NUMERICAL LINEAR ALGEBRA WITH APPLICATIONS
*Numer. Linear Algebra Appl.* 2000; **00**:1–6 *Prepared using* nlaauth.cls *[Version: 2000/03/22 v1.0]*

# Change of basis in polynomial interpolation

W. Gander

*Institute of Computational Science*
*ETH Zurich*
*CH-8092 Zurich*
*Switzerland*

SUMMARY

Several representations for the interpolating polynomial exist: Lagrange, Newton, orthogonal polynomials etc. Each representation is characterized by some basis functions. In this paper we investigate the transformations between the basis functions which map a specific representation to another. We show that for this purpose the $LU$- and the $QR$ decomposition of the Vandermonde matrix play a crucial role. Copyright © 2000 John Wiley & Sons, Ltd.

KEY WORDS: Polynomial interpolation, basis change

## 1. REPRESENTATIONS OF THE INTERPOLATING POLYNOMIAL

Given function values

| $x$ | $x_0$, | $x_1$, | $\cdots$ | $x_n$ |
|---|---|---|---|---|
| $f(x)$ | $f_0$, | $f_1$, | $\cdots$ | $f_n$ |

with $x_i \neq x_j$ for $i \neq j$. There exists a unique polynomial $P_n$ of degree less or equal $n$ which interpolates these values, i.e.

$$P_n(x_i) = f_i, \quad i = 0, 1, \ldots, n. \tag{1}$$

Several representations of $P_n$ are known, we will present in this paper the basis transformations among four of them.

### *1.1. Monomial Basis*

We consider first the monomials $\boldsymbol{m}(x) = (1, x, x^2, \ldots, x^n)^T$ and the representation

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

†Please ensure that you use the most up to date class file, available from the NLA Home Page at `http://www.interscience.wiley.com/jpages/1070-5325/`

Contract/grant sponsor: Publishing Arts Research Council; contract/grant number: 98–1846389


*Received ??*
*Revised ??*

The coefficients $\boldsymbol{a} = (a_0, a_1, \ldots, a_n)^T$ are determined by the interpolation condition (1) as solution of the linear system $V\boldsymbol{a} = \boldsymbol{f}$ with the Vandermonde matrix

$$V = \begin{pmatrix} 1 & x_0 & \ldots & x_0^{n-1} & x_0^n \\ 1 & x_1 & \ldots & x_1^{n-1} & x_1^n \\ \vdots & \vdots & & \vdots & \vdots \\ 1 & x_n & \ldots & x_n^{n-1} & x_n^n \end{pmatrix}$$

and the right hand side $\boldsymbol{f} = (f_0, f_1, \ldots, f_n)^T$. With this notation the interpolating polynomial becomes $P_n(x) = \boldsymbol{a}^T\boldsymbol{m}(x)$.

*1.2. Lagrange Basis*

A second representation is by means of the *Lagrange polynomials* $\boldsymbol{l}(x) = (l_0(x), l_1(x), \ldots, l_n(x))^T$ with

$$l_i(x) = \prod_{\substack{j=0\\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j}, \quad i = 0, 1, \ldots, n.$$

Since $l_i(x_i) = 1$ and $l_i(x_j) = 0$ for $i \neq j$ the interpolation polynomial can be written as linear combination

$$P_n(x) = \sum_{j=0}^{n} f_j \; l_j(x) = \boldsymbol{f}^T\boldsymbol{l}(x). \tag{2}$$

Interpolating with the Lagrange formula (2) is not very efficient, since for every new value $x$ we have to perform $O(n^2)$ operations. There exists a variant called the *Barycentric Formula* [5] which requires only $O(n)$ operations per interpolation point. We define the coefficients

$$\lambda_i = \frac{1}{(x_i - x_0)\cdots(x_i - x_{i-1})(x_i - x_{i+1})\cdots(x_i - x_n)}, \quad i = 0, \ldots, n.$$

Then for new interpolation points $x$ we compute the weights $\mu_i = \lambda_i/(x - x_i)$ and evaluate so

$$P_n(x) = \frac{\sum_{i=0}^{n} \mu_i(x) f_i}{\sum_{i=0}^{n} \mu_i(x)} \tag{3}$$

with only $O(n)$ operations. Thus in this form the Lagrange polynomials are computed by

$$l_i(x) = \frac{\mu_i(x)}{\sum_{i=0}^{n} \mu_i(x)}, \quad i = 0, 1, \ldots, n.$$

*1.3. Newton Basis*

The basis polynomials are the Newton polynomials $\boldsymbol{\pi}(x) = (\pi_0(x), \pi_1(x), \ldots, \pi_n(x))^T$ with

$$\pi_0(x) \equiv 1, \quad \pi_k(x) = \prod_{j=0}^{k-1}(x - x_j), \quad k = 1, \ldots, n.$$

The interpolation polynomial becomes

$$P_n(x) = d_0\pi_0(x) + d_1\pi_1(x) + \cdots d_n\pi_n(x) = \boldsymbol{d}^T\boldsymbol{\pi}(x)$$

where the coefficients $\boldsymbol{d} = (d_0, d_1, \dots, d_n)^T$ are obtained from the interpolation condition (1) as solution of the linear system $U^T \boldsymbol{d} = \boldsymbol{f}$ with the lower triangular matrix

$$\begin{pmatrix} \pi_0(x_0) & \cdots & \pi_n(x_0) \\ \pi_0(x_1) & \cdots & \pi_n(x_1) \\ \vdots & \cdots & \vdots \\ \pi_0(x_n) & \cdots & \pi_n(x_n) \end{pmatrix} = \begin{pmatrix} 1 & & & & \\ 1 & x_1 - x_0 & & & \\ 1 & x_2 - x_0 & (x_2 - x_0)(x_2 - x_1) & & \\ \vdots & \vdots & \vdots & \ddots & \\ 1 & x_n - x_0 & (x_n - x_0)(x_n - x_1) & \cdots & \prod_{j=0}^{n-1}(x_n - x_j) \end{pmatrix}.$$

The matrix $U$ is (upper) triangular since $\pi_k(x_j) = 0$, $j < k$. Notice that an alternative way to compute the coefficients $\boldsymbol{d}$ is by means of the divided differences:

$$\begin{array}{llllll} x_0 & f_0 = f[x_0] & & & & \\ x_1 & f_1 = f[x_1] & f[x_0, x_1] & & & \\ x_2 & f_2 = f[x_2] & f[x_1, x_2] & f[x_0, x_1, x_2] & & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \\ x_n & f_n = f[x_n] & f[x_{n-1}, x_n] & f[x_{n-2}, x_{n-1}, x_n] & \cdots & f[x_0, \dots, x_n] \end{array}$$

which are defined recursively by

$$f[x_i, x_{i+1}, \dots, x_{i+k}] = \frac{f[x_{i+1}, x_{i+1}, \dots, x_{i+k}] - f[x_i, x_{i+1}, \dots, x_{i+k-1}]}{x_{i+k} - x_i}.$$

The coefficients are given by the diagonal of the divided difference scheme

$$\boldsymbol{d} = (d_0, d_1, \dots, d_n)^T = (f[x_0], f[x_0, x_1], \dots, f[x_0, \dots, x_n])^T.$$

### *1.4. Aitken-Neville-Interpolation*

This representation of the interpolating polynomial is based on a hierarchical computation of interpolating polynomials.

Let $T_{ij}(x)$ be the polynomial of degree less or equal $j$ that interpolates the data

$$\begin{array}{c|cccc} x & x_{i-j}, & x_{i-j+1} & \cdots & x_i \\ \hline f(x) & f_{i-j}, & f_{i-j+1} & \cdots & f_i \end{array}.$$

We arrange these polynomials in a lower triangular matrix (the so called *Aitken-Neville scheme*) (see [4]):

$$\begin{array}{c|lllll} x & y & & & & \\ \hline x_0 & f_0 = T_{00} & & & & \\ x_1 & f_1 = T_{10} & T_{11} & & & \\ \vdots & \vdots & & \ddots & & \\ x_i & f_i = T_{i0} & T_{i1} & \cdots & T_{ii} & \\ \cdots & \cdots & \cdots & & \cdots & \ddots \end{array} \tag{4}$$

The polynomials $T_{ij}$ can be computed through the following recursion

$$\left.\begin{array}{rcl} T_{i0} & = & f_i \\ T_{ij} & = & \dfrac{(x_i - x)T_{i-1,j-1} + (x - x_{i-j})T_{i,j-1}}{x_i - x_{i-j}} \\ j & = & 1, 2, \dots, i \end{array}\right\} \quad i = 0, 1, 2, \dots \tag{5}$$

The interpolating polynomial for the $n+1$ interpolation points then becomes $P_n(x) = T_{nn}(x)$. This representation of the interpolating polynomial is effective and usually used for extrapolation for some *fixed numerical value of* $x$.

### *1.5. Orthogonal Polynomials*

A set $\{p_j(x)\}$ of polynomials is said to be *orthogonal* if

$$\langle p_j, p_k \rangle = 0, \quad j \neq k$$

where the indices $j$ and $k$ indicate the degrees. The scalar product is defined in our case on the discrete set $\{x_i\}, i = 0, \ldots, n$:

$$\langle p_j, p_k \rangle = \sum_{i=0}^{n} p_j(x_i) p_k(x_i).$$

Orthogonal polynomials are related by a three term recurrence (see e.g. [2])

$$p_{-1}(x) \equiv 0, \quad p_0(x) \equiv 1$$
$$p_{k+1}(x) = (x - \alpha_{k+1}) p_k(x) - \beta_k p_{k-1}(x), \quad k = 0, 1, 2, \ldots$$

where

$$\alpha_{k+1} = \frac{\langle x\, p_k, p_k \rangle}{\|p_k\|^2} \qquad \beta_k = \frac{\|p_k\|^2}{\|p_{k-1}\|^2}.$$

We use here the norm: $\|p_k\|^2 = \langle p_k, p_k \rangle$. Thus the coefficients $\alpha_k$ and $\beta_k$ and the value of the polynomials in the nodes $x_i$ can be computed recursively

$$k = 0 : \Rightarrow \alpha_1, p_1, \qquad k = 1 : \Rightarrow \alpha_2, \beta_1, p_2, \quad \text{etc.}$$

Let $\boldsymbol{p}(x) = (p_0(x), \ldots, p_n(x))^T$ and consider now the approximation problem

$$b_0 p_0(x_j) + b_1 p_1(x_j) + \ldots + b_k p_k(x_j) \approx f(x_j), \quad j = 0, \ldots, n \tag{6}$$

or in matrix notation $P\boldsymbol{b} \approx \boldsymbol{f}$

$$\begin{pmatrix} p_0(x_0) & p_1(x_0) & \cdots & p_k(x_0) \\ p_0(x_1) & p_1(x_1) & \cdots & p_k(x_1) \\ \vdots & \vdots & \ldots & \vdots \\ p_0(x_n) & p_1(x_n) & \cdots & p_k(x_n) \end{pmatrix} \begin{pmatrix} b_0 \\ \vdots \\ b_k \end{pmatrix} \approx \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{pmatrix} \tag{7}$$

If $k = n$ then we have $n+1$ equations for $n+1$ unknowns $b_i$. However, if $k < n$ then we will solve (7) as a least squares problem. Since the columns of the matrix $P$ are orthogonal due to the orthogonality of the polynomials the solution is easily obtained with the normal equations:

$$P^T P \boldsymbol{b} = P^T \boldsymbol{f}.$$

Because $P^T P = D^2$ is diagonal with $D = \text{diag}\{\|p_0\|, \|p_1\|, \ldots, \|p_k\|\}$ the solution is

$$b_j = \frac{\sum\limits_{i=0}^{n} p_j(x_i) f_i}{\sum\limits_{i=0}^{n} p_j(x_i)^2} = \frac{\langle p_j, \boldsymbol{f} \rangle}{\|p_j\|^2}, \quad j = 0, \ldots, k.$$

For $k = n$ we obtain the interpolating polynomial in the form $P_n(x) = \boldsymbol{b}^T \boldsymbol{p}(x)$.



*Prepared using* nlaauth.cls

## 2. BASIS TRANSFORMATIONS

We consider the following four representations of the interpolating polynomial

$$P_n(x) = \boldsymbol{a}^T\boldsymbol{m}(x) = \boldsymbol{f}^T\boldsymbol{l}(x) = \boldsymbol{d}^T\boldsymbol{\pi}(x) = \boldsymbol{b}^T\boldsymbol{p}(x).$$

The question we would like to answer is: *what are the transformation matrices between the basis* $\boldsymbol{m}(x)$*,* $\boldsymbol{l}(x)$*,* $\boldsymbol{\pi}(x)$ *and* $\boldsymbol{p}(x)$*?*

### *2.1. Lagrange Representations*

We use the following important observation to relate the Lagrange polynomials to another basis. Let $f_i = Q_k(x_i)$, $i = 0, 1, \ldots, n$ be function values of a polynomial $Q_k$ of degree $k \le n$. Then

$$\sum_{i=0}^{n} f_i\, l_i(x) = \sum_{i=0}^{n} Q(x_i)\, l_i(x) = Q_k(x). \tag{8}$$

Equation (8) is called the *Lagrange-representation* of the polynomial $Q_k$. Using this relation, it is straightforward to obtain the following mappings:

**a)** *Lagrange – monomials*: ($V$ is the Vandermonde matrix):

$$V^T\boldsymbol{l}(x) = \boldsymbol{m}(x)$$

Recall that for the coefficients for the monomial basis we have the relation $V\boldsymbol{a} = \boldsymbol{f}$.

**b)** *Lagrange – Newton*: $U\boldsymbol{l}(x) = \boldsymbol{\pi}(x)$ where

$$U = \begin{pmatrix} \pi_0(x_0) & \pi_0(x_1) & \cdots & \pi_0(x_n) \\ \pi_1(x_0) & \pi_1(x_1) & \cdots & \pi_1(x_n) \\ \vdots & \vdots & \vdots & \vdots \\ \pi_n(x_0) & \pi_n(x_1) & \cdots & \pi_n(x_n) \end{pmatrix} \tag{9}$$

$$= \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ & x_1 - x_0 & x_2 - x_0 & \cdots & x_n - x_0 \\ & & (x_2 - x_0)(x_2 - x_1) & \cdots & (x_n - x_0)(x_n - x_1) \\ & & & \ddots & \vdots \\ & & & & \prod_{j=0}^{n-1}(x_n - x_j) \end{pmatrix}$$

is upper triangular. Recall that the coefficients $\boldsymbol{d}$ for the Newton basis are the solution of $U^T\boldsymbol{d} = \boldsymbol{f}$.

An explicit expression for $U^{-1}$ exists. The divided differences are symmetric functions of their arguments. This is seen from the representation given in [5]:

$$f[x_0, \ldots, x_k] = \sum_{j=0}^{k} \frac{f_j}{\prod\limits_{\substack{p=0 \\ p\ne j}}^{k} (x_j - x_p)}. \tag{10}$$



*Prepared using* nlaauth.cls

Notice that $\prod_{\substack{p=0 \\ p\neq j}}^{k}(x_j - x_p) = \pi_k'(x_j)$ and therefore

$$d_k = f[x_0, \ldots, x_k] = \sum_{j=0}^{k} \frac{f_j}{\pi_k'(x_j)}$$

which is in matrix notation $\boldsymbol{d} = U^{-T}\boldsymbol{f}$ with

$$U^{-T} = \begin{pmatrix} \frac{1}{\pi_1'(x_0)} & & & \\ \frac{1}{\pi_2'(x_0)} & \frac{1}{\pi_2'(x_1)} & & \\ \frac{1}{\pi_3'(x_0)} & \frac{1}{\pi_3'(x_1)} & \frac{1}{\pi_3'(x_2)} & \\ \ldots & \ldots & \ldots & \ddots \end{pmatrix}.$$

Thus we obtain

$$U^{-1} = \begin{pmatrix} \frac{1}{\pi_1'(x_0)} & \frac{1}{\pi_2'(x_0)} & \cdots & \frac{1}{\pi_{n+1}'(x_0)} \\ & \frac{1}{\pi_2'(x_1)} & \cdots & \frac{1}{\pi_{n+1}'(x_1)} \\ & & \ddots & \vdots \\ & & & \frac{1}{\pi_{n+1}'(x_n)} \end{pmatrix}.$$

**c)** *Lagrange – orthogonal polynomials*: The Lagrange representation of the orthogonal polynomials is

$$P^T\boldsymbol{l}(x) = \boldsymbol{p}(x).$$

Recall that the coefficients $\boldsymbol{b}$ for the orthogonal basis are the solution of $P\boldsymbol{b} = \boldsymbol{f}$.

*2.2. Monomials – Newton*

Since both basis functions have the same degrees

$$\text{degree}(m_k(x)) = \text{degree}(\pi_k(x)) = k, \quad k = 0, \ldots, n$$

there must exist a lower triangular matrix $L$ such that

$$L\boldsymbol{\pi}(x) = \boldsymbol{m}(x).$$

By eliminating $\boldsymbol{l}$ in the two equations

$$V^T\boldsymbol{l} = \boldsymbol{m}, \quad U\boldsymbol{l} = \boldsymbol{\pi}$$

we get

$$V^TU^{-1}\boldsymbol{\pi} = \boldsymbol{m}$$

thus $V^TU^{-1} = L$ must be lower triangular and

$$V^T = LU. \tag{11}$$

*Equation (11) is a LU-decomposition of the transposed Vandermonde matrix.*

We can give an explicit expression for the lower triangular matrix $L$. Let $H_p(x_0, \ldots, x_k)$ be the sum of all homogeneous products of degree $p$ of the variables $x_0, \ldots, x_k$, e.g.

$$\begin{aligned} H_p(x_0) &= x_0^p \\ H_1(x_0, \ldots, x_k) &= \sum_{j=0}^{k} x_j \\ H_p(x_0, x_1) &= \sum_{j=0}^{p} x_0^j x_1^{p-j} = \sum_{j=0}^{p} H_j(x_0) H_{p-j}(x_1). \end{aligned}$$

For these functions Miller [3] shows that the recursion

$$H_p(x_0, \ldots, x_k) = \frac{H_{p+1}(x_0, \ldots, x_{k-1}) - H_{p+1}(x_1, \ldots, x_k)}{x_0 - x_k}$$

holds. Furthermore Miller also shows that the divided differences *eliminate coefficients* in the following sense. Let

$$f_i = P_n(x_i) = a_0 + a_1 x_i + \cdots a_n x_i^n$$

then

$$f[x_i, \ldots, x_{i+k}] = a_k + \sum_{j=k+1}^{n} a_j H_{j-k}(x_i, \ldots, x_{i+k}). \tag{12}$$

Thus $a_0, a_1, \ldots, a_{k-1}$ have been eliminated. From Equation (12) we immediately obtain the relation $L^T \boldsymbol{a} = \boldsymbol{d}$ where

$$L = \begin{pmatrix} 1 & & & & \\ H_1(x_0) & 1 & & & \\ H_2(x_0) & H_1(x_0, x_1) & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ H_n(x_0) & H_{n-1}(x_0, x_1) & \cdots & H_1(x_0, \ldots, x_{n-1}) & 1 \end{pmatrix}.$$

Since $\mathrm{diag}(L) = 1$, *Equation (11) is the standard LU-decomposition of $V^T$!* We have obtained the

**Theorem** Let $V^T = LU$ be the standard LU-decomposition of the transposed Vandermonde matrix. Then $L$ maps the Newton polynomials to the monomials and $U$ maps the Lagrange polynomials to the Newton polynomials.

If we solve $V^T \boldsymbol{l} = \boldsymbol{m}$ for $\boldsymbol{l}$ using Gaussian elimination then $L(U\boldsymbol{l}) = \boldsymbol{m}$ and we obtain as intermediate result of the forward-substitution in $L\boldsymbol{\pi} = \boldsymbol{m}$ the vector of the Newton-polynomials. By back-substitution in $U\boldsymbol{l} = \boldsymbol{\pi}$ we obtain the vector of the Lagrange polynomials. The connection between Newton form and Gauss elimination has already been observed by Carl de Boor [1] in one of the examples for his general expression for the inverse of a basis.

### *2.3. Monomials – orthogonal polynomials*

Because again the degrees are the same we conclude that there must exist a lower triangular matrix $C$ with

$$C\, \boldsymbol{p}(x) = \boldsymbol{m}(x).$$

Let $D = \mathrm{diag}\{\|p_0\|, \|p_1\|, \ldots, \|p_n\|\}$ and write this equation for $x = x_0, x_1, \ldots, x_n$. We obtain $CP^T = V^T$ or

$$V = PC^T = \underbrace{(PD^{-1})}_{Q}\underbrace{(DC^T)}_{R}$$

which is the $QR$-decomposition of the Vandermonde $V$! We obtained no explicit expressions for this decomposition. However, to compute the matrix $C$ we can proceed as follows: compute the QR-decomposition $V = QR$ and since $R = DC^T$

$$C = R^T D^{-1}.$$

Alternatively if $V$ and $P$ are known then $P^T V = D^2 C^T$ and

$$C = V^T P D^{-2}.$$

Because $P_n(x) = \boldsymbol{b}^T \boldsymbol{p}(x) = \boldsymbol{a}^T \boldsymbol{m}(x) = \boldsymbol{a}^T C\boldsymbol{p}(x)$ we get for the coefficients of both bases the relation

$$C^T \boldsymbol{a} = \boldsymbol{b}.$$

Thus we obtain the following result:

**Theorem** Let $D = \mathrm{diag}\{\|p_0\|, \|p_1\|, \ldots, \|p_n\|\}$ and $V = QR$ be the QR-decomposition of the Vandermonde matrix. Then the transformation matrix from the orthogonal basis to the monomial basis is given by $C = R^T D^{-1}$ and the coefficients are transformed by $C^T \boldsymbol{a} = \boldsymbol{b}$.

### *2.4. Newton – orthogonal polynomials*

We start with the general remark: consider the $LU$- and $QR$-decomposition of a non-singular $(n \times n)$ matrix $A$

$$A = LU = QR.$$

Then $U^T L^T = R^T Q^T$ and

$$R^{-T} U^T = Q^T L^{-T}. \tag{13}$$

Note that $R^{-T} U^T$ is lower triangular and the right hand side of Equation (13) is the $QR$-decomposition of this matrix.

In our case we have $V = QR = (PD^{-1})(DC^T)$ and $V^T = LU$ thus $V = U^T L^T$. Therefore Equation (13) becomes

$$C^{-1} L = P^T U^{-1}. \tag{14}$$

Since $L\boldsymbol{\pi}(x) = \boldsymbol{m} = C\boldsymbol{p}$ we get

$$C^{-1} L \boldsymbol{\pi} = \boldsymbol{p}.$$

Because of Equation (14) the transformation matrix is also given by $P^T U^{-1}$. For the coefficients we have the relation $U^T \boldsymbol{d} = \boldsymbol{f} = P\boldsymbol{b}$. Therefore

$$\boldsymbol{d} = U^{-T} P\boldsymbol{b} = (PU^{-1})^T \boldsymbol{b} = (C^{-1} L)^T \boldsymbol{b}.$$

## 3. Summary of the results

We have considered the interpolation polynomial represented in four bases:

$$P_n(x) = \boldsymbol{a}^T \boldsymbol{m}(x) = \boldsymbol{d}^T \boldsymbol{\pi}(x) = \boldsymbol{f}^T \boldsymbol{l}(x) = \boldsymbol{b}^T \boldsymbol{p}(x).$$

We obtained explicit expressions for the $LU$-decomposition of $V^T = LU$ and also an explicit expression for $U^{-1}$.

Let $D = \text{diag}\{\|p_0\|, \|p_1\|, \ldots, \|p_n\|\}$ and $V = QR$ be the QR-decomposition of the Vandermonde. Then $C = R^T D^{-1}$ and $P = QD$.

| Polynomials | Basis Transform | Transform of Coefficients |
|---|---|---|
| Lagrange/Monomials | $V^T \boldsymbol{l} = \boldsymbol{m}$ | $V\boldsymbol{a} = \boldsymbol{f}$ |
| Lagrange/Newton | $U\boldsymbol{l} = \boldsymbol{\pi}$ | $U^T \boldsymbol{d} = \boldsymbol{f}$ |
| Lagrange/O-Pol | $P^T \boldsymbol{l} = \boldsymbol{p}$ | $P\boldsymbol{b} = \boldsymbol{f}$ |
| Newton/Monomials | $L\boldsymbol{\pi} = \boldsymbol{m}$ | $L^T \boldsymbol{a} = \boldsymbol{d}$ |
| O-Pol/Monomials | $C\boldsymbol{p} = \boldsymbol{m}$ | $C^T \boldsymbol{a} = \boldsymbol{b}$ |
| Newton/O-Pol | $C^{-1} L\boldsymbol{\pi} = \boldsymbol{p}$ | $(C^{-1}L)^T \boldsymbol{b} = \boldsymbol{d}$ |

## REFERENCES

1. de Boor, C., *What is the Inverse of a Basis*, BIT 2001, Vol. 41, No. 5, pp. 880–890.
2. Gautschi W., *Numerical Analysis: An Introduction*, Birkhäuser, Boston, 1997
3. Miller, J. C. P. , *Neville's and Romberg's Processes: A fresh Appraisal with Extensions*, Philosophical Transactions of the Royal Society of London, Series A, Volume 263, 1968-1969.
4. Stoer, J., Bulirsch, R. *Introduction to Numerical Analysis*, 3rd ed., 2002.
5. Rutishauser, H., *Lectures on Numerical Mathematics*, Birkhäuser, 1990.