Chapter 7

Approximation Theory

The primary aim of a general approximation is to represent non-arithmetic quantities by arithmetic quantities so that the accuracy can be ascertained to a desired degree. Secondly, we are also concerned with the amount of computation required to achieve this accuracy. These general notions are applicable to functions $f(x)$ as well as to functionals $F(f)$ (A functional is a mapping from the set of functions to the set of real or complex numbers). Typical examples of quantities to be approximated are transcendental functions, integrals and derivatives of functions, and solutions of differential or algebraic equations. Depending upon the nature to be approximated, different techniques are used for different problems.

A complicated function $f(x)$ usually is approximated by an easier function of the form $\phi(x; a_0, \ldots, a_n)$ where $a_0, \ldots, a_n$ are parameters to be determined so as to characterize the best approximation of f. Depending on the sense in which the approximation is realized, there are three types of approaches:

1. Interpolatory approximation: The parameters a_i are chosen so that on a fixed prescribed set of points $x_i, i = 0, 1, \ldots, n$, we have

$$\phi(x_i; a_0, \ldots, a_n) = f(x_i) := f_i. \tag{7.1}$$

   Sometimes, we even further require that, for each i, the first r_i derivatives of ϕ agree with those of f at x_i.

2. Least-square approximation: The parameters a_i are chosen so as to

$$\text{Minimize}\|f(x) - \psi(x; a_0, \ldots, a_n)\|_2. \tag{7.2}$$

3. Min-Max approximation: the parameters a_i are chosen so as to minimize

$$\|f(x) - \phi(x; a_0, \ldots, a_n)\|_\infty. \tag{7.3}$$

Definition 7.0.1 *We say ϕ is a linear approximation of f if ϕ depends linearly on the parameters a_i, that is, if*

$$\phi(x_i; a_0, \ldots, a_n) = a_0\varphi_0(x) + \ldots + a_n\varphi(x_n) \tag{7.4}$$

where $\varphi_i(x)$ are given and fixed functions.

Choosing $\varphi_i(x) = x^i$, the approximating function ϕ becomes a polynomial. In this case, the theory for all the above three types of approximation is well established. The solution for the min-max approximation problem is the so called Chebyshev polynomial. We state without proof two fundamental results concerning the first two types of approximation:

Theorem 7.0.1 *Let $f(x)$ be a piecewise continuous function over the interval $[a, b]$. Then for any $\epsilon > 0$, there exist an integer n and numbers $a_0, \ldots, a_n$ such that $\int_a^b \{f(x) - \sum_{i=0}^{n} a_i x^i\}^2 dx < \epsilon$.*

Theorem 7.0.2 *(Weierstrass Approximation Theorem) Let $f(x)$ be a continuous function on $[a, b]$. For any $\epsilon > 0$, there exist an integer n and a polynomial $p_n(x)$ of degree n such that $\max_{x\in[a,b]} |f(x) - p_n(x)| < \epsilon$. In fact, if $[a, b] = [0, 1]$, then the Bernstein polynomial*

$$B_n(x) := \sum_{k=0}^{n} \binom{n}{k} x^k (1 - x)^{n-k} f(\frac{k}{n}) \tag{7.5}$$

converges to $f(x)$ as $n \to \infty$.

In this chapter, we shall consider only the interpolatory approximation. Choosing $\varphi_i(x) = x^i$, we have the so called polynomial interpolation; choosing $\varphi_i(x) = e^{ix}$, we have the so called trigonometric interpolation. The so called rational interpolation where

$$\varphi(x_i; a_0, \dots, a_n, b_0, \dots, b_m) = \frac{a_0\varphi_0(x) + \dots + a_n\varphi_n(x)}{b_0\varphi_0(x) + \dots + b_m\varphi_m(x)} \tag{7.6}$$

is an important non-linear interpolation.

7.1 Lagrangian Interpolation Formula

Theorem 7.1.1 *Let $f \in C[a,b]$. Let $x_i, i = 1, \dots, n$, be n distinct points in $[a,b]$,. There exists a unique polynomial $p(x)$ of degree $\leq n-1$ such that $p(x_i) = f(x_i)$. In fact,*

$$p(x) = \sum_{i=1}^{n} f(x_i)\ell_i(x) \tag{7.7}$$

where

$$\ell_i(x) := \prod_{\substack{j=1 \\ j\neq 1}}^{n} \frac{x - x_j}{x_i - x_j}. \tag{7.8}$$

In the case when $f \in C^n[a,b]$, then

$$E(x) := f(x) - p(x) = \frac{\prod_{j=1}^{n}(x - x_j)}{n!} f^{(n)}(\xi) \tag{7.9}$$

where $\min\{x_1, \dots, x_n, x\} < \xi < \max\{x_1, \dots, x_n, x\}$.

(pf): Suppose $p(x) = \sum_{k=0}^{n-1} a_k x^k$ where the coefficients a_k are to be determined. Then $p(x_i) = \sum_{k=0}^{n-1} a_k x_i^k = f(x_i), i = 1, \dots, n$, can be written in the form

$$\begin{bmatrix} 1, & x_1, \dots, & x_1^{n-1} \\ & & \\ 1, & x_n, & \dots, x_n^{n-1} \end{bmatrix} \begin{bmatrix} a_0 \\ \\ a_{n-1} \end{bmatrix} = \begin{bmatrix} f(x_1) \\ \\ f(x_n) \end{bmatrix}. \tag{7.10}$$

The matrix, known as the van Dermonde matrix, has determinant $\prod_{i>j}(x_i - x_j)$. Since all x_i's are distinct, we can uniquely solve (7.10) for the unknowns

$a_0, \ldots, a_{n-1}$. Note that each $\ell_i(x)$ is a polynomial of degree $n-1$ and $\ell_i(x_j) = \delta_{ij}$, the Kronecker delta notation. Therefore, by uniqueness, (7.7) is proved. Let $x_0 \in [a, b]$ and $x_0 \neq x_i$ for any $i = 1, \ldots, n$. Construct the C^n-function

$$F(x) = f(x) - p(x) - (f(x_0) - p(x_0)) \frac{\prod_{i=1}^{n}(x - x_i)}{\prod_{i=1}^{n}(x_0 - x_i)}.$$

It is easy to see that $F(x_i) = 0$ for $i = 0, \ldots, n$. By the Rolle's theorem, there exists ξ between $x_0, \ldots, x_n$ such that $F^{(n)}(\xi) = 0$. It follows that

$$f^{(n)}(\xi) - (f(x_0) - p(x_0)) \frac{n!}{\prod_{i=1}^{n}(x_0 - x_i)} = 0.$$

Thus $E(x_0) = f(x_0) - p(x_0) = \dfrac{\prod_{i=1}^{n}(x_0 - x_i)}{n!} f^{(n)}(\xi)$. Since x_0 is arbitrary, the theorem is proved. $\oplus$

Definition 7.1.1 *The polynomial $p(x)$ defined by (7.7) is called the Lagrange interpolation polynomial.*

Remark. The evaluation of a polynomial $p(x) = a_0 + a_1 x + \ldots + a_n x^n$ for $x = \xi$ may be done by the so called Horner scheme:

$$p(\xi) = (\ldots((a_n \xi + a_{n-1})\xi + a_{n-2})\xi \ldots + a_1)\xi + a_0 \tag{7.11}$$

which only takes n multiplications and n additions.

Remark. While theoretically important, Lagrange's formula is, in general, not efficient for applications. The efficiency is especially bad when new interpolating points are added, since then the entire formula is changed. In contrast, Newton's interpolation formula, being equivalent to the Lagrange's formula mathematically, is much more efficient.

Remark. Suppose polynomials are used to interpolate the function

$$f(x) = \frac{1}{1 + 25x^2} \tag{7.12}$$

in the interval $[-1, 1]$ at equally spaced points. Runge (1901) discovered that as the degree n of the interpolating polynomial $p_n(x)$ tends toward infinity, $p_n(x)$ diverges in the intervals $.726\ldots \leq |x| < 1$ while $p_n(x)$ works pretty well in the central portion of the interval.

7.2 Newton's Interpolation Formula

Interpolating a function by a very high degree polynomial is not advisable in practice. One reason is because we have seen the danger of evaluating high degree polynomials (e.g. the Wilkinson's polynomial and the Runge's function). Another reason is because local interpolation (as opposed to global interpolation) usually is sufficient for approximation.

One usually starts to interpolate a function over a smaller sets of support points. If this approximation is not enough, one then updates the current interpolating polynomial by adding in more support points. Unfortunately, each time the data set is changed Lagrange's formula must be entirely recomputed. For this reason, Newton's interpolating formula is preferred to Lagrange's interpolation formula.

Let $P_{i_0 i_1 \ldots i_k}(x)$ represent the k-th degree polynomial for which

$$P_{i_0 i_1 \ldots i_k}(x_{i_j}) = f(x_{i_j}) \tag{7.13}$$

for $j = 0, \ldots, k$.

Theorem 7.2.1 *The recursion formula*

$$p_{i_0 i_1 \ldots i_k}(x) = \frac{(x - x_{i_0}) P_{i_1 \ldots i_k}(x) - (x - x_{i_k}) P_{i_0 \ldots i_{k-1}}(x)}{x_{i_k} - x_{i_0}} \tag{7.14}$$

holds.

(pf): Denote the right-hand side of (7.14) by $R(x)$. Observe that $R(x)$ is a polynomial of degree $\leq k$. By definition, it is easy to see that $R(x_{i_j}) = f(x_{i_j})$ for all $j = 0, \ldots, k$. That is, $R(x)$ interpolates the same set of data as does the polynomial $P_{i_0 i_1 \ldots i_k}(x)$. By Theorem 7.1.1 the assertion is proved. $\oplus$

The difference $P_{i_0 i_1 \dots i_k}(x) - P_{i_0 i_1 \dots i_{k-1}}(x)$ is a k-th degree polynomial which vanishes at x_{i_j} for $j = 0, \dots, k-1$. Thus we may write

$$P_{i_0 i_1 \dots i_k}(x) = P_{i_0 i_1 \dots i_{k-1}}(x) + f_{i_0 \dots i_k}(x - x_{i_0})(x - x_{i_1}) \dots (x - x_{i_{k-1}}). \quad (7.15)$$

The leading coefficients $f_{i_0 \dots i_k}$ can be determined recursively from the formula (7.14), i.e.,

$$f_{i_0 \dots i_k} = \frac{f_{i_1 \dots i_k} - f_{i_0 \dots i_{k-1}}}{x_{i_k} - x_{i_0}} \quad (7.16)$$

where $f_{i_1 \dots i_k}$ and $f_{i_0 \dots i_{k-1}}$ are the leading coefficients of the polynomials $P_{i_1 \dots i_k}(x)$ and $P_{i_0 \dots i_{k-1}}(x)$, respectively.

Remark. Note that the formula (7.16) starts from $f_{i_0} = f(x_{i_0})$.

Remark. The polynomial $P_{i_0 \dots i_k}(x)$ is uniquely determined by the set of support data $\{(x_{i_j}, f_{i_j})\}$. The polynomial is invariant to any permutation of the indices $i_0, \dots, i_k$. Therefore, the divided differences (7.16) are invariant to permutation of the indices.

Definition 7.2.1 *Let $x_0, \dots, x_k$ be support arguments (but not necessarily in any order) over the interval $[a, b]$. We define the Newton's divided difference as follows:*

$$f[x_0] : = f(x_0) \quad (7.17)$$

$$f[x_0, x_1] : = \frac{f[x_1] - f[x_0]}{x_1 - x_0} \quad (7.18)$$

$$f[x_0, \dots, x_k] : = \frac{f[x_1, \dots, x_k] - f[x_0, \dots, x_{k-1}]}{x_k - x_0} \quad (7.19)$$

It follows that the k-th degree polynomial that interpolates the set of support data $\{(x_i, f_i) | i = 0, \dots, k\}$ is given by

$$\begin{aligned} P_{x_0 \dots x_k}(x) &= f[x_0] + f[x_0, x_1](x - x_0) \quad (7.20) \\ &+ \dots + f[x_0, \dots, x_k](x - x_0)(x - x_1) \dots (x - x_{k-1}). \end{aligned}$$

7.3 Osculatory Interpolation

Given $\{x_i\}, i = 1, \dots k$ and values $a_i^{(0)}, \dots, a_i^{(r_i)}$ where r_i are nonnegative integers. We want to construct a polynomial $P(x)$ such that

$$P^{(j)}(x_i) = a_i^{(j)} \tag{7.21}$$

for $i = 1, \dots, k$ and $j = 0, \dots, r_i$. Such a polynomial is said to be an osculatory interpolating polynomial of a function f if $a_i^{(j)} = f^{(j)}(x_i) \dots$

Remark. The degree of $P(x)$ is at most $\sum_{i=1}^{k}(r_i + 1) - 1$.

Theorem 7.3.1 *Given the nodes $\{x_i\}, i = 1, \dots, k$ and values $\{a_i^{(j)}\}, j = 0, \dots, r_i$, there exists a unique polynomial satisfying (7.21).*

(pf): For $i = 1, \dots, k$, denote

$$\begin{aligned} q_i(x) &= c_i^{(0)} + c_i^{(1)}(x - x_i) + \dots + c_i^{(r_i)}(x - x_i)^{r_i} && (7.22)\\ P(x) &= q_1(x) + (x - x_1)^{r_1+1}q_2(x) + \dots && (7.23)\\ &+ (x - x_1)^{r_1+1}(x - x_2)^{r_2+1} \dots (x - x_{k-1})^{r_{k-1}+1}q_k(x). \end{aligned}$$

Then $P(x)$ is of degree $\leq \sum_{i=1}^{k}(r_i + 1) - 1$. Now $P^{(j)}(x_1) = a_1^{(j)}$ for $j = 0, \dots, r_1$ implies $a_1^{(0)} = c_1^{(0)}, \dots, a_1^{(j)} = c_1^{(j)}j!$. So $q_1(x)$ is determined with $c_1^{(j)} = \frac{a_1^{(j)}}{j!}$. Now we rewrite (7.23) as

$$R(x) := \frac{P(x) - q_1(x)}{(x - x_1)^{r_1+1}} = q_2(x) + (x - x_2)^{r_2+1}q_3(x) + \dots$$

Note that $R^{(j)}(x_2)$ are known for $j = 0, \dots, r_2$ since $P^{(j)}(x_2)$ are known. Thus all $c_2^{(j)}$, hence $q_2(x)$, may be determined. This procedure can be continued to determine all $q_i(x)$. Suppose $Q(x) = P_1(x) - P_2(x)$ where $P_1(x)$ and $P_2(x)$ are two polynomials of the theorem. Then $Q(x)$ is of degree $\leq \sum_{i=1}^{k}(r_i + 1) - 1$, and has zeros at x_i with multiplicity $r_i + 1$. Counting multiplicities, $Q(x)$ has $\sum_{i=1}^{k}(r_i + 1)$ zeros. This is possible only if $Q(x) \equiv 0$.

Examples. (1) Suppose $k = 1, x_1 = a, r_1 = n-1$, then the polynomial (7.23) becomes $P(x) = \sum_{j=0}^{n-1} f^{(j)}(a)\frac{(x-a)^j}{j!}$ which is the Taylor's polynomial of f at $x = x_1$.

(2) Suppose $r_i = 1$ for all $i = 1, \ldots, k$. That is, suppose values of $f(x_i)$ and $f'(x_i)$ are to be interpolated. Then the resultant $(2k-1)$-degree polynomial is called the Hermite interpolating polynomial. Recall that the $(k-1)$-degree polynomial

$$\ell_i(x) = \prod_{\substack{j=1 \\ j\neq i}}^{k} \frac{x - x_j}{x_i - x_j} \tag{7.24}$$

has the property

$$\ell_i(x_j) = \delta_{ij}. \tag{7.25}$$

Define

$$\begin{aligned} h_i(x) &= [1 - 2(x - x_i)\ell_i'(x_i)]\ell_i^2(x) && (7.26)\\ g_i(x) &= (x - x_i)\ell_i^2(x). && (7.27) \end{aligned}$$

Note that both $h_i(x)$ and $g_i(x)$ are of degree $2k-1$. Furthermore,

$$\begin{aligned} h_i(x_j) &= \delta_{ij};\\ g_i(x_j) &= 0; && (7.28)\\ h_i'(x_j) &= [1 - 2(x - x_i)\ell_i'(x_i)]2\ell_i(x)\ell_i'(x) - 2\ell_i'(x_i)\ell_i^2(x)|_{x=x_j} = 0;\\ g_i'(x_j) &= (x - x_i)2\ell_i(x)\ell_i'(x) + \ell_i^2(x)|_{x=x_j} = \delta_{ij}. \end{aligned}$$

So the Hermite interpolating polynomial can be written down as

$$P(x) = \sum_{i=1}^{k} f(x_i)h_i(x) + f'(x_i)g_i(x)). \tag{7.29}$$

(3) Suppose $r_i = 0$ for all i. Then the polynomial becomes $P(x) = c_1 + c_2(x - x_1) + \ldots + c_k(x - x_1)\ldots(x - x_{k-1})$ which is exactly the Newton's formula.

7.4 Spline Interpolation

Thus far for a given function f of an interval $[a, b]$, the interpolation has been to construct a polynomial over the entire interval $[a, b]$. There are at least two disadvantages for the global approximation:

1. For better accuracy, we need to supply more support data. But then the degree the resultant polynomial gets higher and such a polynomial is difficult to work with.
2. Suppose f is not smooth enough. Then the error estimate of an high degree polynomial is difficult to establish. In fact, it is not clear whether or not that the accuracy will increase with increasing number of support data.

As an alternative way of approximation, the spline interpolation is a local approximation of a function f, which, nonetheless, yields global smooth curves and is less likely to exhibit the large oscillation characteristic of high-degree polynomials. (Ref: A Practical Guide to Splines, Springer-Verlga, 1978, by C. de Boor).

We demonstrate the idea of cubic spline as follows.

Definition 7.4.1 *Let the interval $[a, b]$ be partitioned into $a = x_1 < x_2 < \ldots < x_n = b$. A function $p(x)$ is said to be a cubic plite of f on the partition if*

1. The restriction of $p(x)$ on each subinterval $[x_i, x_{i+1}], i = 1, \ldots, n-1$ is a cubic polynomial;
2. $p(x_i) = f(x_i), i = 1, \ldots, n$;
3. $p'(x_i)$ and $p''(x_i)$ are continuous at each $x_i, i = 2, \ldots, n-1$.

Since there are $n-1$ subintervals, condition (1) requires totally $4(n-1)$ coefficients to be determined. Condition (2) is equivalent to $2(n-2)+2$ equations. Condition (3) is equivalent to $(n-2)+(n-2)$ equations. Thus we still need two more conditions to completely determine the cubic spline.

Definition 7.4.2 *A cubic spline $p(x)$ of f is said to be*

1. *A clamped spline if $p'(x_1)$ and $p'(x_n)$ are specified.*
2. *A natural spline if $p''(x_1) = 0$ and $p''(x_n) = 0$.*
3. *A periodic spline if $p(x_1) = p(x_n), p'(x_1) = p'(x_n)$ and $p''(x_1) = p''(x_n)$.*

Denote $M_i := p''(x_i), i = 1, \ldots, n$. Since $p(x)$ is piecewise cubic and continuously differentiable, $p''(x)$ is piecewise linear and continuous on $[a, b]$. In particular, over the interval $[x_i, x_{i+1}]$, we have

$$p''(x) = M_i \frac{x - x_{i+1}}{x_i - x_{i+1}} + M_{i+1} \frac{x - x_i}{x_{i+1} - x_i}. \tag{7.30}$$

Upon integrating $p''(x)$ twice, we obtain

$$p(x) = \frac{(x_{i+1} - x)^3 M_i + (x - x_i)^3 M_{i+1}}{6\delta_i} + c_i(x_{i+1} - x) + d_i(x - x_i) \tag{7.31}$$

where $\delta_i := x_{i+1} - x_i$, and c_i and d_i are integral constants. By setting $p(x_i) = f(x_i), p(x_{i+1}) = f(x_{i+1})$, we get

$$c_i = \frac{f(x_i)}{\delta_i} - \frac{\delta_i M_i}{6} \tag{7.32}$$

$$d_i = \frac{f(x_{i+1})}{\delta_i} - \frac{\delta_i M_{i+1}}{6}. \tag{7.33}$$

Thus on $[x_i, x_{i+1}]$,

$$\begin{aligned} p(x) = & \frac{(x_{i+1} - x)^3 M_i + (x - x_i)^3 M_{i+1}}{6\delta_i} \\ & + \frac{(x_{i+1} - x) f(x_i) + (x - x_i) f(x_{i+1})}{\delta_i} \\ & - \frac{\delta_i}{6}[(x_{i+1} - x) M_i + (x - x_i) M_{i+1}]. \end{aligned} \tag{7.34}$$

It only remains to determine M_i. We first use the continuity condition of $p'(x)$ at x_i for $i = 2, \ldots, n-1$, that is,

$$\lim_{x \to x_i^-} p'(x) = \lim_{x \to x_i^+} p'(x). \tag{7.35}$$

Thus

$$\begin{aligned} & \frac{3\delta_{i-1}^2 M_i}{6\delta_{i-1}} + \frac{f(x_i) - f(x_{i-1})}{\delta_{i-1}} - \frac{\delta_{i-1}(M_i - M_{i-1})}{6} \\ = & \frac{-3\delta_i^2 M_i}{6\delta_i} + \frac{f(x_{i+1}) - f(x_i)}{\delta_i} - \frac{\delta_i(M_{i+1} - M_i)}{6}, \end{aligned} \tag{7.36}$$

or equivalently,

$$\begin{aligned} & \frac{\delta_{i-1}}{6} M_{i-1} + \frac{\delta_i + \delta_{i-1}}{3} M_i + \frac{\delta_i}{6} M_{i+1} \\ = & \frac{f(x_{i+1}) - f(x_i)}{\delta_i} - \frac{f(x_i) - f(x_{i-1})}{\delta_{i-1}}. \end{aligned} \tag{7.37}$$

Note that we have $n-2$ equations in n unknowns. Suppose, for example, we work with the clamped spline, that is,

$$\begin{aligned} p'(x_1) &= f'(x_1) && (7.38) \\ p'(x_n) &= f'(x_n). && (7.39) \end{aligned}$$

Then we have

$$\begin{aligned} \frac{\delta_1}{3}M_1 + \frac{\delta_1}{6}M_2 &= \frac{f(x_2)-f(x_1)}{\delta_1} - f'(x_1) && (7.40) \\ \frac{\delta_{n-1}}{6}M_{n-1} + \frac{\delta_{n-1}}{3}M_n &= f'(x_n) - \frac{f(x_n)-f(x_{n-1})}{\delta_{n-1}}. && (7.41) \end{aligned}$$

In matrix form, we obtain a linear algebraic equation

$$\begin{bmatrix} \frac{\delta_1}{3} & \frac{\delta_1}{6} & 0 & & & \\ \frac{\delta_1}{6} & \frac{\delta_1+\delta_2}{3} & \frac{\delta_2}{6} & 0 & & \\ 0 & & & & & \\ & \ddots & \ddots & \ddots & & \\ & & & \frac{\delta_{n-2}}{6} & \frac{\delta_{n-2}+\delta_{n-1}}{3} & \frac{\delta_{n-1}}{6} \\ & & & 0 & \frac{\delta_{n-1}}{6} & \frac{\delta_{n-1}}{3} \end{bmatrix} \begin{bmatrix} M_1 \\ M_2 \\ \\ \vdots \\ M_n \end{bmatrix}$$

$$= \begin{bmatrix} \frac{f(x_2)-f(x_1)}{\delta_1} - f'(x_1) \\ vdots \\ \frac{f(x_{i+1})-f(x_i)}{\delta_i} - \frac{f(x_i)-f(x_{i-1})}{\delta_{i-1}} \\ \vdots \\ f'(x_n) - \frac{f(x_n)-f(x_{n-1})}{\delta_{n-1}} \end{bmatrix}.$$

We note that in (7.42) the coefficient matrix is real, symmetric, tridiagonal and strictly diagonally dominant. Therefore, there is a unique solution for $M_i, i=1,\ldots,n$.

7.5 Trigonometric Interpolation

For a given set of N support points $(x_k, f_k), k=0,\ldots,N-1$, we consider linear interpolation of the following forms:

$$\begin{aligned} \phi(x) &= \frac{a_0}{2} + \sum_{h=1}^{M}(a_h \cos hx + b_h \sin hx) && (7.42) \\ \phi(x) &= \frac{a_0}{2} + \sum_{h=1}^{M-1}(a_h \cos hx + b_h \sin hx) + \frac{a_M}{2}\cos Mx, && (7.43) \end{aligned}$$

depending upon whether $N = 2M + 1$ or $N = 2M$.

For simplicity, we shall consider equally spaced notes. Without loss of generality, we shall assume

$$x_k = \frac{2\pi k}{N}, k = 0, \ldots, N-1.$$

Observe that

$$e^{-i(hx_k)} = e^{-i(2\pi hk/N)} = e^{i2\pi(N-h)k/N} = e^{i(N-h)x_k}.$$

Thus, we may write

$$\begin{aligned} \cos hx_k &= \frac{e^{ihx_k} + e^{i(N-h)x_k}}{2}; \\ \sin hx_k &= \frac{e^{ihx_k} - e^{i(N-h)x_k}}{2i}. \end{aligned} \tag{7.44}$$

Upon substitution into (7.42) and (7.43), we obtain

$$\begin{aligned} \phi(x_k) &= \frac{a_0}{2} + \sum_{h=1}^{M}(a_h \frac{e^{ihx_k} + e^{i(N-h)x_k}}{2} + b_h \frac{e^{ihx_k} - e^{i(N-h)x_k}}{2i}) \\ &= \frac{a_0}{2} + \sum_{h=1}^{M}(\frac{a_h - ib_h}{2} e^{ihx_k} + \frac{a_h + ib_h}{2} e^{i(N-h)x_k}) \\ &= \beta_0 + \beta_1 e^{ix_k} + \ldots + \beta_{2M} e^{i2Mx_k}, \qquad (7.45) \\ \phi(x_k) &= \frac{a_0}{2} + \sum_{h=1}^{M-1}(\frac{a_h - ib_h}{2} e^{ihx_k} + \frac{a_h + ib_h}{2} e^{i(N-h)x_k}) \\ &+ \frac{a_M}{2} \frac{e^{iMx_k} + e^{i(N-M)x_k}}{2} \\ &= \beta_0 + \beta_1 e^{ix_k} + \ldots + \beta_{2M-1} e^{i(2M-1)x_k}, \qquad (7.46) \end{aligned}$$

respectively. Thus instead of considering the trigonometric expressions $\phi(x)$, we are motivated to consider the phase polynomial

$$p(x) := \beta_0 + \beta_1 e^{ix} + \ldots + \beta_{N-1} e^{i(N-1)x}, \tag{7.47}$$

or equivalently, by setting $\omega := e^{ix}$, the standard polynomial

$$P(\omega) := \beta_0 + \beta_1 \omega + \ldots + \beta_{N-1} \omega^{N-1}. \tag{7.48}$$

Denoting $\omega_k := e^{ix_k}$, the interpolating condition becomes

$$P(\omega_k) = f_k, k = 0, \ldots, N-1. \tag{7.49}$$

Since all ω_k are distinct, by Theorem 7.1.1, we know

Theorem 7.5.1 *For any support data* $(x_k, f_k), k = 0, \ldots, N-1$ *with* $x_k = 2\pi k/N$, *there exists a unique phase polynomial* $p(x)$ *of the form (7.47) such that* $p(x_k) = f_k$ *for* $k = 0, \ldots, N-1$. *In fact,*

$$\beta_j = \frac{1}{N}\sum_{k=0}^{N-1} f_k(\omega_k)^{-j} = \frac{1}{N}\sum_{k=0}^{N-1} f_k(e^{ix_k})^{-j}. \tag{7.50}$$

(pf): It only remains to show (7.50). Recall that $\omega_k = e^{ix_k} = e^{i\frac{2\pi k}{N}}$. Thus $\omega_k^j = \omega_j^k$ and $\omega_k^{-j} = \overline{\omega_k^j}$. Observe that $\omega_{j-h}^N = 1$. That is, ω_{j-h} is a root of the polynomial $\omega^N - 1 = (\omega - 1)\sum_{k=0}^{N-1}\omega^k$. Thus it must be either $\omega_{j-h} = 1$ which is the case $j = h$, or $\sum_{k=0}^{N-1}\omega_{j-h}^k = \sum_{k=0}^{N-1}\omega_k^{j-h} = \sum_{k=0}^{N-1}\omega_k^j\omega_k^{-h} = 0$. We may therefore summarize that

$$\sum_{k=0}^{N-1}\omega_k^j\omega_k^{-h} = \begin{cases} 0, & \text{if} \quad j \neq 0 \\ N, & \text{if} \quad j = h. \end{cases} \tag{7.51}$$

Introducing $\omega^{(h)} := (1, \omega_1^h, \ldots, \omega_{N-1}^h)^T \in C^N$, we may rewrite (7.51) in terms of the complex inner product

$$\langle \omega^{(j)}, \omega^{(h)} \rangle = \begin{cases} 0, & \text{if} \quad j \neq h \\ & \qquad\qquad . \\ N, & \text{if} \quad j = h \end{cases} \tag{7.52}$$

That is, the vectors $\omega^{(0)}, \ldots \omega^{(N-1)}$ form an orthogonal basis of C^N. Denote $f := (f_0, \ldots, f_{N-1})^T$.

Then the interpolating condition (7.49) can be written as

$$\begin{bmatrix} 1 & 1 & & 1 \\ 1 & \omega_1 & \ldots & \omega_1^{N-1} \\ & \vdots & & \\ 1 & \omega_{N-1} & & \omega_{N-1}^{N-1} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \\ \\ \beta_{N-1} \end{bmatrix} = \begin{bmatrix} f_0 \\ \\ \\ f_{N-1} \end{bmatrix}$$

or simply

$$\beta_0\omega^{(0)} + \beta_1\omega^{(1)} + \ldots + \beta_{N-1}\omega^{(N-1)} = f. \tag{7.53}$$

By the orthogonality of $\omega^{(h)}$, (7.50) follows.

Corollary 7.5.1 *The trigonometric interpolation of the data set* $(x_k, f_k), k = 0, \ldots, N-1$ *with* $x_k = 2\pi k/N$ *is given by (7.42) or (7.43) with*

$$a_h = \frac{2}{N}\sum_{k=0}^{N-1} f_k \cos hx_k \tag{7.54}$$

$$b_h = \frac{2}{N}\sum_{k=0}^{N-1} f_k \sin hx_k. \tag{7.55}$$

Definition 7.5.1 *Given the phase polynomial (7.47) and* $0 \le s \le N$*, the s-segment* $p_s(x)$ *is defined to be*

$$p_s(x) := \beta_0 + \beta_1 e^{ix} + \ldots + \beta_s e^{isx}. \tag{7.56}$$

Theorem 7.5.2 *Let* $p(x)$ *be the phase polynomial that interpolates a given set of support data* $(x_k, f_k), k = 0, \ldots, N-1$ *with* $x_k = 2\pi k/N$*. Then the s-segment* $p_s(x)$ *of* $p(x)$ *minimizes the sum*

$$S(q) = \sum_{k=0}^{N-1} |f_k - q(x_k)|^2 \tag{7.57}$$

over all phase polynomials $q(x) = \gamma_0 + \gamma_1 e^{ix} + \ldots + \gamma_s e^{isx}$*.*

(pf): Introducing the vectors $p_s := (p_s(x_0), \ldots p_s(x_{N-1}))^T = \sum_{j=0}^{s} \beta_j \omega^{(j)}$ and $q := (q(x_0), \ldots, q(x_{N-1}))^T = \sum_{j=0}^{s} \gamma_j \omega^{(j)}$ in C^N, we write $S(q) = \langle f - q, f - q\rangle$. By theorem 7.5.1, we know $\beta_j = \frac{1}{N}\langle f, w^{(j)}\rangle$. Thus for $j \le s$, we have

$$\frac{1}{N}\langle f - p_s, w^{(j)}\rangle = \beta_j - \beta_j = 0,$$

and hence

$$\langle f - p_s, p_s - q\rangle = \sum_{j=1}^{s}\langle f - p_s, (\beta_j - \gamma_j)\omega^{(j)}\rangle = 0.$$

It follows that $S(q) = \langle f - q, f - q\rangle = \langle f - p_s + p_s - q\rangle = \langle f - p_s, f - p_s\rangle + \langle p_s - q, p_s - q\rangle \ge \langle f - p_s, f - p_s\rangle = S(p_s)$.

Remark. (7.5.2) states the important property that the truncated trigonometric interpolation polynomial $p(x)$ produces the least-squares trigonometric approximation $p_s(x)$ of all data.

7.6 Fast Fourier Transform

Suppose that we have a series of sines and cosines which represent a given function of $[-L, L]$, say,

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos\frac{n\pi x}{L} + b_n \sin\frac{n\pi x}{L}. \tag{7.58}$$

Using the facts that

$$\int_{-L}^{L} \cos\frac{n\pi x}{L} dx = \int_{-L}^{L} \sin\frac{n\pi x}{L} dx = 0 \tag{7.59}$$

$$\int_{-L}^{L} \cos\frac{n\pi x}{L} \cos\frac{m\pi x}{L} dx = \begin{cases} 0, & \text{if } n \neq m \\ L, & \text{if } n = m \end{cases} \tag{7.60}$$

and

$$\int_{-L}^{L} \sin\frac{n\pi x}{L} \cos\frac{m\pi x}{L} dx = 0, \tag{7.61}$$

one can show that

$$a_n = \frac{1}{L}\int_{-L}^{L} f(x) \cos\frac{n\pi x}{L} dx, n = 0, 1, \ldots \tag{7.62}$$

$$b_n = \frac{1}{L}\int_{-L}^{L} f(x) \sin\frac{n\pi x}{L} dx, n = 1, 2, \ldots \tag{7.63}$$

Definition 7.6.1 *The series (7.58), with coefficients defined by (7.62) and (7.63), is called the fourier series of $f(x)$ on $[-L, L]$.*

Given a function $f(x)$, its Fourier series does not necessarily converge to $f(x)$ at every x. In fact,

Theorem 7.6.1 *Suppose $f(x)$ is piecewise continuous on $[-L, L]$. Then (1) If $x_0 \in (-L, L)$ and $f'(x_0^+)$ and $f'(x_0^-)$ both exist (where $f'(x^\pm) := \lim_{h\to 0^\pm} \frac{f(x_0+h) - f(x_0)}{h}$), then the Fourier series converges to $\frac{f(x_0^+)+f(x_0^-)}{2}$. (2) At $-L$ or L, if $f'(-L^+)$ and $f'(L^-)$ exist, then the Fourier series converges to $\frac{f(-L^+)+f(L^-)}{2}$.*

Remark. Suppose that $f(x)$ is defined on $[0, L]$. We may extend $f(x)$ to become an even function on $[-L, L]$ (simply by defining $f(x) := f(-x)$ for $x \in [-L, 0]$). In this way, the Fourier coefficients for the extended function become

$$\begin{aligned} a_n &= \frac{2}{L}\int_0^L f(x)\cos\frac{n\pi x}{L}dx \\ b_n &\equiv 0. \end{aligned} \tag{7.64}$$

Thus we are left with a pure cosine series

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos\frac{n\pi x}{L}. \tag{7.65}$$

Similarly, we may extend $f(x)$ to become an odd function on $[-L, L]$, in which case we obtain a pure sine series

$$f(x) = \sum_{n=1}^{\infty} b_n \sin\frac{n\pi x}{L} \tag{7.66}$$

with

$$b_n = \frac{2}{L}\int_0^L f(x)\sin\frac{n\pi x}{L}dx. \tag{7.67}$$

Example.

1. If $f(x) = |x|$, $-\pi \le x \le \pi$, then

$$f(x) = \frac{\pi}{2} - \frac{4}{\pi}(\cos x + \frac{\cos 3x}{3^2} + \frac{\cos 5x}{5^2} + \ldots).$$

2. If $g(x) = x, -\pi \le x \le \pi$, then

$$f(x) = 2(\sin x - \frac{\sin 2x}{2} + \frac{\sin 3x}{3} - \frac{\sin 4x}{4} + \ldots).$$

3. If

$$h(x) = \begin{cases} x(\pi - x), & \text{for} \quad 0 \le x \le \pi \\ x(\pi + x), & \text{for} \quad -\pi \le x \le 0 \end{cases}$$

then

$$h(x) = \frac{8}{\pi}(\sin x + \frac{\sin 3x}{3^3} + \frac{\sin 5x}{5^3} + \ldots).$$

Remark. The relationship between the trigonometric interpolation and the Fourier series of a function $f(x)$ can easily be established. For demonstration purpose, we assume $f(x)$ is defined over $[0, 2\pi]$. Define $g(y) = f(y+\pi) = f(x)$ for $y \in [-\pi, \pi]$. Then a typical term in (7.58) for $g(y)$ is $a_n \cos ny$ with

$$a_n = \frac{1}{\pi}\int_{-\pi}^{\pi} g(y)\cos(ny)dy. \tag{7.68}$$

Suppose we partition the interval $[-\pi, \pi]$ into N equally spaced subinterval and define $y_k = -\pi + \frac{2\pi k}{N}$ for $k = 0, \ldots, N-1$. Then the approximation to (7.68) is

$$\begin{aligned} a_n &\approx \frac{2}{N}\sum_{k=0}^{N-1} g(y_k)\cos(ny_k) \\ &= (-1)^n \frac{2}{N}\sum_{k=0}^{N-1} f(\frac{2\pi k}{N})\cos\frac{2\pi kn}{N} := \hat{a}_n. \end{aligned} \tag{7.69}$$

Now observe that

$$\begin{aligned} f(x) = g(y) &= \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos ny + b_n \sin ny \\ &\approx \frac{a_0}{2} + \sum_{n=1}^{M} a_n \cos ny + b_n \sin ny \\ &\approx \frac{\hat{a}_0}{2} + \sum_{n=1}^{M} \hat{a}_n \cos n(x-\pi) + \hat{b}_n \sin n(x-\pi). \end{aligned}$$

Comparing (7.69) with (7.54), we realize that the trigonometric interpolation polynomial converges to the Fourier series as $N \to \infty$. Thus the trigonometric interpolation can be interpreted as the Fourier analysis applied to discrete data.

Remark. The basis of the Fourier analysis method for smoothing data is as follows: If we think of given numerical data as consisting of the true values of a function with random errors superposed, the true functions being relatively smooth and the superposed errors quite unsmooth, then the examples above suggest a way of partially separating functions from error. Since the true function is smooth, its Fourier coefficients will decrease quickly. But the unsmoothness of the error suggests that its Fourier coefficients may decrease very slowly, if at all. The combined series will consist almost entirely of error, therefore, beyond a certain place. If we simply truncate the series at the right place, then we are discarding mostly error (although there will be error contribution in the terms retained).

Remark. Since truncation produces a least-squares approximation (See (7.5.2), the fourier analysis method may be viewed as least-squares smoothing.

We now derive the Fourier series in complex form.

Lemma 7.6.1 *The functions* $e^{i\frac{j\pi x}{L}}$ *and* $e^{i\frac{k\pi x}{L}}$ *are orthogonal in the following sense:*

$$\int_{-L}^{L} e^{i\overline{\frac{j\pi x}{L}}} e^{i\frac{k\pi x}{L}} = \begin{cases} 0 & \textit{if } k \neq j \\ 2L, & \textit{if } k = j \end{cases}. \tag{7.70}$$

Assume the Fourier series takes the form

$$f(x) = \sum_{n=-\infty}^{\infty} f_n e^{i\frac{n\pi x}{L}}. \tag{7.71}$$

Multiplying both sides of (7.71) by $e^{-i\frac{k\pi x}{L}}$ and integrating brings

$$\int_{-L}^{L} f(x) e^{-i\frac{k\pi x}{L}} dx = \sum_{n=-\infty}^{\infty} f_n \int_{-L}^{L} e^{i\frac{n\pi x}{L}} e^{i\frac{k\pi x}{L}} dx. \tag{7.72}$$

By the orthogonality property, it is therefore suggested that

$$f_k = \frac{1}{2L} \int_{-L}^{L} f(x) e^{i\frac{k\pi x}{L}} dx. \tag{7.73}$$

Remark. The complex form (7.71) can be written as

$$\begin{aligned} \sum_{n=-\infty}^{\infty} f_n e^{i\frac{n\pi x}{L}} &= \sum_{n=-\infty}^{\infty} f_n (\cos\frac{n\pi x}{L} + i \sin\frac{n\pi x}{L}) \\ &= f_0 + \sum_{n=1}^{\infty} (f_n + f_{-n}) \cos\frac{n\pi x}{L} + i(f_n - f_{-n}) \sin\frac{n\pi x}{L}. \end{aligned} \tag{7.74}$$

It is easy to see that $f_0 = \frac{a_0}{2}$, $f_n + f_{-n} = a_n$ and $f_n + f_{-n} = b_n$. That is, the series (7.71) is precisely the same as the series (7.58).

Remark. We consider the relation between the trigonometric interpolation and the fourier series again. Let $f(x)$ be a function defined on $[0, 2\pi]$. Then the Fourier series of f may be written in the form

$$f(x) = \sum_{n=-\infty}^{\infty} f_n e^{inx} \tag{7.75}$$

where

$$f_n = \frac{1}{2\pi}\int_0^{2\pi} f(x)e^{-inx}dx. \tag{7.76}$$

Consider the function $g(x)$ of the form

$$g(x) = \sum_{j=-\ell}^{\ell} d_j e^{ijx} \tag{7.77}$$

for $x \in [0, 2\pi]$. Suppose $g(x)$ interpolates $f(x)$ at $x = x_k = \frac{2\pi n}{N}$ for $k = 0, 1, \ldots, N-1$. Multiplying both sides of (7.77) by e^{-inx_k} and sum over k, we obtain

$$\sum_{k=0}^{N-1} f(x_k)e^{-inx_k} = \sum_{k=0}^{N-1}\sum_{j=-\ell}^{\ell} d_j e^{ijx_k}e^{-inx_k} \tag{7.78}$$

$$= \sum_{j=-\ell}^{\ell} d_j (\sum_{k=0}^{N-1} e^{ijx_k}e^{-inx_k}). \tag{7.79}$$

We recall from (7.51) the orthogonality in the second summation of (7.74). Therefore,

$$d_n = \frac{1}{N}\sum_{k=0}^{N-1} f(x_k)e^{-inx_k} \tag{7.80}$$

for $n = 0, \pm 1, \ldots \pm \ell$. Once, we see that (7.80) is a Trapezoidal approximation of the integral (7.76). The match between (7.49) and (7.80) is conspicuous except that the ranges of validity do not coincide. Consider the case where $N = 2\ell + 1$. Then obviously $\beta_j = d_j$ for $j = 0, 1, \ldots, \ell$. But $\beta_{N+j} = \frac{1}{N}\sum_{k=0}^{N-1} f(x_k)e^{-i(N+j)x_k} = \frac{1}{N}\sum_{k=0}^{N-1} f(x_k)e^{-ijx_k} = d_j$ for $j = -1, \ldots, -\ell$.

The central idea behind the fast Fourier transform (FFT) is that when N is the product of integers, the numbers d_j (or β_j) prove to be closely interdependent. This interdependence can be exploited to substantially reduce the amount of computation required to generated these numbers. We demonstrate the idea as follows:

Suppose $N = t_1 t_2$ where both t_1 and t_2 are integers. Let

$$\begin{aligned} j: &= j_1 + t_1 j_2 \\ n: &= n_2 + t_2 n_1 \end{aligned}$$

for $j_1, n_1 = 0, 1, \ldots, t_1 - 1$ and $j_2, n_2 = 0, 1, \ldots, t_2 - 1$. Note that both j and n run their required ranges 0 to $N-1$. Let $\omega := e^{-i\frac{2\pi}{N}}$. Then $\omega^N = 1$. Thus

$$\beta_j = \beta_{j_1+t_1j_2} = \frac{1}{N}\sum_{n=0}^{N-1} f(x_n)e^{-i\frac{2\pi jn}{N}} = \frac{1}{N}\sum_{n=0}^{N-1} f(x_n)\omega^{nj}$$

$$= \frac{1}{N} \sum_{n=0}^{N-1} f(x_n) \omega^{j_1 n_2 + j_1 t_2 n_1 + t_1 j_2 n_2}$$

$$= \frac{1}{N} \sum_{n_2=0}^{t_2-1} \left(\sum_{n_1=0}^{t_1-1} f(x_{n_2+t_2 n_1}) \omega^{j_1 t_2 n_1} \right) \omega^{j_1 n_2 + t_1 j_2 n_2}. \tag{7.81}$$

The equation (7.81) can be arranged in a two-step algorithm:

$$F_1(j_1, n_2) := \sum_{n_1=0}^{t_1-1} f(x_{n_2+t_2 n_1}) \omega^{j_1 t_2 n_1}; \tag{7.82}$$

$$\beta_j = F_2(j_1, j_2) := \frac{1}{N} \sum_{n_2=0}^{t_2-1} F_1(j_1, n_2) \omega^{j_1 n_2 + t_1 j_2 n_2}. \tag{7.83}$$

To compute F_1 there are t_1 terms to processed; to compute F_2 there are t_2. The total is $t_1 + t_2$. This must be done for each (j_1, n_2) and (j_1, j_2) pair, or N pairs. The final count is, thus, $N(t_1 + t_2)$ terms processed. The original form processed N terms for each j, a total of N^2 terms. The gain in efficiency, if measured by this standard, is thus $\frac{t_1+t_2}{N}$ and depends very much on N. If, for instance, $N = 1000 = 10 \times 100$, then only 11% of the original 1000000 terms are needed.

Example. Consider the case $N = 6 = 2 \times 3, x_n = \frac{2n\pi}{6}$ and the following discrete data:

n	0	1	2	3	4	5
$f(x_n)$	0	1	1	0	-1	-1

Then values of $F_1(j_1, n_2)$, according to (7.82), are given in the table

		n_2		
		0	1	2
j_1	0	0	0	0
	1	0	2	2

Values of $F_2(j_1, j_2)$, according to (7.84), are given by

		j_2		
		0	1	2
j_1	0	0	0	0
	1	$2\sqrt{3}i$	0	$-2\sqrt{3}i$

Note that in programming language, the j_2 loop is external to the j_1 loop.

Suppose $N = t_1 t_2 t_3$. Let

$$\begin{aligned} j &= j_1 + t_1 j_2 + t_1 t_2 j_3 \\ n &= n_3 + t_3 n_2 + t_3 t_2 n_1. \end{aligned}$$

Then in the nine power terms in ω^{nj}, three will contain the product $t_1 t_2 t_3$ and hence may be neglected. The remaining six power terms may be grouped into a three step algorithm:

$$F_1(j_1, n_2, n_3) := \sum_{n_1=0}^{t_1-1} f(x_n)\omega^{j_1 t_3 t_2 n_1} \tag{7.84}$$

$$F_2(j_1, j_2, n_3) := \sum_{n_2=0}^{t_2-1} F(j_1, n_2, n_3)\omega^{(j_1+t_1 j_2)t_3 n_2} \tag{7.85}$$

$$\beta_j = F_3(j_1, j_2, j_3) := \frac{1}{N}\sum_{n_3=0}^{t_3-1} F_2(j_1, j_2, n_3)\omega^{(j_1+t_1 j_2+t_1 t_2 j_3)n_3}. \tag{7.86}$$

We note that if $N = 10^3$, then only 3% of the original terms are needed.

7.7 Uniform Approximation

We have seen that the problem of approximating a continuous function by a finite linear combination of given functions can be approached in various ways. In this section, we want to use the maximal deviation of the approximation as a measure of the quality of the approximation. That is, we want to consider the normed linear space $C[a, b]$ equipped with the sup-norm $\|\cdot\|_\infty$. We shall limit out attention to approximate a continuous function by elements from the subspace $\mathcal{P}_{n-1}$ of all polynomials of degree $\leq n-1$. Since the error provides a uniform bound on the deviation throughout the entire interval, we refer to the result as a uniform approximation.

We first present a sufficient condition for checking if a given polynomial is a best approximation.

Theorem 7.7.1 *Let $g \in \mathcal{P}_{n-1}$, $f \in C[a, b]$ and $\rho := \|f - g\|_\infty$. Suppose there exist $n+1$ points $a \leq x_1 < \ldots < x_{n+1} \leq b$ such that*

$$|f(x_\nu) - g(x_\nu)| = \rho \tag{7.87}$$

$$f(x_{\nu+1}) - g(x_{\nu+1}) \quad = \quad -(f(x_\nu) - g(x_\nu)) \tag{7.88}$$

for all ν. Then g is a best approximation of f.

(pf): Let

$$M := \{x \in [a,b] || f(x) - g(x)| = \rho\}. \tag{7.89}$$

Certainly $x_\nu \in M$ for all $\nu = 1, \ldots, n+1$. If g is not the best approximation, then there exists a best approximation $\tilde{f}$ that can be written in the form $\tilde{f} = g + p$ for some $p \in \mathcal{P}_{n-1}$ and p is not identically zero. Observer that for all $x \in M$ we have

$$|e(x) - p(x)| < |e(x)| \tag{7.90}$$

where $e(x) := f(x) - g(x)$. The inequality in (7.90) is possible if and only if the sign of $p(x)$ is the same as that of $e(x)$. That is, we must have $(f(x)-g(x))p(x) > 0$ for all $x \in M$. By (7.88), it follows that the polynomial p must change signs at least n times in $[a, b]$. That is, p must have at least n zeros. This contradict with the assumption that p is not identically zero.

Remark. The above theorem asserts only that g is a best approximation whenever there are *at least* $n+1$ points satisfying (7.87) and (7.88). In general, there can be more points where the maximal deviation is achieved.

Example. Suppose we want to approximate $f(x) = \sin 3x$ over the interval $[0, 2\pi]$. It follows from the theorem that if $n - 1 \leq 4$, then the polynomial $g = 0$ is a best approximation of f. Indeed, in this case the difference $f - g$ alternates between its maximal absolute value at six points, whereas the theorem only requires $n + 1$ points. On the other hand, for $n - 1 = 5$ we have $n + 1 = 7$, and $g = 0$ no longer satisfies conditions (7.87) and (7.88). In fact, in this case $g = 0$ is *not* a best approximation from $\mathcal{P}_5$.

Remark. The only property of $\mathcal{P}_{n-1}$ we have used to establish Theorem 7.7.1 is a weaker form of the Fundamental Theorem of Algebra, i.e., any polynomial of degree $n - 1$ has at most $n - 1$ distinct zeros in $[a, b]$. This property is in fact shared by a larger class of functions.

Definition 7.7.1 *Suppose that $g_1, \ldots, g_n \in C[a, b]$ are n linearly independent functions such that every non-trivial element $g \in U := span\{g_1, \ldots, g_n\}$ has at most $n - 1$ distinct zeros in [a,b]. Then we say that U is a Haar space. The basis $\{g_1, \ldots, g_n\}$ of a Haar space is called a Chebyshev system.*

Remark. We have already seen that $\{1, x, x^2, \ldots, x^{n-1}\}$ forms a Chebyshev system. Two other interesting examples are

1. $\{1, e^x, e^{2x}, \ldots, e^{(n-1)x}\}$ over R .
2. $\{1, \sin x, \ldots, \sin mx, \cos x, \ldots, \cos mx\}$ over $[0, 2\pi]$.

We now state without proof the famous result that Theorem 7.7.1 is not only sufficient but is also necessary for a polynomial g to a best approximation. The following theorem is also known as the Alternation Theorem:

Theorem 7.7.2 *The polynomial $g \in \mathcal{P}_{n-1}$ is a best approximation of the function $f \in [a, b]$ if and only if there exist points $a \leq x_1 < \ldots < x_{n+1} \leq b$ such that conditions (7.87) and (7.88) are satisfied.*

Definition 7.7.2 *The set of points $\{x_1, \ldots, x_{n+1}\}$ in Theorem 7.7.2 is referred to as an alternant for f and g.*

Corollary 7.7.1 *For any $f \in C[a, b]$, there is a unique best approximation.*

Theorem 7.7.1 also provides a basis for designing a method for the computation of best approximations of continuous functions. The idea, known as the exchange method of Remez, is as follows:

1. Initially select points such that $a = x_1 < \ldots x_{n+1} = b$.
2. Compute the coefficients of a polynomial $p(x) := a_{n-1}x^{n-1} + \ldots a_0$ and a number ρ so that
$$(f - p^{(0)})(x_\nu) = (-1)^{\nu-1}\rho. \tag{7.91}$$
for all $1 \leq \nu \leq n + 1$. Note that the equations in (7.91) form a linear system which is solvable.

   The problem in step (2) is that even the property of alternating signs has been satisfied in (7.91), it is not necessarily ture that $\rho = \|f - p\|_\infty$. We thus need to replace a new alternant.
3. Locate the extreme points in $[a, b]$ of the absolute error function $e(x) := f(x) - p(x)$. For the sake of simplicity, we assume that there are exactly $n + 1$ extreme points, including a and b.

4. Replace $\{x_k\}$ by the new extreme points and repeat the sequence of steps given above beginning with step (2).

The objective here is to have the set $\{x_k\}$ converge to a true alternant and hence the polynomial converge to a best approximation. It can be proved that the process does converge for any choice of starting values in step (1) for which the value of ρ computed in step (2) in not zero. With additional assumptions on the differentibility of f, it can also be shown that convergence is quadratic. Also the assumption that $e(x)$ possesses exactly $n + 1$ extrem points in step (3) is not essential. For more details, refer to G. Meinardus's book "Approximation of Functions: Theory and Numerical Methods", Springer-Verlag, New York, 1967.