Numerical Analysis

10th ed

R L Burden, J D Faires, and A M Burden

Beamer Presentation Slides
Prepared by
Dr. Annette M. Burden
Youngstown State University

September 8, 2015

Motivation

Let $a_1 x_i + a_0$ denote the ith value on the approximating line and y_i be the ith given y-value.

Assume that the independent variables, x_i, are exact and the dependent variables, the y_i, are suspect.

Problem: Find the equation of the best linear approximation to the data $\{(x_i, y_i)\}_{i=1}^{m}$.

Minimax Problem

This approach requires that values of a_0 and a_1 be found to minimize

$$E_\infty(a_0, a_1) = \max_{1 \le i \le m} \{|y_i - (a_1 x_i + a_0)|\}.$$

DRAWBACK: assigns too much weight to a bit of data that is badly in error.

Absolute Deviation Problem

This approach requires finding values of a_0 and a_1 to minimize

$$E_1(a_0, a_1) = \sum_{i=1}^{m} |y_i - (a_1 x_i + a_0)|.$$

DRAWBACK: does not give sufficient weight to a point that is considerably out of line with the approximation.

NOTE

To minimize a function of two variables, we need to set its partial derivatives to zero and simultaneously solve the resulting equations. In the case of the absolute deviation, we need to find a_0 and a_1 with

$$0 = \frac{\partial}{\partial a_0} \sum_{i=1}^{m} |y_i - (a_1 x_i + a_0)| \quad \text{and} \quad 0 = \frac{\partial}{\partial a_1} \sum_{i=1}^{m} |y_i - (a_1 x_i + a_0)|.$$

ISSUE: Absolute-value function is not differentiable at zero, and we might not be able to find solutions to this pair of equations.

Least Squares Problem

This approach involves determining the best approximating line when error involved is the sum of the squares of the differences between the y-values on the approximating line and the given y-values. Hence, constants a_0 and a_1 must be found that minimize the least squares error:

$$E_2(a_0, a_1) = \sum_{i=1}^{m} [y_i - (a_1 x_i + a_0)]^2 .$$

Taking partial derivatives with respect to a_0 and a_1 leads to a system of equations with solution

$$a_0 = \frac{\sum_{i=1}^m x_i^2 \sum_{i=1}^m y_i - \sum_{i=1}^m x_i y_i \sum_{i=1}^m x_i}{m\left(\sum_{i=1}^m x_i^2\right) - \left(\sum_{i=1}^m x_i\right)^2}$$

and

$$a_1 = \frac{m\sum_{i=1}^m x_i y_i - \sum_{i=1}^m x_i \sum_{i=1}^m y_i}{m\left(\sum_{i=1}^m x_i^2\right) - \left(\sum_{i=1}^m x_i\right)^2}.$$

Polynomial Least Squares Problem

The general problem of approximating a set of data, $\{(x_i, y_i) \mid i = 1, 2, \ldots, m\}$, with an algebraic polynomial

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

of degree $n < m - 1$, using the least squares procedure is handled similarly. We choose the constants a_0, a_1, ..., a_n to minimize the least squares error $E = E_2(a_0, a_1, \ldots, a_n)$, where

$$E = \sum_{i=1}^{m} (y_i - P_n(x_i))^2$$

Exponential Least Squares Problem

At times it is appropriate to assume that the data are exponentially related. This requires the approximating function to be of the form

$$y = be^{ax}, \quad \text{or} \quad y = bx^a,$$

for some constants a and b. The difficulty with applying the least squares procedure in a situation of this type comes from attempting to minimize

$$E = \sum_{i=1}^{m} (y_i - be^{ax_i})^2, \quad \text{or} \quad E = \sum_{i=1}^{m} (y_i - bx_i^a)^2.$$

Definition (8.1)

The set of functions $\{\phi_0, \ldots, \phi_n\}$ is said to be **linearly independent** on $[a, b]$ if, whenever

$$c_0\phi_0(x) + c_1\phi_1(x) + \cdots + c_n\phi_n(x) = 0, \quad \text{for all } x \in [a, b],$$

we have $c_0 = c_1 = \cdots = c_n = 0$. Otherwise the set of functions is said to be **linearly dependent**.

Theorem (8.2)

Suppose that, for each $j = 0, 1, \ldots, n$, $\phi_j(x)$ is a polynomial of degree j. Then $\{\phi_0, \ldots, \phi_n\}$ is linearly independent on any interval $[a, b]$.

Theorem (8.3)

Suppose that $\{\phi_0(x), \phi_1(x), \ldots, \phi_n(x)\}$ is a collection of linearly independent polynomials in $\prod_n$. Then any polynomial in $\prod_n$ can be written uniquely as a linear combination of $\phi_0(x)$, $\phi_1(x)$, ..., $\phi_n(x)$.

Definition (8.4)

An integrable function w is called a **weight function** on the interval I if $w(x) \geq 0$, for all x in I, but $w(x) \not\equiv 0$ on any subinterval of I.

Definition (8.5)

$\{\phi_0, \phi_1, \ldots, \phi_n\}$ is said to be an **orthogonal set of functions** for the interval $[a, b]$ with respect to the weight function w if

$$\int_a^b w(x)\phi_k(x)\phi_j(x)\,dx = \begin{cases} 0, & \text{when } j \neq k, \\ \alpha_j > 0, & \text{when } j = k. \end{cases}$$

If, in addition, $\alpha_j = 1$ for each $j = 0, 1, \ldots, n$, the set is said to be **orthonormal**.

Theorem (8.6)

If $\{\phi_0, \ldots, \phi_n\}$ is an orthogonal set of functions on an interval $[a, b]$ with respect to the weight function w, then the least squares approximation to f on $[a, b]$ with respect to w is

$$P(x) = \sum_{j=0}^{n} a_j \phi_j(x),$$

where, for each $j = 0, 1, \ldots, n$,

$$a_j = \frac{\int_a^b w(x)\phi_j(x)f(x)\,dx}{\int_a^b w(x)[\phi_j(x)]^2\,dx} = \frac{1}{\alpha_j}\int_a^b w(x)\phi_j(x)f(x)\,dx.$$

Theorem (8.7)

The set of polynomial functions $\{\phi_0, \phi_1, \ldots, \phi_n\}$ *defined in the following way is orthogonal on* $[a, b]$ *with respect to the weight function* w.

$$\phi_0(x) \equiv 1, \quad \phi_1(x) = x - B_1, \quad \text{for each } x \text{ in } [a, b] \text{ where}$$

$$B_1 = \frac{\int_a^b xw(x)[\phi_0(x)]^2 \, dx}{\int_a^b w(x)[\phi_0(x)]^2 \, dx}, \quad \text{and when } k \geq 2,$$

$$\phi_k(x) = (x - B_k)\phi_{k-1}(x) - C_k\phi_{k-2}(x), \quad \text{for each } x \text{ in } [a, b] \text{ where}$$

$$B_k = \frac{\int_a^b xw(x)[\phi_{k-1}(x)]^2 \, dx}{\int_a^b w(x)[\phi_{k-1}(x)]^2 \, dx} \quad \text{and}$$

$$C_k = \frac{\int_a^b xw(x)\phi_{k-1}(x)\phi_{k-2}(x) \, dx}{\int_a^b w(x)[\phi_{k-2}(x)]^2 \, dx}.$$

Corollary (8.8)

For any $n > 0$, the set of polynomial functions $\{\phi_0, \ldots, \phi_n\}$ given in Theorem 8.7 is linearly independent on $[a, b]$ and

$$\int_a^b w(x)\phi_n(x)Q_k(x)\,dx = 0,$$

for any polynomial $Q_k(x)$ of degree $k < n$.

NOTE

The Chebyshev polynomials $\{T_n(x)\}$ are orthogonal on $(-1, 1)$ with respect to the weight function $w(x) = (1 - x^2)^{-1/2}$.
For $x \in [-1, 1]$, define

$$T_n(x) = \cos[n \arccos x], \quad \text{for each } n \geq 0.$$

For each n, $T_n(x)$ is a polynomial in x.

Theorem (8.9)

The Chebyshev polynomial $T_n(x)$ *of degree* $n \geq 1$ *has n simple zeros in* $[-1, 1]$ *at*

$$\bar{x}_k = \cos\left(\frac{2k-1}{2n}\pi\right), \quad \textit{for each } k = 1, 2, \ldots, n.$$

Moreover, $T_n(x)$ *assumes its absolute extrema at*

$$\bar{x}'_k = \cos\left(\frac{k\pi}{n}\right) \quad \textit{with} \quad T_n(\bar{x}'_k) = (-1)^k,$$

for each $k = 0, 1, \ldots, n.$

NOTE

The monic (polynomials with leading coefficient 1) Chebyshev polynomials $\tilde{T}_n(x)$ are derived from the Chebyshev polynomials $T_n(x)$ by dividing by the leading coefficient 2^{n-1}. Hence

$$\tilde{T}_0(x) = 1 \quad \text{and} \quad \tilde{T}_n(x) = \frac{1}{2^{n-1}} T_n(x), \quad \text{for each } n \geq 1.$$

The recurrence relationship satisfied by the Chebyshev polynomials implies that

$$\begin{aligned} \tilde{T}_2(x) &= x\tilde{T}_1(x) - \frac{1}{2}\tilde{T}_0(x) \quad \text{and} \\ \tilde{T}_{n+1}(x) &= x\tilde{T}_n(x) - \frac{1}{4}\tilde{T}_{n-1}(x), \quad \text{for each } n \geq 2. \end{aligned}$$

Chapter 8.3: Chebyshev Polynomials, Economization of Power Series

Theorem (8.10)

The polynomials of the form $\tilde{T}_n(x)$, when $n \geq 1$, have the property that

$$\frac{1}{2^{n-1}} = \max_{x\in[-1,1]} |\tilde{T}_n(x)| \leq \max_{x\in[-1,1]} |P_n(x)|, \quad \text{for all } P_n(x) \in \widetilde{\prod}_n,$$

where $\widetilde{\prod}_n$ denotes the set of monic polynomials of degree n. Moreover, equality occurs only if $P_n \equiv \tilde{T}_n$.

Corollary (8.11)

Suppose that $P(x)$ is the interpolating polynomial of degree at most n with nodes at the zeros of $T_{n+1}(x)$. Then for each $f \in C^{n+1}[-1, 1]$

$$\max_{x\in[-1,1]} |f(x) - P(x)| \leq \frac{1}{2^n(n+1)!} \max_{x\in[-1,1]} |f^{(n+1)}(x)|.$$

Algorithm 8.1: PADÉ APPROXIMATION

To obtain the rational approximation

$$r(x) = \frac{p(x)}{q(x)} = \frac{\sum_{i=0}^{n} p_i x^i}{\sum_{j=0}^{m} q_j x^j}$$

for a given function $f(x)$:

INPUT nonnegative integers m and n.

OUTPUT coefficients $q_0, q_1, \ldots, q_m$ and $p_0, p_1, \ldots, p_n$.

Step 1 Set $N = m + n$.

Step 2 For $i = 0, 1, \ldots, N$ set $a_i = \dfrac{f^{(i)}(0)}{i!}$.

(*The coefficients of the Maclaurin polynomial are* $a_0, \ldots, a_N$*, which could be input instead of calculated.*)

Algorithm 8.1: PADÉ APPROXIMATION

Step 3 Set $q_0 = 1$;
$p_0 = a_0$.
Step 4 For $i = 1, 2, \ldots, N$ do Steps 5–10.
(*Set up a linear system with matrix B.*)
Step 5 For $j = 1, 2, \ldots, i - 1$
if $j \le n$ then set $b_{i,j} = 0$.
Step 6 If $i \le n$ then set $b_{i,i} = 1$.
Step 7 For $j = i + 1, i + 2, \ldots, N$ set $b_{i,j} = 0$.
Step 8 For $j = 1, 2, \ldots, i$
if $j \le m$ then set $b_{i,n+j} = -a_{i-j}$.
Step 9 For $j = n + i + 1, n + i + 2, \ldots, N$ set $b_{i,j} = 0$.
Step 10 Set $b_{i,N+1} = a_i$.
(*Steps 11–22 solve the linear system using partial pivoting.*)

Algorithm 8.1: PADÉ APPROXIMATION

Step 11 For $i = n + 1, n + 2, \ldots, N - 1$ do Steps 12–18.

Step 12 Let k be the smallest integer with $i \leq k \leq N$ and
$|b_{k,i}| = \max_{i \leq j \leq N} |b_{j,i}|$.
(*Find pivot element.*)

Step 13 If $b_{k,i} = 0$ then OUTPUT
("The system is singular ");
STOP.

Step 14 If $k \neq i$ then (*Interchange row i and row k.*)
for $j = i, i + 1, \ldots, N + 1$ set

$$\begin{aligned} b_{COPY} &= b_{i,j}; \\ b_{i,j} &= b_{k,j}; \\ b_{k,j} &= b_{COPY}. \end{aligned}$$

Step 15 For $j = i + 1, i + 2, \ldots, N$ do Steps 16–18.
(*Perform elimination.*)

Algorithm 8.1: PADÉ APPROXIMATION

Step 16 Set $xm = \dfrac{b_{j,i}}{b_{i,i}}$.

Step 17 For $k = i+1, i+2, \ldots, N+1$
set $b_{j,k} = b_{j,k} - xm \cdot b_{i,k}$.

Step 18 Set $b_{j,i} = 0$.

Step 19 If $b_{N,N} = 0$ then OUTPUT ("The system is singular");
STOP.

Step 20 If $m > 0$ then set $q_m = \dfrac{b_{N,N+1}}{b_{N,N}}$. (*Start backward subs.*)

Step 21 For $i = N-1, N-2, \ldots, n+1$, $q_{i-n} = \dfrac{b_{i,N+1} - \sum_{j=i+1}^{N} b_{i,j} q_{j-n}}{b_{i,i}}$.

Step 22 For $i = n, n-1, \ldots, 1$ set $p_i = b_{i,N+1} - \sum_{j=n+1}^{N} b_{i,j} q_{j-n}$.

Step 23 OUTPUT $(q_0, q_1, \ldots, q_m, p_0, p_1, \ldots, p_n)$;
STOP. (*The procedure was successful.*)

Algorithm 8.2: CHEBYSHEV RATIONAL APPROXIMATION

To obtain the rational approximation

$$r_T(x) = \frac{\sum_{k=0}^{n} p_k T_k(x)}{\sum_{k=0}^{m} q_k T_k(x)}$$

for a given function $f(x)$:

INPUT nonnegative integers m and n.

OUTPUT coefficients $q_0, q_1, \ldots, q_m$ and $p_0, p_1, \ldots, p_n$.

Step 1 Set $N = m + n$.
Step 2 Set $a_0 = \frac{2}{\pi} \int_0^{\pi} f(\cos\theta)\, d\theta$;
(*The coefficient a_0 is doubled for computational efficiency.*

Algorithm 8.2: CHEBYSHEV RATIONAL APPROXIMATION

For $k = 1, 2, \ldots, N + m$ set
$a_k = \frac{2}{\pi} \int_0^\pi f(\cos\theta) \cos k\theta \, d\theta$.
(*The integrals can be evaluated using a numerical integration procedure or the coefficients can be input) directly.*)

Step 3 Set $q_0 = 1$.

Step 4 For $i = 0, 1, \ldots, N$ do Steps 5–9.
(*Set up a linear system with matrix B.*)

Step 5 For $j = 0, 1, \ldots, i$ if $j \leq n$, then set $b_{i,j} = 0$.

Step 6 If $i \leq n$ then set $b_{i,i} = 1$.

Step 7 For $j = i + 1, i + 2, \ldots, n$ set $b_{i,j} = 0$.

Step 8 For $j = n + 1, n + 2, \ldots, N$
if $i \neq 0$ then set $b_{i,j} = -\frac{1}{2}(a_{i+j-n} + a_{|i-j+n|})$
else set $b_{i,j} = -\frac{1}{2} a_{j-n}$.

Algorithm 8.2: CHEBYSHEV RATIONAL APPROXIMATION

Step 9 If $i \neq 0$ then set $b_{i,N+1} = a_i$
else set $b_{i,N+1} = \frac{1}{2}a_i$.

(*Steps 10–21 solve the linear system using partial pivoting.*)

Step 10 For $i = n+1, n+2, \ldots, N-1$ do Steps 11–17.

Step 11 Let k be the smallest integer with $i \leq k \leq N$ and
$|b_{k,i}| = \max_{i \leq j \leq N} |b_{j,i}|$. (*Find pivot element.*)

Step 12 If $b_{k,i} = 0$ then OUTPUT ("The system is singular");
STOP.

Step 13 If $k \neq i$ then (*Interchange row i and row k.*)
for $j = i, i+1, \ldots, N+1$ set
$b_{COPY} = b_{i,j}$;
$b_{i,j} = b_{k,j}$;
$b_{k,j} = b_{COPY}$.

Algorithm 8.2: CHEBYSHEV RATIONAL APPROXIMATION

Step 14 For $j = i+1, i+2, \ldots, N$ do Steps 15–17.
(*Perform elimination.*)

Step 15 Set $xm = \dfrac{b_{j,i}}{b_{i,i}}$.

Step 16 For $k = i+1, i+2, \ldots, N+1$
set $b_{j,k} = b_{j,k} - xm \cdot b_{i,k}$.

Step 17 Set $b_{j,i} = 0$.

Step 18 If $b_{N,N} = 0$ then OUTPUT ("The system is singular"); STOP.

Step 19 If $m > 0$ then set $q_m = \dfrac{b_{N,N+1}}{b_{N,N}}$. (*Start backward subs.*)

Step 20 For $i = N-1, N-2, \ldots, n+1$, set $q_{i-n} = \dfrac{b_{i,N+1} - \sum_{j=i+1}^{N} b_{i,j} q_{j-n}}{b_{i,i}}$.

Step 21 For $i = n, n-1, \ldots, 0$ set $p_i = b_{i,N+1} - \sum_{j=n+1}^{N} b_{i,j} q_{j-n}$.

Step 22 OUTPUT $(q_0, q_1, \ldots, q_m, p_0, p_1, \ldots, p_n)$; STOP. (*successful.*)

Lemma (8.12)

Suppose that the integer r is not a multiple of $2m$. Then

- $$\sum_{j=0}^{2m-1} \cos rx_j = 0 \quad \textit{and} \quad \sum_{j=0}^{2m-1} \sin rx_j = 0.$$

Moreover, if r is not a multiple of m, then

- $$\sum_{j=0}^{2m-1} (\cos rx_j)^2 = m \quad \textit{and} \quad \sum_{j=0}^{2m-1} (\sin rx_j)^2 = m.$$

where

$$x_j = -\pi + \left(\frac{j}{m}\right)\pi, \quad \textit{for each } j = 0, 1, \ldots, 2m-1.$$

Theorem (8.13)

The constants in the summation

$$S_n(x) = \frac{a_0}{2} + a_n \cos nx + \sum_{k=1}^{n-1} (a_k \cos kx + b_k \sin kx)$$

that minimize the least squares sum

$$E(a_0, \ldots, a_n, b_1, \ldots, b_{n-1}) = \sum_{j=0}^{2m-1} (y_j - S_n(x_j))^2 \quad \textit{are}$$

- $$a_k = \frac{1}{m} \sum_{j=0}^{2m-1} y_j \cos kx_j, \quad \textit{for each } k = 0, 1, \ldots, n, \quad \textit{and}$$
- $$b_k = \frac{1}{m} \sum_{j=0}^{2m-1} y_j \sin kx_j, \quad \textit{for each } \; k = 1, 2, \ldots, n-1.$$

Algorithm 8.3: FAST FOURIER TRANSFORM

To compute the coefficients in the summation

$$\frac{1}{m}\sum_{k=0}^{2m-1} c_k e^{ikx} = \frac{1}{m}\sum_{k=0}^{2m-1} c_k(\cos kx + i\sin kx), \quad \text{where } i = \sqrt{-1},$$

for the data $\{(x_j, y_j)\}_{j=0}^{2m-1}$ where $m = 2^p$ and $x_j = -\pi + j\pi/m$ for $j = 0, 1, \ldots, 2m-1$:

INPUT $m, p; y_0, y_1, \ldots, y_{2m-1}$.

OUTPUT complex numbers $c_0, \ldots, c_{2m-1}$; real numbers $a_0, \ldots, a_m; b_1, \ldots, b_{m-1}$.

Algorithm 8.3: FAST FOURIER TRANSFORM

Step 1 Set $M = m$;
$q = p$;
$\zeta = e^{\pi i/m}$.

Step 2 For $j = 0, 1, \ldots, 2m - 1$ set $c_j = y_j$.

Step 3 For $j = 1, 2, \ldots, M$ set $\xi_j = \zeta^j$;
$\xi_{j+M} = -\xi_j$.

Step 4 Set $K = 0$;
$\xi_0 = 1$.

Step 5 For $L = 1, 2, \ldots, p + 1$ do Steps 6–12.

Step 6 While $K < 2m - 1$ do Steps 7–11.

Step 7 For $j = 1, 2, \ldots, M$ do Steps 8–10.

Algorithm 8.3: FAST FOURIER TRANSFORM

Step 7 For $j = 1, 2, \ldots, M$ do Steps 8–10.

Step 8 Let $K = k_p \cdot 2^p + k_{p-1} \cdot 2^{p-1} + \cdots + k_1 \cdot 2 + k_0$;

(*Decompose* k.)

set $K_1 = K/2^q = k_p \cdot 2^{p-q} + \cdots + k_{q+1} \cdot 2 + k_q$;

$K_2 = k_q \cdot 2^p + k_{q+1} \cdot 2^{p-1} + \cdots + k_p \cdot 2^q$.

Step 9 Set $\eta = c_{K+M}\xi_{K_2}$;

$c_{K+M} = c_K - \eta$;

$c_K = c_K + \eta$.

Step 10 Set $K = K + 1$.

Step 11 Set $K = K + M$.

Step 12 Set $K = 0$;

$M = M/2$;

$q = q - 1$.

Algorithm 8.3: FAST FOURIER TRANSFORM

Step 13 While $K < 2m - 1$ do Steps 14–16.

Step 14 Let $K = k_p \cdot 2^p + k_{p-1} \cdot 2^{p-1} + \cdots + k_1 \cdot 2 + k_0$;
(*Decompose* k.)
set $j = k_0 \cdot 2^p + k_1 \cdot 2^{p-1} + \cdots + k_{p-1} \cdot 2 + k_p$.

Step 15 If $j > K$ then interchange c_j and c_k.

Step 16 Set $K = K + 1$.

Step 17 Set $a_0 = c_0/m$;
$a_m = \mathrm{Re}(e^{-i\pi m} c_m/m)$.

Step 18 For $j = 1, \ldots, m-1$ set $a_j = \mathrm{Re}(e^{-i\pi j} c_j/m)$;
$b_j = \mathrm{Im}(e^{-i\pi j} c_j/m)$.

Step 19 OUTPUT $(c_0, \ldots, c_{2m-1}; a_0, \ldots, a_m; b_1, \ldots, b_{m-1})$;
STOP.