that the Gaussian quadrature formula has degree of precision $2n + 1$, we obtain

$$\int_a^b \ell_i^2(x)\rho(x)dx = \sum_{j=0}^{n} A_j \ell_i^2(x_j) + 0 = A_i \ell_i^2(x_i),$$

i.e.,

$$A_i = \frac{\int_a^b \ell_i^2(x)\rho(x)dx}{\ell_i^2(x_i)} > 0,$$

$i = 0, \ldots, n$.

ML.5

Elements of Approximation Theory

The study of approximation theory involves two general types of problems. One problem arises when a function is given explicitly but we wish to find a "simpler" type of function, such as a polynomial, that can be used to determine approximate values of the given function. The other problem in approximation theory is concerned with fitting functions to given data and finding the "best" function in a certain class that can be used to represent the data.

ML.5.1 Discrete Least Squares Approximation

Consider the problem of estimating the values of a function, given the values y_i of the function at distinct points $x_i, i = 1, \ldots, n$. An approach for this problem would be to find a line that could be used as an approximating function. The problem of finding the equation $y = ax + b$ of the best linear approximation in the absolute sense requires that values of a and b be found to minimize

$$\max_{1 \le i \le n} |y_i - (ax_i + b)|.$$

This is commonly called a *minimax* problem.

Another approach to determine the best linear approximation involves finding values of a and b to minimize

$$\sum_{i=1}^{n} |y_i - (ax_i + b)|.$$

This quantity is called the *absolute deviation.* The *least squares method* approach to this problem involves determining the best approximation line $y = ax + b$ that minimize the least squares error:

$$\sum_{i=1}^{n} (y_i - (ax_i + b))^2.$$

For a minimum to occur, it is necessary that

$$\begin{cases} 0 = \dfrac{\partial}{\partial a}\displaystyle\sum_{i=1}^{n}(y_i - (ax_i + b))^2 \\ 0 = \dfrac{\partial}{\partial b}\displaystyle\sum_{i=1}^{n}(y_i - (ax_i + b))^2 \end{cases}$$

that is

$$\begin{cases} \displaystyle\sum_{i=1}^{n}(y_i - ax_i - b)x_i = 0 \\ \displaystyle\sum_{i=1}^{n}(y_i - ax_i - b) = 0 \end{cases}$$

These equation simplify to the *normal equations:*

$$\begin{cases} a\displaystyle\sum_{i=n}^{n} x_i^2 + b\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} x_i y_i, \\ a\displaystyle\sum_{i=1}^{n} x_i + b \cdot n = \sum_{i=1}^{n} y_i. \end{cases}$$

The solution to this system is

$$a = \frac{n\displaystyle\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n\displaystyle\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2},$$

$$b = \frac{\displaystyle\sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} y_i - \sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i}{n\displaystyle\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}.$$

The general problem of approximating a set of data, $\{(x_i, y_i)\}_{i=1}^{n}$, with an algebraic polynomial

$$P_m(x) = \sum_{k=0}^{m} a_k x^k,$$

$m < n-1$, using the least squares procedure is handled in similar manner. It requires choosing the constants $a_0, \ldots, a_m$ to minimize the least squares error

$$E = \sum_{i=1}^{n}(y_i - P_m(x_i))^2.$$

For E to be minimized it is necessary that

$$\left\{\frac{\partial E}{\partial a_j} = 0, \qquad j = 0, \ldots, m.\right.$$

This gives $m+1$ normal equations in $m+1$ unknowns, a_j,

$$\left\{\sum_{k=0}^{m} a_k \sum_{i=1}^{n} x_i^{j+k} = \sum_{i=1}^{n} y_i x_i^j, \qquad j = 0, \ldots, m; \quad m < n-1.\right.$$

The matrix of the previous system

$$A = \begin{bmatrix} \sum_{i=1}^{n} x_i^0 & \cdots & \sum_{i=1}^{n} x_i^m \\ \vdots & \ddots & \vdots \\ \sum_{i=1}^{n} x_i^m & \cdots & \sum_{i=1}^{n} x_i^{2m} \end{bmatrix}$$

can be written as

$$A = B \cdot B^T,$$

where

$$B = \begin{bmatrix} x_1^0 & \cdots & x_n^0 \\ \vdots & \ddots & \vdots \\ x_1^m & \cdots & x_n^m \end{bmatrix}.$$

Taking into account the fact that the Vandermonde type determinant

$$\begin{vmatrix} x_1^0 & \cdots & x_{m+1}^0 \\ \vdots & \ddots & \vdots \\ x_1^m & \cdots & x_{m+1}^m \end{vmatrix}$$

is different from zero, it follows that $\operatorname{rank}(B) = m+1$, hence A is nonsingular. So, the normal equations have a unique solution.

ML.5.2 Orthogonal Polynomials and Least Squares Approximation

Suppose that $f \in C[a,b]$. A polynomial of degree n

$$P_n(x) = \sum_{k=0}^{n} a_k x^k$$

is required to minimize the error

$$\int_a^b (f(x) - P_n(x))^2 dx.$$

Define

$$E(a_0, \ldots, a_n) = \int_a^b (f(x) - \sum_{k=0}^{n} a_k x^k)^2 dx.$$

A necessary condition for the parameters $a_0, \ldots, a_n$ to minimize E is that

$$\left\{ \frac{\partial E}{\partial a_j} = 0, \qquad j = 0, \ldots, n \right.$$

or

$$\left\{ \sum_{k=0}^{n} a_k \int_a^b x^{j+k} dx = \int_a^b x^j f(x) dx, \qquad j = 0, \ldots, n \right.$$

The matrix of the above linear system,

$$\left[\frac{b^{j+k+1} - a^{j+k+1}}{j+k+1} \right]_{j,k=0,\ldots,n}$$

is known as the *Hilbert matrix.*

It is an ill-conditioned matrix which can be considered as classic example for demonstrating round-off error difficulties.

A different technique to obtain least squares approximation will be presented. Let us introduce the concept of a weight function and orthogonality.

DEFINITION ML.5.2.1

An integrable function w is called a *weight function* on the interval $[a, b]$ if $w(x) > 0$ for all $x \in (a, b)$.

DEFINITION ML.5.2.2

A set $\{\varphi_1, \ldots, \varphi_n\}$ is said to be an *orthogonal set of functions* over the interval $[a, b]$ with respect to a weight function w if

$$\int_a^b \varphi_j(x)\varphi_k(x)w(x)\,dx = \begin{cases} 0, & \text{when } j \neq k, \\ \alpha_k > 0, & \text{when } j = k \end{cases}$$

If, in addition, $\alpha_k = 1$ for each $k = 0, \ldots, n$, the set is said to be *orthonormal.*

EXAMPLE ML.5.2.3

The set of functions $\{\varphi_0, \ldots, \varphi_{2n-1}\}$,

$$\begin{aligned} \varphi_0(x) &= \tfrac{1}{\sqrt{2\pi}} \\ \varphi_k(x) &= \tfrac{1}{\sqrt{\pi}} \cos kx, \quad k = 1, \ldots, n \\ \varphi_{n+k}(x) &= \tfrac{1}{\sqrt{\pi}} \sin kx, \quad k = 1, \ldots, n-1 \end{aligned}$$

is an orthonormal set on $[-\pi, \pi]$ with respect to $w(x) = 1$.

EXAMPLE ML.5.2.4

The set of *Legendre polynomials*, $\{P_0, P_1, \ldots, P_n\}$ where

$$\begin{array}{lll} P_0 & = & 1, \\ P_1 & = & x, \\ P_2 & = & x^2 - \frac{1}{3} \\ \ldots & \ldots & \ldots \end{array}$$

is orthogonal on $[-1, 1]$ with respect to the weight function $w(x) = 1$.

EXAMPLE ML.5.2.5

The set of *Chebyshev polynomials*,
$\{T_0, T_1, \ldots, T_n\}$

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

is orthogonal on $[-1, 1]$ with respect to the weight function

$$w(x) = \frac{1}{\sqrt{1 - x^2}}.$$

ML.5.3 Rational Function Approximation

A *rational function* r of degree N has the form

$$r(x) = \frac{P(x)}{Q(x)}$$

where P and Q are polynomials whose degrees sum to N.

Rational functions whose numerator and denominator have the same or nearly the same degree generally produce approximation results superior to polynomial methods for the same amount of computation effort. (This is based on the assumption that the amount of computation effort required for division is approximately the same as for multiplication).

Rational functions also permit efficient approximation for functions which are not bounded near, but outside, the interval of approximation.

ML.5.4 Padé Approximation

Suppose r is a function of degree $N = m + n$ and has the form

$$r(x) = \frac{P(x)}{Q(x)} = \frac{p_0 + p_1 x + \cdots + p_n x^n}{q_0 + q_1 x + \cdots + q_m x^m}.$$

The function r is used to approximate a function f on a closed interval containing zero. $Q(0) \neq 0$ implies $q_0 \neq 0$. Without loss of generality, we can assume that $q_0 = 1$, for if is not the case we simply replace P by P/q_0 and Q by Q/q_0.

The *Padé approximation technique* chooses $N+1$ parameters so that

$$f^{(k)}(0) = r^{(k)}(0), \qquad k = 0, \ldots, N$$

Padé approximation is an extension of Taylor polynomial approximation to rational functions. When $m = 0$, it is just the Nth Maclaurin polynomial.

Suppose f has the Maclaurin series expansion

$$f(x) = \sum_{i=0}^{\infty} a_i x^i.$$

We have

$$f(x) - r(x) = \frac{\displaystyle\sum_{i=0}^{\infty} a_i x^i \sum_{i=0}^{m} q_i x^i - \sum_{i=0}^{n} p_i x^i}{Q(x)}$$

The condition $f^{(k)}(0) = r^{(k)}(0)$, $k = 0, \ldots, N$, is equivalent to the fact that $f - r$ has a root of multiplicity $N+1$ at zero. Consequently, we choose $q_1, q_2, \ldots, q_m$ and $p_0, p_1, \ldots, p_n$ such that

$$(a_0 + a_1 x + \cdots)(1 + q_1 x + \cdots + q_m x^m) - (p_0 + p_1 x + \cdots + p_n x^x)$$

has no terms of degree less than or equal to N. The coefficient of x^k in the previous expression is

$$\sum_{i=0}^{k} a_i q_{k-i} - p_k,$$

where we define:

$$p_k = 0, \qquad k = n+1, \ldots, N; \qquad q_k = 0, \qquad k = m+1, \ldots, N;$$

so that the rational function for Padé approximation results from

$$\left\{ \sum_{i=0}^{k} a_i q_{k-i} = p_k, \qquad k = 0, \ldots, N. \right.$$

EXAMPLE ML.5.4.1

The Maclaurin series expansion for e^x is

$$1 + x + \frac{x^2}{2} + \cdots + \frac{x^n}{n!} + \cdots$$

To find the Padé approximation of degree two with $n = 1$ and $m = 1$ to e^x, we choose p_0, p_1 and q_1 such that the coefficients of x^k, $k = 0, 1, 2$, in the expression

$$(1 + x + \frac{x^2}{2})(1 + q_1 x) - (p_0 + p_1 x)$$

are zero. Expanding and collecting terms produces

$$1 - p_0 + (1 + q_1 - p_1)x + (\frac{1}{2} + q_1)x^2 + \frac{q_1}{2}x^3$$

therefore

$$\begin{cases} 1 - p_0 & = & 0 \\ 1 + q_1 - p_1 & = & 0 \\ 1/2 + q_1 & = & 0 \end{cases}$$

The solution of this system is

$$p_0 = 1, \ p_1 = \frac{1}{2}, \ q_1 = -\frac{1}{2}.$$

The Padé approximation is

$$r(x) = \frac{2 + x}{2 - x}.$$

ML.5.5 Trigonometric Polynomial Approximation

Let m, n be two given integers such that $m > n \geq 2$. Suppose that a set of $2m$ paired data points $\{(x_j, y_j)\}_{j=0}^{2m-1}$ is given, where

$$x_j = -\pi + \frac{j}{m}\pi, \quad j = 0, \ldots, 2m - 1$$

Consider the set of functions $\{\varphi_0, \ldots, \varphi_m\}$, where

$$\begin{array}{lcll} \varphi_0(x) & = & \frac{1}{2} & \\ \varphi_k(x) & = & \cos kx, & k = 1, \ldots, n, \\ \varphi_{n+k}(x) & = & \sin kx, & k = 1, \ldots, n-1. \end{array}$$

The goal is to determine the trigonometric polynomial S_n,

$$S_n(x) = \sum_{k=0}^{2n-1} a_k \varphi_k(x),$$

or

$$S_n(x) = \frac{a_0}{2} + \sum_{k=1}^{n-1}(a_k \cos kx + a_{n+k} \sin kx) + a_n \cos nx,$$

to minimize

$$E(S_n) = \sum_{j=0}^{2m-1} (y_j - S_n(x_j))^2.$$

In order to determine the constants a_k, we use the fact that the system $\{\varphi_0, \ldots, \varphi_{2m-1}\}$ is orthogonal, i.e.,

$$\sum_{j=0}^{2m-1} \varphi_k(x_j)\varphi_l(x_j) = \begin{cases} 0, & k \neq l, \\ m, & k = l. \end{cases}$$

THEOREM ML.5.5.1

The constants in the summation

$$\boldsymbol{S_n(x) = \frac{a_0}{2} + \sum_{k=1}^{n-1}(a_k \cos kx + a_{n+k} \sin kx) + a_n \cos nx}$$

that minimize the least square sum

$$\boldsymbol{E(a_0, \ldots, a_{2n-1}) = \sum_{j=0}^{2m-1} (y_j - S_n(x_j))^2}$$

are

$$\boldsymbol{a_k = \frac{1}{m} \sum_{j=0}^{2m-1} y_j \cos kx_j}$$

for each $k = 0, \ldots, n$,

$$\boldsymbol{a_{n+k} = \frac{1}{m} \sum_{j=0}^{2m-1} y_j \sin kx_j}$$

for each $k = 1, \ldots, n-1$.

Proof. By setting the partial derivatives of E with respect to a_k to zero, we obtain

$$0 = \frac{\partial E}{\partial a_k} = 2 \sum_{j=0}^{2m-1} (y_j - S_n(x_j)(-\varphi_k(x_j)),$$

so

$$\sum_{j=0}^{2m-1} y_j \varphi_k(x_j) = \sum_{j=0}^{2m-1} S_n(x_j)\varphi_k(x_j)$$

$$= \sum_{j=0}^{2m-1} \sum_{l=0}^{2n-1} a_l \, \varphi_l(x_j)\varphi_k(x_j) = \sum_{l=0}^{2n-1} a_l \sum_{j=0}^{2m-1} \varphi_l(x_j)\varphi_k(x_j) = m \cdot a_k,$$

$k = 0, \ldots, 2n-1$.

□

ML.5.6 Fast Fourier Transform

It was shown in Section ML.5.5 that the least squares trigonometric polynomial has the form

$$S_m = \frac{a_0}{2} + a_m \cos mx + \sum_{k=1}^{m-1} (a_k \cos kx + b_k \sin kx)$$

where

$$a_k = \frac{1}{m} \sum_{j=0}^{2m-1} y_j \cos kx_j, \quad k = 0, \ldots, m,$$

$$b_k = \frac{1}{m} \sum_{j=0}^{2m-1} y_j \sin kx_j, \quad k = 0, \ldots, m-1.$$

The polynomial S_n can be used for interpolation with only a minor modification. The approximation of many thousands of data points require multiplication and addition operations numbering in the millions. The round-off error associated with this number of calculation generally dominates the approximation.

In 1965 J. W. Tukey, in a paper with J. W. Cooley, published in the Mathematics of Computation, introduced the important *Fast Fourier Transform* algorithm. They applied a different method to calculate the constants in the interpolating trigonometric polynomial. Their method reduces the number of calculations to thousands compared to millions for the direct technique.

The method described by Cooley and Tukey has come to be known either as *Cooley-Tukey Algorithm* or the *Fast Fourier Transform (FFT) Algorithm* and led to a revolution in the use of interpolatory trigonometric polynomials. The method consists of organizing the problem so that the number of data points being used can be easily factored, particularly into powers of two. The interpolation polynomial has the form

$$S_m(x) = \frac{a_0 + a_m \cos mx}{2} + \sum_{k=1}^{m-1} (a_k \cos kx + b_k \sin kx).$$

The nodes are given by

$$x_j = -\pi + \frac{j}{m}\pi, j = 0, \ldots, 2m-1,$$

and the coefficients are

$$a_k = \frac{1}{m} \sum_{j=0}^{2m-1} y_j \cos kx_j, \quad k = 0, \ldots, m,$$

$$b_k = \frac{1}{m}\sum_{j=0}^{2m-1} y_j \sin kx_j, \quad k = 0, \dots, m.$$

Despite both constant b_0 and b_m are zero, they have been added to the collection for notational convenience. Instead of directly evaluating the constants a_k and b_k the FFT method compute the complex coefficients

$$c_k = \sum_{j=0}^{2m-1} y_j \mathrm{e}^{\mathrm{i}\frac{\pi jk}{m}}, \quad k = 0, \dots, 2m-1.$$

We have

$$\frac{1}{m} c_k \mathrm{e}^{-\mathrm{i}\pi k} = \frac{1}{m}\sum_{j=0}^{2m-1} y_i \mathrm{e}^{\mathrm{i}k(-\pi+\frac{j}{m}\pi)} = \frac{1}{m}\sum_{j=0}^{2m-1} y_j \mathrm{e}^{\mathrm{i}kx_j}$$

$$= \frac{1}{m}\sum_{j=0}^{2m-1} y_j(\cos kx_j + \mathrm{i}\sin kx_j) = a_k + \mathrm{i}b_k.$$

Once the constants c_k have been determinate, a_k and b_k can be recovered using the equalities

$$a_k + \mathrm{i}b_k = \frac{1}{m} c_k \mathrm{e}^{-\mathrm{i}\pi k}, \quad k = 0, \dots, 2m-1$$

Suppose $m = 2^p$ for some positive integer p. For each $k = 0, \dots, 2m-1$, we have

$$c_k + c_{m+k} = \sum_{j=0}^{2m-1} y_j(\mathrm{e}^{\mathrm{i}\frac{\pi jk}{m}} + \mathrm{e}^{\frac{\pi j(m+k)}{m}}) = \sum_{j=0}^{2m-1} y_j \mathrm{e}^{\mathrm{i}\frac{\pi jk}{m}}(1 + \mathrm{e}^{\mathrm{i}\pi j}).$$

But

$$1 + \mathrm{e}^{\mathrm{i}\pi j} = \begin{cases} 0, & \text{if } j \text{ is odd,} \\ 2, & \text{if } j \text{ is even,} \end{cases}$$

so, replacing j by $2j$ in the index of the sum, we obtain

$$c_k + c_{m+k} = 2\sum_{j=0}^{m-1} y_{2j} \mathrm{e}^{\mathrm{i}\frac{\pi jk}{\frac{m}{2}}}.$$

Likewise

$$c_k - c_{m+k} = 2\mathrm{e}^{\mathrm{i}\frac{\pi k}{n}} \sum_{j=0}^{m-1} y_{2j+1} \mathrm{e}^{\mathrm{i}\frac{\pi jk}{\frac{m}{2}}}.$$

Therefore, we can recover c_k, and c_{m+k}, $k = 0, \dots, m-1$, hence all c_k can be determinate.

ML.5.7 The Bernstein Polynomial

Let f be a real valued function defined on the interval $[0,1]$.

DEFINITION ML.5.7.1

The polynomial $B_n(f)$ defined by

$$B_n(f)(x) = \sum_{k=0}^{n} \binom{n}{k} (1-x)^{n-k} x^k f\left(\frac{k}{n}\right), \qquad x \in [0,1],$$

is called the *Bernstein polynomial* related to the function f.

The polynomials

$$p_{n,k}(x) = \binom{n}{k} (1-x)^{n-k} x^k,$$

$k = 0, \ldots, n$, are called the *Bernstein polynomials.* They were introduced in mathematics in 1912 by S. N. Bernstein [Lor53] and can be derived from the binomial formula

$$1 = (1 - x + x)^n = \sum_{k=0}^{n} \binom{n}{k} (1-x)^{n-k} x^k.$$

It follows immediately from the definition of the Bernstein polynomials that they satisfy the recurrence relations

$$\begin{gathered} p_{n,k}(x) = (1-x)\, p_{n-1,k}(x) + x\, p_{n-1,k-1}(x), \\ n = 1, \ldots; \quad k = 1, \ldots, n, \end{gathered} \tag{5.7.1}$$

$$\begin{gathered} p_{n,k}(x) \left(\tfrac{k}{n} - x\right) = x(1-x)(p_{n-1,k-1}(x) - p_{n-1,k}(x)), \\ k = 0, \ldots, n+1, \end{gathered} \tag{5.7.2}$$

([DL93, p.305]), and they form a partition of unity

$$\sum_{k=0}^{n} p_{n,k}(x) = 1. \tag{5.7.3}$$

(For the purpose of this formulas $p_{n,-1}(x) := p_{n,n+1}(x) := 0$).

For a function $f \in C[0,1]$, formula (ML.5.7.1) produces a linear map $f \mapsto B_n f$ from $C[0,1]$ to $\mathcal{P}_n$, the space of polynomials of degree at most n. It is positive (i.e., satisfies $B_n f \geq 0$ if $f \geq 0$) bounded with norm 1, because of (5.7.3).

Using the translation operator T, $Tf(x) := f(x + 1/n)$, the identity operator I and the forward difference operator $\Delta = T - I$, we can write

$$B_n(f, x) = (x \cdot T + (1-x) \cdot I)^n f(0). \tag{5.7.4}$$

Consequently, we obtain the Taylor expansion

$$B_n(f, x) = (I + x \cdot \Delta)^n f(0), \tag{5.7.5}$$

i.e.,

$$B_n(f,x) = \sum_{i=0}^{n} \binom{n}{i} \Delta^i f(0)\, x^i. \tag{5.7.6}$$

It follows that

$$B_n(\mathcal{P}_k) \subseteq \mathcal{P}_k, \quad n,k = 0,\dots$$

It is instructive to compute $B_n e_k$, $e_k(x) = x^k$, $(k = 0,\dots)$. By using (5.7.6) we obtain

$$\begin{aligned}
B_n e_k(x) &= \sum_{i=0}^{k} \binom{n}{i} x^i \Delta^i e_k(0) \\
&= \sum_{i=0}^{k} \binom{n}{i} (T-I)^i e_k(0) x^i \\
&= \sum_{i=0}^{k} \binom{n}{i} \sum_{j=0}^{i} \binom{i}{j} (-1)^j T^{j-i} e_k(0) x^i \\
&= \sum_{i=0}^{k} \binom{n}{i} \sum_{j=0}^{i} \binom{i}{j} (-1)^{i-j} T^{j} e_k(0) x^i \\
&= \sum_{i=0}^{k} \sum_{j=0}^{i} (-1)^{i-j} \binom{n}{i} \binom{i}{j} \left(\frac{i}{n}\right)^j x_i, \\
&\qquad k = 0,\dots,n.
\end{aligned}$$

We have obtained

$$B_n e_k = \sum_{i=0}^{k} m_{ij} e_i, \qquad k = 0,\dots,n,$$

where

$$m_{ij} := \sum_{j=0}^{i} (-1)^{i-j} \binom{n}{i} \binom{i}{j} \left(\frac{i}{n}\right)^j, \tag{5.7.7}$$

$i,j = 0,\dots,n$ (see $\binom{i}{j} = 0$ if $j > i$).

The linear operator $B_n : \mathcal{P}_n \to \mathcal{P}_n$, can be written in matrix form

$$\begin{bmatrix} B_n e_0 \\ \vdots \\ B_n e_n \end{bmatrix} = \begin{bmatrix} m_{00} & \cdots & m_{0n} \\ \vdots & \ddots & \vdots \\ m_{n0} & \cdots & m_{nn} \end{bmatrix} \cdot \begin{bmatrix} e_0 \\ \vdots \\ e_n \end{bmatrix}.$$

Following similar analysis, one can easily compute $B_n(e_k)$, $e_k(x) = x^k$, for $k = 0,1,2$:

$$\begin{aligned}
B_n(e_0,x) &= 1, \\
B_n(e_1,x) &= x, \\
B_n(e_2,x) &= \tfrac{n-1}{n} x^2 + \tfrac{1}{n} x, \\
B_n(e_2,x) - e_2(x) &= \tfrac{x(1-x)}{n}.
\end{aligned}$$

By differentiating (5.7.5) r-times, we can express the derivatives of the Bernstein polynomial in terms of finite differences,

$$B_n^{(r)}(f,x) = n(n-1)\dots(n-r+1)\,\Delta^r\,(I + x\cdot\Delta)^{n-r}\,f(0), \tag{5.7.8}$$

$r = 0,\dots,n$, hence,

$$B_n^{(r)}(f,x) = n(n-1)\dots(n-r+1)\sum_{k=0}^{n-r} p_{n-r,k}(x)\,\Delta^r\,T^k\,f(0), \tag{5.7.9}$$

$r = 0,\dots,n$, that is,

$$B_n^{(r)}(f,x) = n(n-1)\dots(n-r+1)\sum_{k=0}^{n-r} p_{n-r,k}(x)\,\Delta^r\,f\left(\frac{k}{n}\right), \tag{5.7.10}$$

$r = 0,\dots,n$. From (5.7.8), we also obtain

$$B_n^{(r)}(f,x) = n(n-1)\dots(n-r+1)\sum_{k=0}^{n-r}\binom{n-r}{k}\,x^k\,\Delta^{r+k}\,f(0), \tag{5.7.11}$$

$r = 0,\dots,n$.

For a function $f \in C[0,1]$, the *modulus of continuity* is defined by:

$$\omega(f,t) := \sup_{0<h\le t}\ \sup_{0\le x\le 1-h} |f(x+h) - f(x)|, \quad 0 \le t \le 1.$$

One can easily prove that

$$\omega(f,\alpha\,t) \le (1+\alpha)\,\omega(f,t), \quad 0 \le t \le 1,\ \alpha \ge 0.$$

One of the most used property of the Bernstein operators is the approximation property.

THEOREM ML.5.7.2

If $f : [0,1] \to \mathbb{R}$ is bounded on $[0,1]$ and continuous at some point $x \in [0,1]$, then

$$\lim_{n\to\infty} B_n(f,x) = f(x).$$

Proof. Since f is bounded, then there exists a constant M such that

$$|f(x)| \le M, \qquad \forall\, x \in [0,1].$$

Let $\varepsilon > 0$. Since f is continuous at x there exists a $\delta > 0$ such that

$$|x-y| < \delta \quad \text{implies} \quad |f(x) - f(y)| < \frac{\varepsilon}{2}.$$

Let n_ε be a number such that

$$\frac{M}{2n\delta^2} < \frac{\varepsilon}{2} \quad \text{if } n > n_\varepsilon.$$

We have

$$\begin{aligned}
|B_n f(x) - f(x)| &= \left| \sum_{i=0}^{n} p_{n,i}(x) \left(f\left(\frac{i}{n}\right) - f(x) \right) \right| \\
&= \leq \sum_{i=0}^{n} p_{n,i}(x) \left| f\left(\frac{i}{n}\right) - f(x) \right| \\
&= \sum_{|\frac{i}{n}-x|\geq\delta} + \sum_{|\frac{i}{n}-x|<\delta} \\
&< \sum_{i=0}^{n} \frac{(x-\frac{i}{n})^2}{\delta^2} p_{n,i}(x) \left| f\left(\frac{i}{n}\right) - f(x) \right| + \sum_{i=0}^{n} p_{n,i}(x) \frac{\varepsilon}{2} \\
&\leq \frac{2Mx(1-x)}{n\delta^2} + \frac{\varepsilon}{2} \leq \frac{M}{2n\delta^2} + \frac{\varepsilon}{2} < \varepsilon.
\end{aligned}$$

Consequently

$$|B_n f(x) - f(x)| < \varepsilon, \qquad \forall\, n > n_\varepsilon.$$

□

Another result related to the approximation property of the Bernstein operator is the following.

THEOREM ML.5.7.3

[Pop42] **For all $f \in C[0,1]$, the following relation is valid**

$$|f(x) - B_n(f,x)| \leq \frac{3}{2}\omega\left(f, \frac{1}{\sqrt{n}}\right), \quad x \in [0,1].$$

Proof. We have

$$\begin{aligned}
|f(x) - B_n(f,x)| &= |B_n(1,x)\cdot f(x) - B_n(f,x)| \\
&= \left| \sum_{k=0}^{n} \binom{n}{k} (1-x)^{n-k} x^k \left(f(x) - f\left(\frac{k}{n}\right) \right) \right| \\
&\leq \sum_{k=0}^{n} \binom{n}{k} (1-x)^{n-k} x^k \omega\left(f, \left|x - \frac{k}{n}\right|\right).
\end{aligned}$$

On the other hand, we have

$$\omega\left(f, \left|x - \frac{k}{n}\right|\right) \leq \left(1 + \left|x - \frac{k}{n}\right| \delta^{-1}\right) \omega(f,\delta).$$

By using Schwartz inequality, we obtain

$$\begin{aligned}
&\sum_{k=0}^{n}\binom{n}{k}(1-x)^{n-k}x^k\left|x-\frac{k}{n}\right| \\
\leq\ &\sqrt{\sum_{k=0}^{n}\binom{n}{k}(1-x)^{n-k}x^k\left(x-\frac{k}{n}\right)^2}\cdot\sqrt{\sum_{k=0}^{n}\binom{n}{k}(1-x)^{n-k}x^k} \\
\leq\ &\sqrt{\frac{x(1-x)}{n}}\cdot\sqrt{1}\leq\frac{1}{2\sqrt{n}}.
\end{aligned}$$

Consequently,

$$|f(x)-B_n(f,x)|\leq\sum_{k=0}^{n}\binom{n}{k}(1-x)^{n-k}x^k\left(1+\frac{1}{2\sqrt{n}}\,\delta^{-1}\right)\omega(f,\delta).$$

With $\delta=n^{-1/2}$, the proof is concluded. □

Let $r\,,p\in\mathbb{N}$. One of most important shape preserving properties of the Bernstein operator is presented in the following theorem.

THEOREM ML.5.7.4

If $f:[0,1]\longrightarrow\mathbb{R}$, is convex of order r, then B_nf is also convex of order r.

It is the shape preserving property that makes from Bernstein polynomials one of the most used tool in CAGD. By using (5.7.10), the proof is obvious.

THEOREM ML.5.7.5

If $f\in C^p[0,1]$, then the sequence $(B_n^{(i)}(f))_{n=0}^{\infty}$ converges uniformly to $f^{(i)}(i=0,\ldots,p)$.

THEOREM ML.5.7.6

(Voronovskaja 1932, [DL93, p. 307]) **If f is bounded on $[0,1]$, differentiable in some neighborhood of a point $x\in[0,1]$, and has derivative $f''(x)$, then**

$$\lim_{n\to\infty}n((B_nf)(x)-f(x))=\frac{x(1-x)}{2}f''(x).$$

THEOREM ML.5.7.7

([Ara57]) **If $f:[0,1]\longrightarrow\mathbb{R}$ is continuous, then for every $x\in[0,1]$ there exist three distinct points $x_1,x_2,x_3\in[0,1]$ such that**

$$B_n(f,x)-f(x)=\frac{x(1-x)}{n}\,[x_1,x_2,x_3;\,f].$$

Taking into account its approximation properties, the Bernstein polynomial sequence are often used as an approximation tool.

ML.5.8 Bézier Curves

In the forties, the automobile industry wanted to develop a more modern curved shape for cars. Paul de Faget de Casteljau, at Citroën (1959), and Pierre Etienne Bézier, at Rénault (1962), had developed what is now known as the theory of *Bézier curves and surfaces.* Because de Casteljau never published his results, many of the entities in this theory (and of course the theory itself) are named after Bézier.

Let $(M_0, \ldots, M_n)$ be an ordered system of points in $\mathbb{R}^p$.

DEFINITION ML.5.8.1

The curve defined by

$$B[M_0, \ldots, M_n](t) := \sum_{k=0}^{n} \binom{n}{k} (1-t)^{n-k} t^k M_k, \quad t \in [0,1].$$

is called the *Bézier curve* related to the system $(M_0, \ldots, M_n)$.

The curve $B[M_0, \ldots, M_n]$ mimes the form of the system $(M_0, \ldots, M_n)$. The polygon formed by $M_0, \ldots, M_n$ is called the *Bézier polygon* or *control polygon.* Similarly, the polygon vertices M_i are called *control points* or *Bézier points.* As P. E. Bézier realized during 1960s it became an indispensable tool in computer graphics.

The Bézier curve has the following important properties:

- It begins at M_0, heading in the direction $\overline{M_0, M_1}$;
- It ends at M_n, heading in the direction $\overline{M_{n-1} M_n}$;
- It stays entirely within the convex hull of $\{M_0, \ldots, M_n\}$.

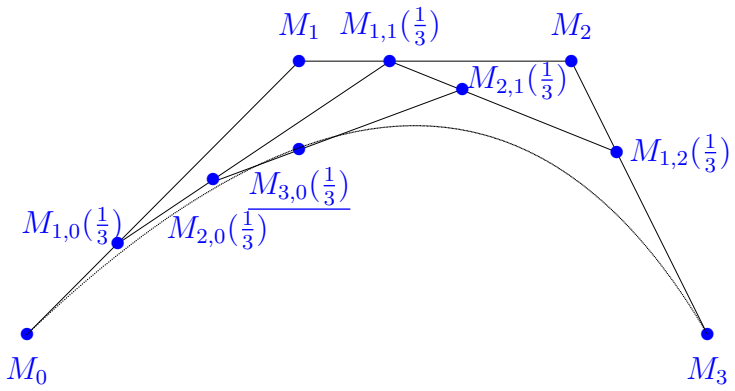
Let us present a recursive method used to determine the Bézier $B[M_0, \ldots, M_n]$ curve. The *Popoviciu-Casteljau Algorithm* presented by Tiberiu Popoviciu ([Pop37, p.39, (29)]) at classes (1933) (many years before Paul de Faget de Casteljau [dC59]), and described below, is probably the most fundamental one in the field of curve and surface design.

Let

$$\begin{aligned} M_{0,i}(t) &= M_i, \\ M_{r,i}(t) &= (1-t) M_{r-1,i}(t) + t\, M_{r-1,i+1}(t) \\ &\quad (r = 1, \ldots, n;\ i = 0, \ldots, n-r). \end{aligned}$$

The $M_{n,0}(t)$ is the point with parameter value t on the Bézier curve $B[M_0, \ldots, M_n]$.

The following picture presents an example with $n = 3$, $t = 1/3$.



Let $M_1, M_2, M_3 \in \mathbb{R}^3$. Another Bézier type curve is defined by Mircea Ivan in [Iva84]

$$\boldsymbol{I}_{\alpha[M_1,\,M_2,\,M_3]}(t) = (1-t)^\alpha M_1 + (1 - t^\alpha - (1-t)^\alpha)M_2 + t^\alpha M_3,$$

$t \in [0,1]$ and $\alpha > 0$.
For a suitable choice of control points, we obtain a continuous differentiable piecewise $\boldsymbol{I}_\alpha$ curve which closely mimes the control polygonal line.

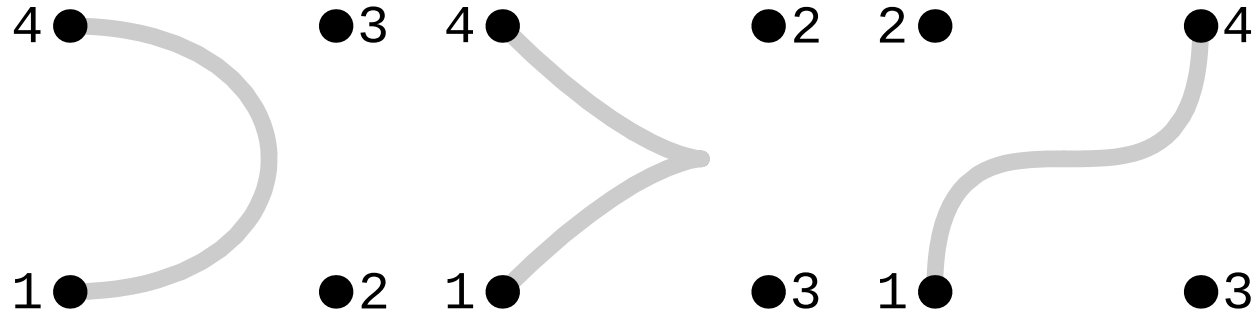
ML.5.9 The METAFONT Computer System

Sets of characters that are used in typesetting are traditionally known as *fonts* or *type.* Albrecht Dürer and other Renaissance men attempted to established mathematical principles of type design, but the letters they come up with were not especially beautiful. Their methods failed because they restricted themselves to "ruler and compass" constructions, which cannot adequately express the nuances of good calligraphy [Knu86, p. 13].

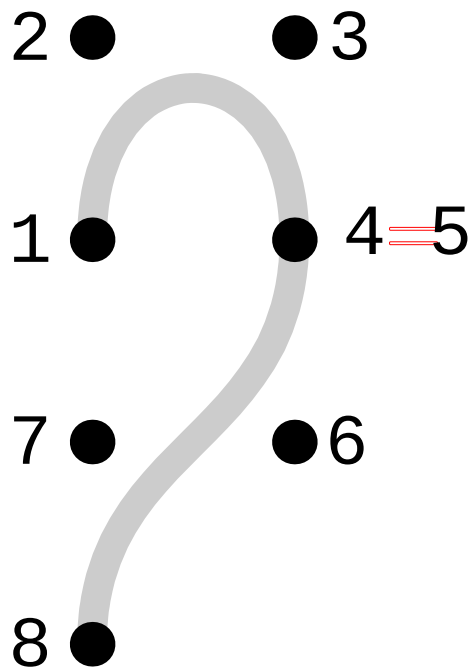
METAFONT [Knu86], a computer system for the design of fonts suited to raster-based modern printing equipment, gets around this problem by using powerful mathematical techniques. The basic idea is to start with four control points M_0, M_1, M_2, M_3 and consider the Bézier curve $B_3[M_0, M_1, M_2, M_3]$.

The fonts used to print this book were created by using Bézier curves.

Curves traced out by Bernstein polynomials of degree 3 are often called *Bézier cubics.* We obtain rather different curves from the same starting points if we number the points differently:



The four-point method is good enough to obtain satisfactory approximation to any curve we want, provided we break into enough segments and give four suitable control points for each segment.



ML.6

Integration of Ordinary Differential Equations

ML.6.1 Introduction

Equations involving one or several derivatives of a function are called *ordinary differential equations.* A problem involving ordinary differential equations is not completely specified by its equations. Even more important in determining how to solve the problem numerically is the nature of the problem's boundary conditions.

Boundary conditions can be divided into two broad categories:

- Initial value problems;
- Two-point boundary value problems.

Let $f : [a, b] \times [c, d] \to \mathbb{R}$ be a continuous function possessing continuous partial derivatives of order two.

We will consider the following initial value problem:

$$\begin{cases} \dfrac{dy}{dt} = f(t, y) \\ y(a) = y_0, \end{cases} \qquad t \in [a, b], \tag{6.1.1}$$

which has a unique solution $y \in C^2[a, b]$.

Define

$$h = \frac{b-a}{n}, \qquad t_i = a + ih, \quad i = 0, \ldots, n,$$

where n is a positive integer. Some numerical methods for solving the initial value problem (6.1.1) will be presented.

ML.6.2 The Euler Method

Expand the solution y of the problem (6.1.1) into a Taylor series,

$$y(t_{i+1}) = y(t_i) + hy'(t_i) + \frac{h^2}{2} y''(\xi_i),$$

$t_i < \xi_i < t_{i+1}, \quad i = 0, \ldots, n-1$. But $y'(t) = f(t, y(t))$, and hence

$$y(t_{i+1}) = y(t_i) + hf(t_i, y(t_i)) + \frac{h^2}{2} y''(\xi_i).$$

The *Euler Method* can be obtained by neglecting the remainder term $\frac{h^2}{2} y''(\xi_i)$, in the above expansion

$$y_{i+1} = y_i + hf(t_i, y_i), \quad i = 0, \ldots, n-1. \tag{6.2.1}$$

The Euler method is one of the oldest and best known numerical method for integrating differential equations. It can be improved in a number of ways.

ML.6.3 The Taylor Series Method

Suppose that equation (6.1.1) has a solution $y \in C^{p+1}[a, b]$. Also $y(t_{i+1})$ is approximated by a Taylor polynomial

$$y(t_i) + hy'(t_i) + \cdots + \frac{h^p}{p!} y^{(p)}(t_i).$$

Use the relations:

$$y'(t) = f(t, y(t)),$$
$$y''(t) = f'_t(t, y(t)) + f'_y(t, y(t)) \cdot f(t, y(t)), \quad \text{etc.},$$

and denote by $T_p(t, v)$ the expression obtained by replacing $y(t)$ by v in

$$g_p(t) = \sum_{j=0}^{p-1} \frac{h^{j+1}}{(j+1)!} \frac{d}{dt}^j (f(t, y(t))).$$

We obtain a *Taylor method* of order p :

$$y_{i+1} = y_i + T_p(t_i, y_i), \quad i = 0, \ldots, n-1. \tag{6.3.1}$$

For $p = 1$, we have

$$g_1(t) = h\, f(t, y(t)), \quad T_1(t, v) = h\, f(t, v),$$

hence we obtain the Euler's method:

$$y_{i+1} = y_i + h\, f(t_i, y_i), \quad i = 0, \ldots, n-1.$$

For $p = 2$, we get

$$g_2(t) = h\, f(t, y(t)) + \frac{h^2}{2} \frac{d}{dt} f(t, y(t))$$
$$= h\, f(t, y(t)) + \frac{h^2}{2} \left(f'_t(t, y(t)) + f'_y(t, y(t)) \cdot f(t, y(t)) \right),$$

hence

$$T_2(t,v) = h\, f(t,v) + \frac{h^2}{2}\left(f'_t(t,v) + f'_y(t,v)\cdot f(t,v)\right),$$

and we obtain the Taylor method of order two:

$$y_{i+1} = y_i + h\, f(t_i,y_i) + \frac{h^2}{2}\left(f'_t(t_i,y_i) + f'_y(t_i,y_i)\cdot f(t_i,y_i)\right),$$

$i = 0,\ldots,n-1.$

ML.6.4 The Runge-Kutta Method

Let y be a solution of equation (6.1.1). The constants

$$a_1,\ldots,a_p,\quad b_1,\ldots,b_{p-1},\quad c_{11},c_{21},c_{22}\ldots,c_{p-1,p-1},$$

given in the relations:

$$\begin{cases} k_1 &= h\, f(t_i,y(t_i)),\\ k_2 &= h\, f(t_i+b_1h, y(t_i)+c_{11}k_1),\\ \ldots & \ldots\\ k_p &= h\, f(t_i+b_{p-1}h, y(t_i)+c_{p-1,1}k_1+\cdots+c_{p-1,p-1}k_{p-1}),\end{cases}$$

will be determined such that the functions:

$$h\mapsto y(t_i+h),\qquad \text{and}\qquad h\mapsto y(t_i)+a_1k_1+\cdots+a_pk_p,$$

have the same Taylor approximation of order $o(h^p)$.

For example, when $p=2$, we need only to find the constants

$$a_1,a_2,b_1,c_{11},$$

given in the relations:

$$\begin{cases} k_1 &= h\, f(t_i,y(t_i)),\\ k_2 &= h\, f(t_i+b_1h, y(t_i)+c_{11}k_1),\end{cases}$$

such that the two functions:

$$h\mapsto y(t_i+h),\qquad \text{and}\qquad h\mapsto y(t_i)+a_1k_1+a_2k_2,$$

have the same Taylor approximation of order $o(h^2)$.

The approximation of order $o(h^2)$ of the function

$$y(t_{i+1}) = y(t_i+h)$$

is

$$\begin{aligned} &y(t_i)+y'(t_i)\,h+y''(t_i)\tfrac{h^2}{2}\\ &= y(t_i)+f(t_i,y(t_i))\,h+\left(f'_t(t_i,y(t_i))+f'_y(t_i,y(t_i))\,f(t_i,y(t_i))\right)\tfrac{h^2}{2},\end{aligned}$$

and the approximation of order $o(h^2)$ of the function

$$\begin{aligned} &y(t_i)+a_1k_1+a_2k_2\\ &= y(t_i)+a_1h\, f(t_i,y(t_i))+a_2h\, f(t_i+b_1h,y(t_i)+c_{11}k_1),\end{aligned}$$

is

$$\begin{gathered} y(t_i) + a_1 h\, f(t_i, y(t_i)) \\ +a_2 h(f(t_i, y(t_i)) + b_1 h\, f'_t(t_i, y(t_i)) + c_{11} k_1 f'_y(t_i, y(t_i))\, f(t_i, y(t_i))) \\ = y(t_i) + (a_1 + a_2) f(t_i, y(t_i)))\, h \\ +(a_2 b_1 f'_t(t_i, y(t_i)) + a_2 c_{11} f'_y(t_i, y(t_i))\, f(t_i, y(t_i)))\, h^2. \end{gathered}$$

Consequently, we obtain the system

$$\begin{cases} a_1 + a_2 &= 1 \\ a_2 b_1 &= \frac{1}{2} \\ a_2 c_{11} &= \frac{1}{2} \end{cases},$$

hence, by taking $a_1 = \frac{1}{2}$, we obtain:

$$a_2 = \frac{1}{2}, \quad b_1 = 1, \quad c_{11} = 1.$$

The Runge-Kutta algorithm of order two can be defined by the equations:

$$\begin{gathered} y_0 = y(t_0) \\ \begin{cases} k_1 &= h\, f(t_i, y_i) \\ k_2 &= h\, f(t_i + h, y_i + k_1) \\ y_{i+1} &= y_i + \frac{1}{2}(k_1 + k_2) \end{cases} \end{gathered} \qquad i = 0, \ldots, n-1.$$

The classical Runge-Kutta method (that of order four), can be defined by the following equations:

$$\begin{gathered} y_0 = y(t_0) \\ \begin{cases} k_1 &= h\, f(t_i, y_i) \\ k_2 &= h\, f(t_i + \frac{1}{2} h, y_i + \frac{1}{2} k_1) \\ k_3 &= h\, f(t_i + \frac{1}{2} h, y_i + \frac{1}{2} k_2) \\ k_4 &= h\, f(t_i + h, y_i + k_3) \\ y_{i+1} &= y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \end{cases} \end{gathered} \qquad i = 0, \ldots, n-1.$$

ML.6.5 The Runge-Kutta Method for Systems of Equations

Consider the system of differential equations

$$\begin{cases} \frac{dx_1}{dt} &= f_1(t, x_1, \ldots, x_n) \\ \vdots & \vdots \\ \frac{dx_n}{dt} &= f_n(t, x_1, \ldots, x_n) \end{cases}. \tag{6.5.1}$$

with initial conditions: $x_1(t_0) = x_1^0, \ldots, x_n(t_0) = x_n^0$.

Using the notations

$$F = [f_1, \ldots, f_n], \quad X = [x_1, \ldots, x_n],$$

system (6.5.1) can be written in the form

$$\frac{dX}{dt} = F(t, X),$$

with the initial condition $X(t_0) = X_0 = [x_1^0, \ldots, x_n^0]$.

The Runge-Kutta method of order four for system (6.5.1) is:

$$X_0 = X(t_0)$$

$$\begin{cases} K_1 &= h\,F(t_i, X_i) \\ K_2 &= h\,F(t_i + \frac{1}{2}\,h, X_i + \frac{1}{2}\,K_1) \\ K_3 &= h\,F(t_i + \frac{1}{2}\,h, X_i + \frac{1}{2}\,K_2) \\ K_4 &= h\,F(t_i + h, X_i + K_3) \\ X_{i+1} &= X_i + \frac{1}{6}\,(K_1 + 2K_2 + 2K_3 + K_4) \end{cases} \qquad i = 0, \ldots, n-1.$$

ML.7

Integration of Partial Differential Equations

ML.7.1 Introduction

Partial differential equations (PDE) are at the heart of many computer analysis and simulations of continuous physical systems.

The intend of this chapter is to give the briefest possible useful introduction. We limit our treatment to partial differential equations of order two. A linear partial differential equation of order two has the following form

$$A\,\frac{\partial^2 u}{\partial x^2} + 2B\,\frac{\partial^2 u}{\partial x \partial y} + C\,\frac{\partial^2 u}{\partial y^2} + D\,\frac{\partial u}{\partial x} + E\,\frac{\partial u}{\partial y} + F\,u + G = H, \tag{7.1.1}$$

where the coefficients $A, B, \ldots, H$ are functions of x and y. Equation (7.1.1) is called homogeneous when $H = 0$. Otherwise it is called nonhomogeneous. For a given domain, in most mathematical books, the partial differential equations are classified, on the basis of their discriminant $\Delta = B^2 - AC$, into three categories:

- elliptic, if $\Delta < 0$;
- parabolic, if $\Delta = 0$;
- hyperbolic, if $\Delta > 0$.

A general method to approximate the solution is the *grid method*. The first step is to choose integers n, m, and *step sizes* h, k such that

$$h = \frac{b-a}{n}, \qquad k = \frac{d-c}{m}.$$

Partitioning the interval $[a, b]$ into n equal parts of width h and the interval $[c, d]$ into m equal parts of width k provides a *grid* by drawing vertical and horizontal lines through the points with coordinates (x_i, y_j), where

$$x_i = a + ih, \qquad y_j = c + jk,$$

$i = 0, \ldots, n$; $j = 0, \ldots, m$. The lines $x = x_i$, $y = y_j$, are called *grid lines,* and their intersections are the *mesh points* of the grid.

ML.7.2 Parabolic Partial-Differential Equations

Consider a particular case of the heat or diffusion parabolic partial differential equation

$$\begin{aligned} &\frac{\partial u}{\partial t}(x,t) = \frac{\partial^2 u}{\partial x^2}(x,t), && 0 < x < l, \quad 0 < t < T \\ &\text{subject to conditions} \\ &u(0,t) = u(l,t) = 0, && 0 < t < T \\ &u(x,0) = f(x), \quad 0 \le x \le l, \end{aligned}$$

One of the algorithms used to approximate the solution of this equation is the *Crank-Nicolson algorithm.*

It is an implicit algorithm defined by the relations:

$$\frac{\partial u}{\partial t}(ih, jk) \approx \frac{u_{i,j+1} - u_{ij}}{k},$$

$$\frac{\partial^2 u}{\partial x^2}(ih, jk) \approx \frac{u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1} + u_{i+1,j} - 2u_{ij} + u_{i-1,j}}{2h^2}.$$

By letting $r = \frac{k}{h^2}$, we obtain the system:

$$2(1+r)u_{i,j+1} - ru_{i-1,j+1} - ru_{i+1,j+1} = ru_{i-1,j} + 2(1-r)u_{ij} + ru_{i+1,j}.$$

The values in the nodes of the line "$j+1$" are calculated using the already computed values in the line "j".

ML.7.3 Hyperbolic Partial Differential Equations

For this type of PDE, we consider the numerical solution of particular case of the *wave equation:*

$$\begin{aligned} &\frac{\partial^2 u}{\partial t^2}(x,t) = \frac{\partial^2 u}{\partial x^2}(x,t), && 0 < x < l, \quad 0 < t < T, \\ &\text{subject to conditions} \\ &u(0,t) = u(l,t) = 0, && 0 < t < T, \\ &u(x,0) = f(x), && 0 \le x \le l, \\ &\frac{\partial u}{\partial t}(x,0) = g(x), && 0 \le x \le l \end{aligned}$$

Using the finite difference method consider

$$\frac{\partial^2 u}{\partial x^2}(ih, jk) = \frac{u_{i+1,j} - 2u_{ij} + u_{i-1,j}}{h^2},$$

$$\frac{\partial^2 u}{\partial y^2}(ih, jk) = \frac{u_{i,j+1} - 2u_{ij} + u_{i,j-1}}{k^2},$$

hence, letting $r = \dfrac{k^2}{h^2}$, we obtain an explicit algorithm

$$u_{i,j+1} = -u_{i,j-1} + 2(1-r)u_{ij} + r(u_{i+1,j} + u_{i-1,j}).$$

ML.7.4 Elliptic Partial Differential Equations

The approximate solution of the Poisson elliptic partial differential equation

$$\frac{\partial^2 u}{\partial x^2}(x,y) + \frac{\partial^2 u}{\partial y^2}(x,y) = f(x,y), \qquad x \in [a,b], \quad y \in [c,d]$$

subject to boundary conditions

$$u(x,c) = f(x), \quad u(x,d) = g(x), \qquad x \in [a,b],$$
$$u(a,y) = f(y), \quad u(b,y) = g(y), \qquad y \in [c,d].$$

can be found by taking $h = k$,

$$\frac{\partial^2 u}{\partial x^2}(ih, jh) = \frac{u_{i+1,j} - 2u_{ij} + u_{i-1,j}}{h^2},$$

$$\frac{\partial^2 u}{\partial y^2}(ih, jh) = \frac{u_{i,j+1} - 2u_{ij} + u_{i,j-1}}{h^2},$$

Denoting $r = \dfrac{k^2}{h^2}$, we obtain an implicit algorithm

$$u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{ij} = f_{ij}.$$

ML.7

Self Evaluation Tests

ML.7.1 Tests

Let $x_0, \ldots, x_n$ be distinct points on the real axis, f be a function defined on these points and $\ell(x) = (x - x_0) \ldots (x - x_n)$.

TEST ML.7.1.1

Let P be a polynomial. The remainder obtained by dividing the polynomial P by the polynomial $(X - x_0) \ldots (X - x_n)$ is:

(A) The Lagrange Polynomial $\mathrm{L}(x_0, \ldots, x_n; P)$;

(B) The Lagrange Polynomial $\mathrm{L}(x_0, \ldots, x_n; (X - x_0) \ldots (X - x_n))$;

(C) The Lagrange Polynomial $\mathrm{L}(x_0, \ldots, x_n; X^n)$;

(D) The Lagrange Polynomial $\mathrm{L}(x_0, \ldots, x_n; X^n\, P)$

TEST ML.7.1.2

The polynomial $\mathrm{L}(x_0, \ldots, x_n;\ X^{n+1})$ is:

(A) X^{n+1};

(B) $X^{n+1} - (X - x_0)(X - x_1) \ldots (X - x_n)$;

(C) X^n;

(D) 0;

TEST ML.7.1.3

Let $P_n : \mathbb{R} \to \mathbb{R}$ be a polynomial of degree at most n, $n \in \mathbb{N}$. Prove the decomposition in partial fractions

$$\frac{P_n(x)}{(x-x_0)\dots(x-x_n)} = \sum_{k=0}^{n} \frac{1}{x-x_i} \cdot \frac{P(x_i)}{\ell'(x_i)}.$$

TEST ML.7.1.4

Calculate the sum

$$\sum_{i=0}^{n} \frac{x_i^k}{(x_i-x_0)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_n)},$$

$k = 0, \dots, n$.

TEST ML.7.1.5

Calculate $[x_0, \dots, x_n; x^{n+1}]$.

TEST ML.7.1.6

Calculate

$$[x_0, \dots, x_n; (X-x_0)\dots(X-x_k)],$$

$k \in \mathbb{N}$.

TEST ML.7.1.7

Prove that

$$\mathrm{L}(x_0, \dots, x_n; \mathrm{L}(x_1, \dots, x_n; f)) = \mathrm{L}(x_1, \dots, x_n; \mathrm{L}(x_0, \dots, x_n; f)).$$

TEST ML.7.1.8

Prove that, for $x \notin \{x_0, \dots, x_n\}$,

$$\left[x_0, \dots, x_n; \frac{f(t)}{x-t}\right]_t = \frac{\mathrm{L}(x_0, \dots, x_n;\ f)(x)}{(x-x_0)\dots(x-x_n)}.$$

TEST ML.7.1.9

If $f(x) = \frac{1}{x}$ and $0 < x_0 < x_1 < \cdots < x_n$, then

$$[x_0, \ldots, x_n;\ f] = \frac{(-1)^n}{x_0 \ldots x_n}.$$

TEST ML.7.1.10

For $x > 0$, we define the function

$$H(x) = 1 + \sum_{k=1}^{\infty} \frac{1}{(x+2)\ldots(x+k+1)}.$$

Show that

a) $H(x) = e(x+1)J(x)$, where

$$J(x) = \int_0^1 e^{-t} t^x dt.$$

b) $e^{\frac{1}{x+2}} \leq H(x) \leq \frac{e+x+1}{x+2}$.

TEST ML.7.1.11

a) Show that the following quadrature formula

$$\int_0^1 f(x)dx = \frac{1}{4}f(0) + \frac{3}{4}f\left(\frac{2}{3}\right) + R(f)$$

has the degree of exactness 2.

b) Let f be a function from $C^1[0,1]$. Show that there exists a point $\theta = \theta(f)$ such that

$$R(f) = \frac{1}{36}\left[\theta, \theta, 0, \frac{2}{3}; f\right].$$

TEST ML.7.1.12

Show that the following quadrature formula

$$\int_a^b f(x)dx = \frac{b-a}{4}\left[f(a) + 3f\left(\frac{a+2b}{3}\right)\right] + R(f)$$

has the degree of exactness 2 and for $f \in C^3[a,b]$, there exists $c = c(f) \in [a,b]$ such that

$$R(f) = \frac{(b-a)^4}{216} f'''(c).$$

TEST ML.7.1.13

Let $f \in C^3[a,b]$ and let us consider the following quadrature formula

$$\int_a^b f(x)dx = \frac{b-a}{4n}\sum_{k=0}^{n-1}\left[f\left(a + \frac{k(b-a)}{n}\right) + 3f\left(a + \frac{(3k+2)(b-a)}{3n}\right)\right].$$

Show that

$$\lim_{n\to\infty} n^3 R_n(f) = \frac{(b-a)^3}{216}\left[f''(b) - f''(a)\right].$$

TEST ML.7.1.14

Let Π_{2n}^+ be the set of all polynomials of degree $2n$ which are positive on the set

$$\left\{\cos\frac{(2k+1)\pi}{2n} \;\middle|\; k = 0, 1, ..., n-1\right\}$$

and having the leading coefficient 1. Using the gaussian quadrature formula:

$$\int_{-1}^{1} \frac{f(x)}{\sqrt{1-x^2}}dx = \sum_{k=0}^{n-1} A_k f\left(\cos\frac{(2k+1)\pi}{2n}\right) + R_n(f),$$

show that

$$\inf_{P\in\Pi_{2n}^+} \int_{-1}^{1} \frac{P(x)}{\sqrt{1-x^2}}dx = \frac{\pi}{2^{2n-1}}.$$

TEST ML.7.1.15

Find a quadrature formula of the following form

$$\int_0^1 f(x)dx = Af(0) + Bf(x_1) + Cf(x_2) + R(f),\ x_1 \neq x_2,\ x_1, x_2 \in (0,1),$$

with maximum degree of exactness.

TEST ML.7.1.16

Find a quadrature formula of the form

$$\begin{aligned}&\int_0^1 f(x)dx\\ =\ & c_1\left(f\left(\frac{1}{2}-x_1\right)+f\left(\frac{1}{2}+x_1\right)\right)+c_2\left(f'\left(\frac{1}{2}-x_1\right)-f'\left(\frac{1}{2}+x_1\right)\right)\\ &+R(f),\end{aligned}$$

$f \in C^1[0,1]$ with maximum degree of exactness.

TEST ML.7.1.17

Let $P_n(x) = x^n + a_1x^{n-1} + \cdots + a_n$ a polynomial with real coefficients. Show that

$$\int_0^1 P_n^2(x)dx \geq \frac{(n!)^5}{[(2n)!]^2(2n+1)}.$$

For what polynomial does the inequality above become equality?

ML.7.2 Answers to Tests

TEST ANSWER ML.7.1.1

Answer (A). Indeed, let $P(x) = Q(x)\,(x - x_0)\dots(x - x_n) + R(x)$. We have

$$R(x_i) = P(x_i), \qquad i = 0, \dots, n.$$

Since the degree of the polynomial R is at most n, we get $R = L(x_0, \dots, x_n; P)$.

TEST ANSWER ML.7.1.2

Answer (B). Indeed, we have

$$\begin{aligned} X^{n+1} &= 1 \cdot (X - x_0)(X - x_1)\dots(X - x_n) \\ &\quad + X^{n+1} - (X - x_0)(X - x_1)\dots(X - x_n). \end{aligned}$$

By Test (ML.7.1.1), we obtain

$$L(x_0, \dots, x_n;\, X^{n+1}) = X^{n+1} - (X - x_0)(X - x_1)\dots(X - x_n).$$

TEST ANSWER ML.7.1.3

Since the Lagrange operator $\mathrm{L}(x_0, \dots, x_n;\, \cdot)$ preserves polynomials with degree at most n, with $\ell(x) = (x - x_0)\dots(x - x_n)$, we have

$$\frac{P_n(x)}{\ell(x)} = \frac{\mathrm{L}(x_0, \dots, x_n;\, P_n)(x)}{\ell(x)} = \sum_{k=0}^{n} \frac{1}{x - x_i} \cdot \frac{P(x_i)}{\ell'(x_i)}.$$

TEST ANSWER ML.7.1.4

$$\begin{aligned} &\sum_{i=0}^{n} \frac{x_i^k}{(x_i - x_0)\dots(x_i - x_{i-1})(x_i - x_{i+1})\dots(x_i - x_n)} \\ &= [x_0, \dots, x_n;\, X^k] \\ &= \begin{cases} 0, & if \quad 0 \le k \le n - 1; \\ 1, & if \quad k = n. \end{cases} \end{aligned}$$

See(3.3.2).

TEST ANSWER ML.7.1.5

We have:

$$\begin{aligned} 0 &= [x_0, \dots, x_n; (X - x_0) \dots (X - x_n)] \\ &= [x_0, \dots, x_n; X^{n+1} - (x_0 + \dots + x_n)X^n + \dots] \\ &= [x_0, \dots, x_n; X^{n+1}] - (x_0 + \dots + x_n)[x_0, \dots, x_n; X^n] + 0 \\ &= [x_0, \dots, x_n; X^{n+1}] - (x_0 + \dots + x_n) \cdot 1 \end{aligned}$$

Consequently

$$[x_0, \dots, x_n; X^{n+1}] = x_0 + \dots + x_n.$$

TEST ANSWER ML.7.1.6

For $k < n$, the degree of the polynomial $(X - x_0) \dots (X - x_{k-1})$ is at most $n - 1$, hence

$$[x_0, \dots, x_n; (X - x_0) \dots (X - x_{k-1})] = 0.$$

For $k = n$,

$$\begin{aligned} & [x_0, \dots, x_n; (X - x_0) \dots (X - x_{n-1})] \\ = \; & [x_0, \dots, x_n; X^n - \dots] \\ = \; & [x_0, \dots, x_n; X^n] - 0 \\ = \; & 1. \end{aligned}$$

For $k > n$, the polynomial $(X - x_0) \dots (X - x_{k-1})$ takes zero value for $X = x_0, \dots x_n$, hence

$$[x_0, \dots, x_n; (X - x_0) \dots (X - x_{k-1})] = 0.$$

TEST ANSWER ML.7.1.7

Since the degree of the polynomial $L(x_1, \ldots, x_n; f)$ is at most $n-1$, we have

$$L(x_0, \ldots, x_n; L(x_1, \ldots, x_n; f)) = L(x_1, \ldots, x_n; f).$$

Consider the polynomial

$$P = L(x_1, \ldots, x_n; L[x_0, \ldots, x_n; f]).$$

We have:

$$\begin{aligned} P(x_i) &= L(x_1, \ldots, x_n; L[x_0, \ldots, x_n; f])(x_i) \\ &= L[x_0, \ldots, x_n; f](x_i), \qquad i = 1, \ldots, n \\ &= f(x_i), \qquad i = 1, \ldots, n, \end{aligned}$$

hence

$$P = L(x_1, \ldots, x_n; f).$$

TEST ANSWER ML.7.1.8

With $\ell(x) = (x - x_0) \ldots (x - x_n)$, we have

$$\mathrm{L}(x_0, \ldots, x_n;\ f)(x) = \sum_{k=0}^{n} \frac{\ell(x)}{x - x_i} \cdot \frac{f(x_i)}{\ell'(x_i)},$$

hence

$$\frac{\mathrm{L}(x_0, \ldots, x_n;\ f)(x)}{\ell(x)} = \sum_{k=0}^{n} \frac{\frac{f(x_i)}{x - x_i}}{\ell'(x_i)} = \left[x_0, \ldots, x_n;\ \frac{f(t)}{x - t} \right]_t.$$

TEST ANSWER ML.7.1.9

By using problem (ML.3.12.8), we have

$$\left[x_0,\dots,x_n;\ \frac{1}{t}\right]_t = -\frac{\mathrm{L}(x_0,\dots,x_n;\,1)(0)}{(0-x_0)\dots(0-x_n)} = \frac{(-1)^n}{x_0\dots x_n}.$$

SOLUTION 2. **We write the obvious equality**

$$\left[x_0,\dots,x_n;\frac{(t-t_0)\dots(t-x_n)}{t}\right]=0$$

in the form

$$\left[x_0,\dots,x_n;t^n-(t_0+\dots+t_n)t^{n-1}+\dots+(-1)^{n+1}\frac{t_0\dots x_n}{t}\right]=0,$$

i.e.,

$$1-(t_0+\dots+t_n)0+\dots+0+(-1)^{n+1}t_0\dots x_n\left[x_0,\dots,x_n;\frac{1}{t}\right]=0.$$

TEST ANSWER ML.7.1.10

a) We have,

$$\frac{1}{(x+2)...(x+k+1)} = \frac{\Gamma(x+2)\Gamma(k)}{\Gamma(x+k+2)}\frac{1}{(k-1)!}. \tag{7.2.1}$$

From (7.2.1), we get

$$\begin{aligned} H(x) &= 1+\sum_{k=1}^{\infty}\beta(x+2,k)\frac{1}{(k-1)!} \\ &= 1+\sum_{k=1}^{\infty}\frac{1}{(k-1)!}\int_0^1 t^{k-1}(1-t)^{x+1}dt = 1+\int_0^1\sum_{k=0}^{\infty}\frac{1}{k!}t^k(1-t)^{x+1}dt \\ &= 1+\int_0^1 e^t(1-t)^{x+1}dt = 1+e\int_0^1 e^{-t}t^{x+1}. \end{aligned}$$

Integrating by parts, we obtain

$H(x) = 1+e\left[-e^{-t}t^{x+1}\Big|_0^1 + (x+1)\int_0^1 e^{-t}t^x dt\right]$ or,

$H(x) = e(x+1)\int_0^1 e^{-t}t^x dt.$

b) The function e^{-t} is a convex function. We consider the weight function $w(t) = (x+1)t^x$, $t \in [0,1]$ and the quadrature formula

$$\int_0^1 w(t)f(t)dt = f\left(\frac{x+1}{x+2}\right) + R(f),$$

with $x \geq 0$ fixed. The above quadrature formula is a Gauss-type quadrature formula. Let f be a convex function. Then, $R(f) \geq 0$. For $f(t) = e^{-t}$, we obtain

$$H(x) \geq e\cdot e^{-\frac{x+1}{x+2}} \quad \text{or} \quad H(x) \geq e^{\frac{1}{x+2}}.$$

For the second inequality we use the trapezoidal rule with the weight function given by w. We obtain

$$\begin{aligned} H(x) &\leq e\left[1-\int_0^1 (x+1)t^{x+1}dt + e^{-1}\int_0^1 (x+1)t^{x+1}dt\right] \\ &= \frac{e+x+1}{x+2}. \end{aligned}$$

TEST ANSWER ML.7.1.11

a) We have

$$R(e_0) = \int_0^1 dx - \frac{1}{4} - \frac{3}{4} = 0, \qquad R(e_1) = \int_0^1 x dx - \frac{3}{4}\frac{2}{3} = 0$$
$$R(e_2) = \int_0^1 x^2 dx - \frac{3}{4}\frac{4}{9} = 0, \qquad R(e_3) = \int_0^1 x^3 dx - \frac{8}{27}\frac{3}{4} = \frac{1}{36}.$$

b) For the beginning, we remark that

$$f(x) - f\left(\frac{2}{3}\right) = \left(x - \frac{2}{3}\right)\left[x, \frac{2}{3}; f\right], \qquad \int_0^1 x\left(x - \frac{2}{3}\right) dx = 0 \tag{7.2.2}$$

From (7.2.2), we obtain

$$\int_0^1 x\left(x - \frac{2}{3}\right) f(x) dx = \int_0^1 x\left(x - \frac{2}{3}\right)^2 \left[x, \frac{2}{3}; f\right] dx. \tag{7.2.3}$$

Using the mean value theorem for integrals in (7.2.3), implies that there exists a point $c = c(f) \in [0, 1]$ such that

$$\int_0^1 x\left(x - \frac{2}{3}\right) f(x) dx = \left[c, \frac{2}{3}; f\right] \int_0^1 x\left(x - \frac{2}{3}\right)^2 dx = \frac{1}{36}\left[c, \frac{2}{3}; f\right]$$

Applying the mean value theorem for divided differences in (7.2.4), it follows that there exists a point $\theta = \theta(f) \in [0, 1]$ such that

$$\int_0^1 x\left(x - \frac{2}{3}\right) f(x) dx = \frac{1}{36} f'(\theta). \tag{7.2.4}$$

The quadrature formula is an interpolation type formula and therefore $R(f) = \int_0^1 x\left(x - \frac{2}{3}\right)\left[x, 0, \frac{2}{3}; f\right] dx$. From (7.2.4), we get

$$\int_0^1 x\left(x - \frac{2}{3}\right)\left[x, 0, \frac{2}{3}; f\right] dx = \frac{1}{36}\left[x, 0, \frac{2}{3}; f\right]'_{x=\theta}. \tag{7.2.5}$$

On the other hand, we have $\left[x, 0, \frac{2}{3}; f\right]' = \left[x, x, 0\frac{2}{3}; f\right]$ .From here, by using (7.2.5), we obtain $R(f) = \frac{1}{36}\left[\theta, \theta, 0, \frac{2}{3}; f\right]$.

TEST ANSWER ML.7.1.12

A simple computation shows that

$$R(e_i) = \int_a^b x^i dx - \frac{b-a}{4}\left[a^i + 3\left(\frac{a+2b}{3}\right)^i\right] = 0,$$

for $i = 0, 1, 2$ and

$$R(e_3) = \frac{(b-a)^4}{36}.$$

Let us consider the function $g : [0,1] \longrightarrow \mathbb{R}$ defined by $g(t) = f((1-t)a + bt)$. Using previous results (Test ML.7.1.11), we have

$$R(g) = \frac{1}{36}\left[\theta, \theta, 0, \frac{2}{3}; g\right].$$

Using the mean value theorem for divided differences, it follows that there exists a point $c = c(f) \in [0,1]$ such that

$$\left[\theta, \theta, 0, \frac{2}{3}; g\right] = \frac{g''(c)}{6}.$$

We have

$$g'''(t) = (b-a)^3 f'''((1-t)a + tb)$$

and

$$\int_0^1 g(t)dt = \frac{1}{4}\left[g(0) + 3g\left(\frac{2}{3}\right)\right] + \frac{g'''(c)}{216}$$

or

$$\int_0^1 f((1-t)a + tb)dt = \frac{1}{4}\left[f(a) + 3f\left(\frac{a+2b}{3}\right)\right] + \frac{(b-a)^3 f'''(\widetilde{c})}{216}.$$

If we make the change of variable $x = (1-t)a + tb$, we get

$$\int_a^b f(x)dx = \frac{b-a}{4}\left[f(a) + 3f\left(\frac{a+2b}{3}\right)\right] + \frac{(b-a)^4 f'''(\widetilde{c})}{216}.$$

TEST ANSWER ML.7.1.13

Let x_k be the points given by

$$x_k = a + k\frac{b-a}{n}, \quad k = \overline{0,n}.$$

Then

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

But,

$$\int_{x_{k-1}}^{x_k} f(x)dx = \frac{b-a}{4n}\left[f(x_{k-1}) + 3f\left(\frac{x_{k-1}+2x_k}{3}\right)\right] + \frac{(b-a)^4}{216n^4} f'''(c_{k,n}). \tag{7.2.6}$$

Since $c_{k,n} \in [x_{k-1}, x_k]$, $k = \overline{1,n}$,

$$\frac{b-a}{n}\sum_{k=1}^{n} f'''(c_{k,n})$$

is a Riemann sum for the function $f'''(\cdot)$, relative to the division

$$\Delta_n : \quad a = x_0 < x_1 < \ldots < x_n = b.$$

From (7.2.6), we get

$$\begin{aligned}
\lim_{n\to\infty} n^3 R_n(f) &= \frac{(b-a)^3}{216} \lim_{n\to\infty} \frac{b-a}{n} \sum_{k=1}^{n} f'''(c_{k,n}) \\
&= \frac{(b-a)^3}{216} \int_a^b f'''(x)dx \\
&= \frac{(b-a)^3}{216} [f''(b) - f''(a)].
\end{aligned}$$

TEST ANSWER ML.7.1.14

The coefficients A_k, $k = \overline{0, n-1}$ are positive. For $P \in \Pi_{2n}^{+}$, we have

$$\int_{-1}^{1} \frac{P(x)}{\sqrt{1-x^2}} dx \geq R_n(P).$$

Equality in the above relation takes place if and only if $P\left(\cos \frac{(2k+1)\pi}{2n}\right) = 0$ for $k = \overline{0, n-1}$. Since $P \in \Pi_{2n}^{+}$, it follows that $P(x) = T_n^2(x)$, where T_n is the Chebysev polynomial relative to the weight $w(x) = \frac{1}{\sqrt{1-x^2}}$, having the leading coefficient 1, i.e.,

$$T_n(x) = \frac{1}{2^{n-1}} \cos(n \arccos x).$$

On the other hand, we have $R_n(P) = R_n\left(x^{2n}\right)$, for any $P \in \Pi_{2n}^{+}$. Therefore,

$$R_n(P) = R_n\left(x^{2n}\right) = R_n\left(T_n^2(x)\right).$$

From the gaussian quadrature formula , we get

$$R_n\left(T_n^2(x)\right) = \int_{-1}^{1} \frac{T_n^2(x)}{\sqrt{1-x^2}} dx = \int_{-1}^{1} \frac{\cos^2(n \arccos x)}{2^{2n-2}\sqrt{1-x^2}} dx$$

By making the change of variable $x = \cos t$ in the last integral, we get

$$R_n\left(T_n^2(x)\right) = \frac{1}{2^{2n-2}} \int_0^{\pi} \cos^2(nt) dt = \frac{\pi}{2^{2n-1}}.$$

So far, we have proved that

$$R_n(P) = \frac{\pi}{2^{2n-1}},$$

for any $P \in \Pi_{2n}^{+}$. Therefore,

$$\int_{-1}^{1} \frac{P(x)}{\sqrt{1-x^2}} dx \geq \frac{\pi}{2^{2n-1}}.$$

The last inequality together with the identity

$$\int_{-1}^{1} \frac{T_n^2(x)}{\sqrt{1-x^2}} dx = \frac{\pi}{2^{2n-1}}$$

concludes our proof.

TEST ANSWER ML.7.1.15

If the quadrature formula is of interpolation type, having the nodes $0, x_1, x_2$, then its degree of exactness is at least 2. Let P be a polynomial of degree ≥ 3. Then

$$P(x) - L_2(P; 0, x_1, x_2)(x) = x(x - x_1)(x - x_2)Q(x), \tag{7.2.7}$$

where Q is a polynomial such that $degree(Q) = degree(P) - 3$. From (7.2.7), we get,

$$R(P) = \int_0^1 x(x - x_1)(x - x_2)Q(x)dx.$$

If $R(P) = 0$, then the maximum degree of Q is 1. In this case, x_1 and x_2 are the roots of the orthogonal polynomial of degree 2 with respect to the weight function $w(x) = x$. This polynomial is $l(x) = x^2 - \frac{6}{5}x + \frac{3}{10}$. The nodes x_1 and x_2 are the roots of $l(x)$,

$$x_1 = \frac{6 - \sqrt{6}}{10}, \; x_1 = \frac{6 + \sqrt{6}}{10}.$$

The coefficients B, C are given by

$$\begin{aligned} B &= \frac{1}{x_1(x_2 - x_1)} \int_0^1 x(x_2 - x)dx = \frac{16 + \sqrt{6}}{36}, \\ C &= \frac{1}{x_2(x_2 - x_1)} \int_0^1 x(x - x_1)dx = \frac{16 - \sqrt{6}}{36}. \end{aligned}$$

Since the quadrature formula has the degree of exactness 3, we must have $R(l) = 0$. Since,

$$R(l) = \int_0^1 l(x)dx - Al(0),$$

it follows that

$$A = \frac{\int_0^1 l(x)dx}{l(0)} = \frac{1}{3}.$$

The quadrature formula now reads

$$\int_0^1 f(x)dx = \frac{1}{3}f(0) + \frac{16 + \sqrt{6}}{36} f\left(\frac{6 - \sqrt{6}}{10}\right) + \frac{16 - \sqrt{6}}{36} f\left(\frac{6 + \sqrt{6}}{10}\right) + R(f) \tag{7.2.8}$$

and has degree of exactness 4.

Remark. The quadrature formula (7.2.8) is known as Bouzita's quadrature formula.

TEST ANSWER ML.7.1.16

The following calculations hold

$$\begin{aligned}
R(e_0) = 0 &\Leftrightarrow 2c_1 = 1,\\
R(e_1) = 0 &\Leftrightarrow 2c_1 = 1,\\
R(e_2) = 0 &\Leftrightarrow c_1\left(\frac{1}{2} + 2x_1^2\right) - 4x_1c_2 = \frac{1}{3},\\
R(e_3) = 0 &\Leftrightarrow c_1\left[1 - 3\left(\frac{1}{4} - x_1^2\right)\right] - 6x_1c_2 = \frac{1}{4}.
\end{aligned}$$

We get

$$c_1 = \frac{1}{2}, \quad x_1^2 - 4x_1c_2 = \frac{1}{12}.$$

Since

$$R(e_3) = 0 \Leftrightarrow x_1^4 + 6x_1^2 - 4c_2x_1\left(\frac{3}{2} + x_1^2\right) = \frac{11}{80},$$

it follows that x_1 and c_2 are the solutions of the system

$$x_1^2 - 4x_1c_2 = \frac{1}{12}, \quad x_1^4 + 6x_1^2 - 4c_2x_1\left(\frac{3}{2} + x_1^2\right) = \frac{11}{80}.$$

For x_1, we get the equation

$$x_1^4 + 6x_1^2 - \left(x_1^2 - \frac{1}{12}\right)\left(x_1^2 + \frac{3}{2}\right) = \frac{11}{80},$$

which has the unique solution

$$x_1 = \frac{\sqrt{33}}{110} \in \left(0, \frac{1}{2}\right).$$

It follows that

$$c_2 = \frac{x_1^2 - 1/12}{4x_1} = \frac{143}{24\sqrt{33}}.$$

The quadrature has degree of exactness 4 and reads

$$\begin{aligned}
&\int_0^1 f(x)dx\\
&= \frac{1}{2}\left(f\left(\frac{1}{2} - x_1\right) + f\left(\frac{1}{2} + x_1\right)\right) + \frac{143}{24\sqrt{33}}\left(f'\left(\frac{1}{2} - x_1\right) - f'\left(\frac{1}{2} + x_1\right)\right)\\
&\quad + R(f),
\end{aligned}$$

with $x_1 = \frac{\sqrt{33}}{110}$.

TEST ANSWER ML.7.1.17

Let us consider the gaussian formula of the form

$$\int_0^1 f(x)dx = \sum_{k=1}^{n} A_k f(x_k) + R(f),$$

where x_k, $k = 1, ..., n$ are the roots of the Legendre polynomial

$$l_n(x) = \frac{(n!)^2}{(2n)!}\frac{d^n}{dx^n}\left[x^n(x-1)^n\right].$$

The quadrature formula above is exact for any polynomial of degree $\leq 2n-1$. Since the coefficients A_k are positive, we obtain $\int_0^1 P_n^2(x)dx \geq R\left(P_n^2\right)$. Using the fact that $R(\cdot)$ is a linear functional, we obtain

$$R\left(P_n^2\right) = R\left(x^{2n}\right) + R\left(P_n^2(x) - x^{2n}\right).$$

Since $P_n^2(x) - x^{2n}$ is a polynomial of degree at most $2n-1$, it follows that $R\left(P_n^2(x) - x^{2n}\right) = 0$ and therefore $R\left(P_n^2\right) = R\left(x^{2n}\right) = R\left(l_n^2(x)\right)$. We have $R\left(l_n^2(x)\right) = \frac{(n!)^2}{(2n)!}\int_0^1 x^n \frac{d^n}{dx^n}\left[x^n(x-1)^n\right]dx$. Performing n times integration by parts, gives

$$R\left(l_n^2(x)\right) = \frac{(n!)^2}{(2n)!}n!\int_0^1 x^n(1-x)^n dx.$$

Since

$$\begin{aligned}\int_0^1 x^n(1-x)^n dx &= \beta(n+1, n+1) = \frac{\Gamma^2(n+1)}{\Gamma(2n+2)} \\ &= \frac{(n!)^2}{(2n+1)!},\end{aligned}$$

we obtain $R\left(l_n^2(x)\right) = \frac{(n!)^5}{[(2n)!]^2(2n+1)}$ and the inequality is proved. Equality holds if and only if $\sum_{k=1}^{n} A_k P_n^2(x_k) = 0$. Since the coefficients A_k are positive, it follows that we must have

$$P_n(x_k) = 0, \ k = \overline{1, n}.$$

It follows that equality holds if and only if $P_n = l_n$.

Index

Bibliography

[Ara57] Oleg Arama. Some properties concerning the sequence of polynomials of S. N. Bernstein (in Romanian). *Studii si Cerc. (Cluj)*, 8(3–4): 195–210, 1957.

[Bak76] N. Bakhvalov. *Méthodes Numériques*. Edition Mir, Moscou, 1976.

[BDL83] R. E. Barnhill, R. P. Dube, and F. F. Little. Properties of Shepard's surfaces. *Rocky Mountain Journal of Mathematics*, 13(2): 365–382, 1983.

[BF93] Richard L. Burden and J. Douglas Faires. *Numerical Analysis*. PWS-KENT Publishing Company, 1993.

[Blu72] E. K. Blum. *Numerical analysis and computation theory and practice*. Addison-Wesley Publishing Co., Reading, Mass.-London-Don Mills, Ont., 1972. Addison-Wesley Series in Mathematics.

[Boo60] G. Boole. *A treatise of the calculus of finite differences*. Macmillan, Cambridge, London, 1860.

[Bre83] Claude Brezinski. Outlines of Padé approximation. In H. Werner et al. eds., editor, *Computational Aspects of Complex Analysis*, pages 1–50. Reidel, Dordrecht, 1983.

[But87] J. C. Butcher. *The numerical analysis of ordinary differential equations*. A Wiley-Interscience Publication. John Wiley & Sons Ltd., Chichester, 1987. Runge-Kutta and general linear methods.

[BVI95] C. Brezinski and J. Van Iseghem. A taste of Padé approximation. In *Acta numerica, 1995*, Acta Numer., pages 53–103. Cambridge Univ. Press, Cambridge, 1995.

[dB78] Carl de Boor. *A practical guide to splines*. Springer-Verlag, New York Heidelberg Berlin, 1978.

[DB03a] Germund Dahlquist and Åke Björck. *Numerical methods*. Dover Publications Inc., Mineola, NY, 2003. Translated from the Swedish by Ned Anderson, Reprint of the 1974 English translation.

[dB03b] Carl de Boor. A divided difference expansion of a divided difference. *J. Approx. Theory*, 122(1):10–12, 2003.

[dB05] Carl de Boor. Divided differences. *Surveys in Approximation Theory*, 1:46–69, 2005.

[dC59] Paul de Faget de Casteljau. Outilages méthodes calcul. Technical report, A. Citroen, Paris, 1959.

[DL93] Ronald A. DeVore and George G. Lorentz. *Constructive approximation*. Springer Verlag, Berlin Heidelberg New York, 1993.

[DM72] W. S. Dorn and D. D. McCracken. *Numerical Methods with FORTRAN IV Case Studies*. John Wiley & Sons, Inc., New York - London - Sydney - Toronto, 1972.

[Far90] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design – A Practical Guide*. Academic Press, Inc., Boston - San Diego - New York - London, 1990.

[Gav01] Ioan Gavrea. *Aproximarea functiilor prin operatori liniari*. Mediamira, Cluj-Napoca, 2001.

[Gon95] H. H. Gonska. Shepard Methoden. In *Hauptseminar CAGD 1995*, pages 1–34, Duisburg, 1995. Gerhard Mercator Universität.

[GZ93] H. H. Gonska and X. Zhou. Polynomial approximation with side conditions: Recent results and open problems. In *Proc. of the First International Colloquium on Numerical Analysis, Plovdiv 1992*, pages 61–71. Zeist/The Netherlands: VSP International Science Publishers, 1993.

[Her78] Charles Hermite. Sur la formule d'interpolation de Lagrange. *J. Reine Angew. Math.*, 84:70–79, 1878.

[HL89] J. Hoschek and D. Lasser. *Fundamentals of Computer Aided Geometric Design*. A K Peters, Ltd, 1989.

[IP92] Mircea Ivan and Kálmán Pusztai. *Mathematics by Computer*. Comprex Publishing House, Cluj-Napoca, 1992.

[IP03] Mircea Ivan and Kálmán Pusztai. *Numerical Methods with Mathematica*. Mediamira, Cluj-Napoca, 2003.

[Iva73] Mircea Ivan. Mean value theorems in mathematical analysis (Romanian). Master's thesis, Babes-Bolyai University, Cluj, 1973.

[Iva82] Mircea Ivan. *Interpolation Methods and Applications*. PhD thesis, Babes-Bolyai University, Cluj-Napoca, 1982.

[Iva84] Mircea Ivan. On some Bernstein-Bézier type functions. In *Itinerant Seminar on Functional Equations, Approximation and Convexity*, pages 81–84, Cluj-Napoca, 1984. Babeş-Bolyai University Press.

[Iva98a] Mircea Ivan. A proof of Leibniz's formula for divided differences. In *Proceedings of the third Romanian–German Seminar on Approximation Theory (RoGer–98)*, pages 15–18, Sibiu, June 1–3, 1998, 1998.

[Iva98b] Mircea Ivan. A proof of Leibniz's formula for divided differences. In *Proceedings of the third Romanian-German Seminar on Approximation Theory (RoGer-98)*, pages 15–18, Sibiu, June 1-3 1998.

[Iva02] Mircea Ivan. *Calculus*, volume 1. Editura Mediamira, Cluj-Napoca, 2002.

[Iva04] Mircea Ivan. *Elements of Interpolation Theory*. Mediamira Science Publisher, Cluj-Napoca, 2004.

[Iva05] Mircea Ivan. *Numerical Analysis with Mathematica*. Mediamira Science Publisher, Cluj-Napoca, 2005. ISBN 973-713-051-0, 252p.

[Kag03] Yasuo Kageyama. A note on zeros of the Lagrange interpolation polynomial of the function $1/(z-c)$. *Transactions of the Japan Society for Industrial and Applied Mathematics*, 13:391–402, 2003.

[Kar53] S. Karlin. *Total Positivity*. Stanford University Press, 1953.

[KM84] N. V. Kopchenova and I. A. Maron. *Computational Mathematics*. Mir Publishers, Moscow, 1984.

[Knu86] Donald E. Knuth. *The METAFONTbook*. Addison Wesley Publishing Company, Reading-Massachusetts, 1986.

[Kor60] P. P. Korovkin. *Linear operators and approximation theory*. Translated from the Russian ed. (1959). Russian Monographs and Texts on Advanced Mathematics and Physics, Vol. III. Gordon and Breach Publishers, Inc., New York, 1960.

[Loc82] F. Locher. *Numerische Mathematik für Informatiker*. Springer-Verlag, Berlin-Heidelberg-New York, 1982.

[Lor53] G. G. Lorentz. *Bernstein polynomials*. Mathematical Expositions, no. 8. University of Toronto Press, Toronto, 1953.

[Mei02] Erik Meijering. A chronology of interpolation: From Ancient Astronomy to Modern Signal and Image Processing. *Proceedings of the IEEE*, 90(3):319–342, March 2002.

[MŞ81] G. I. Marciuk and V. V. Şaidurov. *Creşterea preciziei soluţiilor în scheme cu diferenţe*. Editura Academiei RSR, Bucureşti, 1981.

[New87] Isaac Newton. *Philosophiae Naturalis Principia Mathematica*. Printed by Joseph Streater by order of the Royal Society, London, 1687.

[NPI+57] Miron Nicolescu, Gh. Pic, D.V. Ionescu, E. Gergely, L Németi, L Bal, and F Radó. The mathematical activity of Professor Tiberiu Popoviciu. *Studii şi cerc. de matematică (Cluj)*, 8(1–2):7–19, 1957.

[Nue89] Guenther Nuernberger. *Approximation by spline functions*. Springer, Berlin, 1989.

[OR03] John O'Connor's and Edmund F. Robertson. MacTutor History of Mathematics Archive. School of Mathematics and Statistics. University of St Andrews. Scotland, 2003.

[Pal03] Radu Paltanea. *Approximation by Linear Positive Operators: Estimates with Second Order Moduli*. Editura Universităţii Transilvania, Braşov, 2003.

[Pop33] Tiberiu Popoviciu. *Sur quelques proprietées des fonctions d'une ou de deux variables réelles*. PhD thesis, La Faculté des Sciences de Paris, June 1933.

[Pop34] Tiberiu Popoviciu. Sur quelques propriété des fonctions d'une ou de deux variables réelles. *Mathematica (Cluj)*, VIII:1–85, 1934.

[Pop37] Tiberiu Popoviciu. *Despre cea mai bună aproximaţie a funcţiilor continue prin polinoame. Cinci lecţii ţinute la Facultatea de Ştiinţe din Cluj în anul şcolar 1933–34 (Romanian)*. Inst. Arte Grafice Ardealul, Cluj, 1937.

[Pop40] Tiberiu Popoviciu. Introduction à la théorie des différences divisées. *Bull. Math. Soc. Roumaine Sci.*, 42(1):65–78, 1940.

[Pop42] Tiberiu Popoviciu. Sur l'approximation des fonctions continues d'une variable réelle par des polynômes. *Ann. Sci. Univ. Jassy. Sect. I., Math.*, 28:208, 1942.

[Pop59] Tiberiu Popoviciu. Sur le reste dans certaines formules lineaires d'approximation de l'analyse. *Mathematica (Cluj)*, 1(24):95–142, 1959.

[Pop72] Elena Popoviciu. *Mean Value Theorems and their Connection to the Interpolation Theory (Romanian)*. Editura Dacia, Cluj, 1972.

[Pop75] Tiberiu Popoviciu. *Numerical Analysis. An Introduction to Approximate Calculus (Romanian)*. Editura Academiei Republicii Socialiste România, Bucharest, 1975.

[Pow81] M. J. D. Powell. *Approximation theory and methods*. Cambridge University Press, Cambridge, 1981.

[PT95] L. Piegl and W. Tiller. *The NURBSbook*. Springer-Verlag, Berlin - Heidelberg - New York, 1995.

[PTTF92] W. H. Press, S. A. Teukolsky, Vetterling W. T., and B. P. Flannery. *Numerical Recipes in C*. Cambridge University Press, 1992.

[RV98] I. Raşa and T. Vladislav. *Analiză Numerică*. Editura Tehnică, Bucureşti, 1998.

[SB61] M. G. Salvadori and M. L. Baron. *Numerical Methods in Engineering*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1961.

[Şe83] I. Gh. Şabac and et al. *Matematici speciale vol II*. Editura Didactică şi Pedagogică, Bucureşti, 1983.

[Ste39] J.F. Steffensen. Note on divided differences. *Danske Vid. Selsk. Math.-Fys. Medd.*, 17(3):1–12, 1939.

[VIS00] Ioan-Adrian Viorel, Dumitru Mircea Ivan, and Loránd Szabó. *Metode numerice cu aplicaţii în ingineria electrică*. Editura Universităţii din Oradea, Oradea, 2000. (nr. pag. 210) 973-8083-29-X.

[Vol90] E. A. Volkov. *Métodos numéricos*. Editorial Mir, Moscow, 1990. Translated from the Russian by K. P. Medkov.