CS323 **Review Sheet 3a** **November 11, 2019**

Approximation by Polynomials

Given a continuous function $f(x)$ on an interval $[a, b]$, the goal is to find a function $g(x)$ as an approximation to f. One of the reasons for wanting such an approximation could be that we want to compute

$$\int f(x)dx \qquad \text{or} \qquad f'(x) \tag{1}$$

but for some reason, we cannot perform these operations directly on f, (for example if the integral of f is not known). The hope is that we can use

$$\int g(x)dx \qquad \text{or} \qquad g'(x) \tag{2}$$

polynomials. Therefore our approximating functions $g(x)$ will *always* be polynomials.

1. **Polynomials** A function of the form

$$P_n(x) = a_0 + a_1x + a_2x^2 + \cdots + a_jx^j + \cdots + a_nx^n\ ,\ a_n \neq 0, \tag{3}$$

is a polynomial of degree n. The coefficients $a_0, \ldots, a_n$ are *real* numbers. P_n is clearly continuous. If we evaluate P_n at $cx + d$, $c \neq 0$, we can collect terms with the same power of x and write the result as $P_n(cx + d) = b_0 + b_1x + \cdots + b_nx^n$. Also it is clear from the binomial theorem that $b_n = a_nc^n$ so $P_n(cx+d)$ is *also a polynomial of degree n.* Some other facts about polynomials are:

   (a) If we differentiate (3) with respect to x we see

$$P_n'(x) = a_1 + 2a_2x + \cdots + ja_jx^{j-1} + \cdots + na_nx^{n-1},$$

   a polynomial of degree $n - 1$. In fact differentiating j times, and writing $P_n^{(j)}$ as the j^{th} derivative,

$$P_n^{(j)}(x) = j!a_j + \frac{(j+1)!}{1!}a_{j+1}x + \cdots + \frac{n!}{(n-j)!}a_nx^{n-j}. \tag{4}$$

   (b) If we integrate (3) with respect to x we see

$$\int P_n(x)dx = a_0x + \frac{a_1}{2}x^2 + \cdots + \frac{a_j}{j+1}x^{j+1} + \cdots + \frac{a_n}{n+1}x^{n+1} + C,$$

   a polynomial of degree $n + 1$

   (c) If we add $P_n(x) + Q_m(x)$, polynomials of different degrees $m \neq n$, the result is a polynomial of degree $= \max(m, n)$. If the degrees are the same, the sum is a polynomial of degree $\leq n$ (e.g., coefficients of x^n could cancel). If we multiply polynomials, the result, $P_n(x)Q_m(x)$ is a polynomial of degree $m + n$.

   (d) **Horner's Method to evaluate P_n:** Given $a_0, \ldots, a_n$ and a point $x = t$ at which we want to evaluate P_n, the following describes an efficient procedure to compute $P_n(t)$ in (3).

   - poly$\leftarrow a_n$
   - **FOR** $i = 1$ **TO** n **DO**

- poly $\leftarrow$ poly$*t + a_{n-i}$
- **ENDFOR**

Clearly this uses n multiply and n add steps and returns the value of $P_n(t)$ in the variable *poly*.

(e) Finally the *Fundamental Theorem of Algebra* states that a non-zero polynomial of degree n has n complex roots. This means that P_n has *at most* n real roots (or else P_n is identically zero). We will use this observation to deduce that if there are $n+1$ distinct values $u_0, u_1, \ldots, u_n$, $u_i \neq u_j$, $i \neq j$, and if $P_n(u_j) = 0$, $j = 0, \ldots, n$, then $P_n(x) = 0$ *for all* x.

2. **Taylor Polynomials:** Given a function $f(x)$ which is n times differentiable, we will take a value $x = u$ and construct a polynomial that "resembles" f in $n+1$ different ways. Specifically the polynomial

$$T_n(x) = a_0 + a_1(x-u) + \cdots + a_j(x-u)^j + \cdots + a_n(x-u)^n \tag{5}$$

will be forced to satisfy the following $n+1$ conditions:

$$f^{(j)}(u) = T_n^{(j)}(u)\,,\ j = 0, 1, \ldots, n. \tag{6}$$

When $j = 0$, this says that $f(u) = T_n(u)$. The other n conditions say that f and T_n have the same first n derivatives at the point $x = u$.

Each one of these conditions will determine one of the coefficients in (5). For example when $j = 0$, (6) says that $f(u) = T_n(u)$ and using this in (5), $T_n(u) = a_0$. Therefore we learn that

$$a_0 = f(u).$$

In general we want to differentiate (5) j times. From (4)

$$T_n^{(j)}(x) = j!a_j + \frac{(j+1)!}{1!}a_{j+1}(x-u) + \cdots + \frac{n!}{(n-j)!}a_n(x-u)^{n-j},$$

so $T_n^{(j)}(u) = j!a_j$. Using (6) we learn that

$$a_j = \frac{f^{(j)}(u)}{j!}.$$

Using these values in (5), the $\underline{n^{th}\text{ Taylor polynomial for } f\text{, expanded about } u}$ is

$$T_n(x) = f(u) + f'(u)(x-u) + \cdots + \frac{f^{(j)}(u)}{j!}(x-u)^j + \cdots + \frac{f^{(n)}(u)}{n!}(x-u)^n. \tag{7}$$

Note also that we can express (7) (use $0! \equiv 1$) as

$$T_n(x) = \sum_{j=0}^{n} \frac{f^{(j)}(u)}{j!}(x-u)^j = T_{n-1}(x) + \frac{f^{(n)}(u)}{n!}(x-u)^n.$$

(a) **An example:** Actually this is two examples of computing T_2, the quadratic Taylor polynomial for $f(x)$; we do it twice with different values of u. Let $f(x) = \sqrt{x}$. We will compute $T_2(x)$ expanded about $u = 1$ and about $u = 9/4$ with the help of the following table:

j	$f^{(j)}(x)$	$f^{(j)}(1)$	a_j	•	$f^{(j)}(9/4))$	a_j
0	$x^{1/2}$	1	1	•	3/2	3/2
1	$\frac{1}{2}x^{-1/2}$	1/2	1/2	•	1/3	1/3
2	$-\frac{1}{4}x^{-3/2}$	−1/4	−1/8	•	−2/27	−1/27
3	$\frac{3}{8}x^{-5/2}$			•		

Therefore the second Taylor Polynomial expanded about $u = 1$ is

$$T_2(x) = 1 + \frac{1}{2}(x-1) - \frac{1}{8}(x-1)^2$$

and the second Taylor Polynomial expanded about $u = 9/4$ is

$$T_2(x) = 3/2 + \frac{1}{3}(x - \frac{9}{4}) - \frac{1}{27}(x - \frac{9}{4})^2.$$

Note that $T_2(2) = 1.375$, in the first case, and 1.4143519 in the second; the latter is an excellent approximation to $\sqrt{2}$.

(b) **Taylor's Theorem:** $f(x) - T_n(x)$ is the error of the n^{th} Taylor polynomial at the point x. Taylor's theorem says exactly what it is in terms of n, x, u, and derivatives of f: *"If f has $n+1$ continuous derivatives, there is a point θ between x and u such that*

$$f(x) - T_n(x) = \frac{f^{(n+1)}(\theta)}{(n+1)!}(x-u)^{n+1}." \tag{8}$$

(c) **The example again:** We will apply Taylor's Theorem to bound the error of the approximations made in (a). First, for the case $u = 1$, Taylor's theorem says that

$$\sqrt{x} - T_2(x) = \frac{\theta^{-5/2}}{16}(x-1)^3$$

for some value of θ between x and 1. Taking $x = 2$,

$$\sqrt{2} - 1.375 = \frac{\theta^{-5/2}}{16}\,,\ \theta \in (1,2),$$

and this implies that $1.385417 \le \sqrt{2} \le 1.4375$. Now for the case $u = 9/4$, Taylor's Theorem says

$$\sqrt{x} - T_2(x) = \frac{\theta^{-5/2}}{16}(x - \frac{9}{4})^3$$

for some value of θ between x and $9/4$. Again taking $x = 2$,

$$\sqrt{2} - 1.4143519 = -\frac{\theta^{-5/2}}{(16)(64)}\,,\ \theta \in (2, 9/4),$$

and this implies that $1.4141078 \le \sqrt{2} \le 1.4143305$.

3. **Interpolation:** Given $n+1$ distinct values $u_0, u_1, \ldots, u_n$, $u_i \ne u_j$, $i \ne j$, we will find a polynomial $I_n(x)$ that has the same value as f at each of the u_i; i.e., $f(u_i) = I_n(u_i)$, $i = 0, 1, \ldots, n$. These u_i are called *collocation points* and I_n is said to *interpolate* f (or agree with f) at these collocation points.

(a) **Interpolation Theorem:** Given $f(x)$ and $n+1$ distinct collocation points $u_0, u_1, \ldots, u_n$, there is a *unique* polynomial I_n of degree at most n that interpolates f (note that this says that $n+1$ distinct points in the plane determine a unique n^{th} degree polynomial passing through them [e.g. 2 points determine a line]). To verify this statement, note that

$$I_n(x) = \sum_{i=0}^{n} f(u_i) \left\{ \prod_{j \neq i} \left(\frac{x - u_j}{u_i - u_j} \right) \right\} \tag{9}$$

is a degree at most n polynomial that interpolates (note that for each i, the expression in curly brackets is a polynomial of degree n which equals 1 when $x = u_i$ and equals 0 when $x = u_j$, $j \neq i$). Suppose P_n is another. The function $h(x) \equiv I_n(x) - P_n(x)$ is a polynomial of degree at most n and it has roots at $u_0, u_1, \ldots, u_n$ (i.e., $n+1$ roots). By the fundamental theorem of algebra h is identically zero, so I_n is unique. The formula in (9) is called Lagrange's Form of I_n.

(b) **An Example:** Let $f(x) = \sqrt{x}$ and take $u_0 = 1$, $u_1 = 9/4$, and $u_2 = 4$. Then from (9),

$$I_2(x) = 1 \left\{ \frac{(x - 9/4)(x - 4)}{(1 - 9/4)(1 - 4)} \right\} + \frac{3}{2} \left\{ \frac{(x - 1)(x - 4)}{(9/4 - 1)(9/4 - 4)} \right\} + 2 \left\{ \frac{(x - 1)(x - 9/4)}{(4 - 1)(4 - 9/4)} \right\}.$$

Langrage's form if I_2, above, can be simplified to $(-4x^2 + 55x + 54)/105$ and $I_2(2) = 148/105 = 1.4095238$ is its approximation to $\sqrt{2}$.

(c) **Error of Interpolation:** Let $I_n(x)$ be the polynomial of degree at most n that interpolates $f(x)$ at $u_0, u_1, \ldots, u_n$, $n+1$ distinct collocation points. Assuming $f^{(n+1)}$ is continuous, the following formula expresses the error, $E_{I_n}(t) \equiv f(x) - I_n(x)$:

$$f(x) - I_n(x) = \frac{f^{(n+1)}(\theta)}{(n+1)!}(x - u_0)(x - u_1) \cdots (x - u_n) = \frac{f^{(n+1)}(\theta)}{(n+1)!} \prod_{j=0}^{n} (x - u_j), \tag{10}$$

for some value of θ, $\min(x, u_0, u_1, \ldots, u_n) \leq \theta \leq \max(x, u_0, u_1, \ldots, u_n)$. Note that this expression is zero at each collocation point. Also, it is instructive to compare (10) to (8).

(d) **The Example Again:** The error formula says there is a θ between $\min(x, 1, 9/4, 4)$ and $\max(x, 1, 9/4, 4)$ for which

$$E_{I_2}(x) \equiv f(x) - I_2(x) = \frac{(x - 1)(x - 9/4)(x - 4)}{16\theta^{5/2}}.$$

Therefore $\sqrt{2} - 1.4095238 = 1/(32\theta^{5/2})$ for some $\theta \in (1, 4)$, and this implies that $1.4105004 \leq \sqrt{2} \leq 1.4407738$.

(e) **Cost to evaluate** $I_n(x)$ We describe the evaluation of the expression in (9) for a given value of x via the following pseudocode for a procedure "LAGRANGE". The inputs are n (the degree), an array U$[0, 1, \ldots, n]$ with the $n+1$ collocation points, and x, where I_n is being evaluated. The output is VAL which $= I_n(x)$. The procedure can call $f(z)$ to obtain the value of f at z.

LAGRANGE$(n, U, x;$ **VAL)**

- **VAL** $\leftarrow 0$
- **FOR** $i = 0$ **TO** n **DO**
- **PROD** $\leftarrow f(U[i])$
- **FOR** $j = 0$ **TO** n **DO**
- **IF** $j \neq i$ **DO**
- **{PROD** $\leftarrow$ **PROD***$(x - U[j])/(U[i] - U[j])$**}**
- **ENDFOR**
- **VAL** $\leftarrow$ **VAL** + **PROD**
- **ENDFOR**

¿From this procedure, or from (9) itself, we easily see that $I_n(x)$ is obtained using $2n(n+1)$ multiply or divide steps, and $2n^2 + 3n$ add or subtract steps, and $n + 1$ evaluations of f. This is much more work than the cost of evaluating the polynomial $a_0 + a_1x + \cdots + a_nx^n$ using Horner's method, namely, n multiply and n add steps (see 1d).

To be able to use the efficient Horner evaluation, we need to write $I_n(x)$ as $a_0 + a_1x + \cdots + a_nx^n$. This is called the standard form of I_n. To learn the coefficients for the standard form note that $I_n(u_i) = f(u_i)$ implies

$$a_0 + a_1(u_i) + a_2(u_i)^2 + \cdots + a_n(u_i)^n = f(u_i),$$

a linear equation in the $n+1$ unknown a_j's. There are $n+1$ such equations, one for each collocation point. The system may be written as

$$C\underline{a} = \underline{f},$$

where $c_{ij} = (u_i)^j$ and $f_i = f(u_i)$, $i, j = 0, 1, \ldots, n$. By the interpolation theorem it has a unique solution. Conclusion: The entries of C may be computed in $(n + 1)(n - 1)$ multiplication steps. Because the first column of C is all ones, the system may be solved in $(n^3-n)/3+n^2$ operations ($*$ or /) which gives the coefficients of I_n in the standard form, and now $I_n(x)$ can be evaluated in n multiplications and n add/subtracts by Horner's method.

(f) **Newton's Form of** I_n is a representation *whose coefficients can be found in* n^2 *multiply or divide OPS* and which *can be evaluated in* n *multiplication OPS*:

i. **What it is:** Suppose $I_n(x)$ interpolate f at $u_0, u_1, \ldots, u_n$. Add a new collocation point u_{n+1} and let I_{n+1} interpolate at the original $n + 1$ points *and* at u_{n+1}. Then $h(x) = I_{n+1}(x) - I_n(x)$ is a polynomial of degree at most $n + 1$ and it has a root at each original collocation point, $u_0, \ldots, u_n$. Therefore for some constant c_{n+1}, $h(x) = c_{n+1}(x - u_0)(x - u_1)\cdots(x - u_n)$ and this gives

$$I_{n+1}(x) = I_n(x) + c_{n+1}(x - u_0)(x - u_1)\cdots(x - u_n); \tag{11}$$

To learn the value of c_{n+1}, put $x = u_{n+1}$ and use the fact that I_{n+1} interpolates to see that

$$c_{n+1} = \frac{f(u_{n+1}) - I_n(u_{n+1})}{(u_{n+1} - u_0)(u_{n+1} - u_1)\cdots(u_{n+1} - u_n)} = \frac{f(u_{n+1}) - I_n(u_{n+1})}{\prod_{i=0}^{n}(u_{n+1} - u_i)}. \tag{12}$$

In what follows now, $I_j(x)$ denotes the degree at most j polynomial that interpolates f at $u_0, u_1, \ldots, u_j$. So $I_0(t)$ is a constant, namely

$$I_0(x) = c_0 = f(u_0).$$

Using (11) with $n = 0$,

$$I_1(x) = c_0 + c_1(x - u_0);$$

and from (12), $c_1 = \frac{f(u_1)-c_0}{u_1-u_0}$. Using the above in (11), with $n = 1$,

$$I_2(x) = c_0 + c_1(x - u_0) + c_2(x - u_0)(x - u_1).$$

c_0 and c_1 were already found; c_2 is obtained from (12). Continuing in this way, we obtain Newton's form of the interpolating polynomial:

$$I_n(x) = \overbrace{\underbrace{c_0 + c_1(x - u_0)}_{I_1(x)} + c_2(x - u_0)(x - u_1)}^{I_2(x)} + \cdots + c_n(x - u_0)(x - u_1)\cdots(x - u_{n-1})$$

which we abbreviate as

$$I_n(x) = c_0 + \sum_{i=1}^{n} c_i \left\{ \prod_{j=0}^{i-1} (x - u_j) \right\}. \tag{13}$$

ii. **Its evaluation cost:** We describe a "Horner-like" method to evaluate (13) efficiently. The following pseudo-code is a procedure "NEWTON". The inputs are n (the degree), an array $U[0, 1, \ldots, n]$ with the $n + 1$ collocation points, an array $c[0, 1, \ldots, n]$ with the $n + 1$ coefficients used in (13), and x, where I_n is being evaluated. The output is VAL which equals $I_n(x)$

NEWTON(n, U, c, x;**VAL)**

- **VAL**$\leftarrow c[n]$
- **FOR** $i = 1$ **TO** n **DO**
- **VAL** $\leftarrow$ **VAL**$*(x - U[n - i]) + c[n - i]$
- **ENDFOR**

Each traversal of the loop performs two + or - OPS and one * OP, a total of n multiplications and $2n$ additions or subtractions, *assuming we already "know" the coefficients* $c_0, c_1, \ldots, c_n$ in (13).

iii. **How to obtain it:** We get the coefficients $c_0, c_1, \ldots, c_n$ for Newton's form as follows. $c_0 = f(u_0)$ and $c_1 = (f(u_1)-f(u_0))/(u_1-u_0)$ always, and we now proceed inductively. Assuming we already know $c_0, \ldots, c_j$, $j \geq 1$, we will use (12) [with $n = j$] to obtain c_{j+1}, the next coefficient. This is possible because $I_j(u_{j+1})$ in (12) depends only on $c_0, \ldots, c_j$, which we have already obtained. The following pseudo-code describes a procedure "GETC". The inputs are $n \geq 1$, the degree, and an array $U[0, 1, \ldots, n]$ with the $n + 1$ collocation points. It outputs the array $c[0, 1, \ldots, n]$ of coefficients for Newtons form of I_n.

GETC$(n, U; c)$

- **c[0]**$\leftarrow f(U[0])$
- **c[1]**$\leftarrow (f(X[1] - f(X[0]))/(X[1] - X[0])$
- **FOR** $i = 2$ **TO** n **DO**
- **run NEWTON(**$i-1, U, c, U[i]$;**VAL)**
- **PROD** $\leftarrow U[i] - U[0]$
- **FOR** $j = 1$ **TO** $i - 1$ **DO**
- **PROD** $\leftarrow$ **PROD**$*(U[i] - U[j])$
- **ENDFOR**
- $c[i] \leftarrow (f(U[i])-$ **VAL**$)/$**PROD**
- **ENDFOR**

We count the total number of multiply or divide steps to compute the $c_i's$: To get c_{j+1} using (12) we note that (i) there are j multiplications and $j+1$ subtractions in the denominator; (ii) the numerator has 1 subtraction; (iii) one division is done to compute the ratio; (iv) finally, using NEWTON, we can evaluate $I_j(u_{j+1})$ in the numerator with j multiply steps and $2j$ additions or subtractions. Therefore the total work for c_{j+1} is $2j+1$ multiply or divide steps and $3j+1$ add or subtract steps. Summing from $j = 0, 1, \ldots, n-1$, the work to obtain all coefficients for I_n is n^2 multiply or divide steps, and $3n^2/2 + n/2$ add/subtract steps. Noticing that $c_0 = f(u_0)$ and that there is one evaluation of f in (12), we can add $n+1$ evaluations of f to the previous cost. [**Note:** The multiplications can be cut roughly in half using a technique called divided differences.]

4. **Runge's Example:** We want to approximate $f(x) = 1/(1+9x^2)$ on $[-1, 1]$. Let $I_n(x)$ be the interpolating polynomial based on $n+1$ evenly spaced collocation points $u_j = -1 + 2j/n$, $j = 0, 1, \ldots, n$. Examination of graphs of this approximation showed large errors near -1 and 1 which increased in size with n (see Handout 7). In fact Runge proved

$$d(\frac{1}{1+9x^2}, I_n(x)) \to \infty \text{ as } n \to \infty\,,$$

where $d(g,h) \equiv \max(|g(x) - h(x)|,\ a \le x \le b)$ (in our case $[a,b] = [-1,1]$). This shows that interpolation does not necessarily give better approximations as the number of collocation points increases (in fact *bad* collocation points can even produce approximations with errors that diverge to ∞, a disastrous result).

5. **Minimax (or "Best") Approximation:** Given a function f on $[a,b]$ and an integer $n > 0$, we want to approximate f by a polynomial of degree at most n. The quality of an approximating polynomial P_n is measured by its distance from f, namely

$$d(f, P_n) = \max\{|f(x) - P_n(x)|,\ a \le x \le b\}.$$

A polynomial $M_n(x)$ *is a "best" approximation of* f if $d(f, M_n) \le d(f, P_n)$ for every polynomial P_n of degree at most n (M_n has least distance from f; no n^{th} degree polynomial is closer). Therefore

$$d(f, M_n) = \min\{d(f, P_n) : P_n \text{ a polynomial of degree} \le n\} = \min_{P_n}\{\max_{a \le x \le b} |f(x) - P_n(x)|\}$$

$$= \min_{a_0,a_1,\ldots,a_n} [\max \{|f(x) - (a_0 + a_1 x + \cdots + a_n x^n)|,\ a \le x \le b\}].$$

The last equation indicates why M_n is also called the *"minimax" approximation*. About 100 years ago Chebycheff proved that

(a) M_n exist and is unique: i.e., $d(f, M_n) < d(f, P_n)$ for all polynomials $P_n \neq M_n$ of degree at most n;

(b) M_n interpolates f: i.e., there are at least $n+1$ collocation points $u_0 < \cdots < u_n$ in $[a, b]$ for which $f(u_i) = M_n(u_i)$, $i = 0, 1, \ldots, n$;

(c) M_n equi-oscillates: i.e., there are at least $n+2$ *oscillation points* $y_0 < \cdots < y_{n+1}$ in $[a, b]$ with $y_i < u_i < y_{i+1}$, $i = 0, \ldots, n$, for which

$$f(y_i) - M_n(y_i) = -[f(y_{i+1}) - M_n(y_{i+1})], i = 0, \ldots, n,$$

and in fact $d(f, M_n) = |f(y_i) - M_n(y_i)|$, $i = 0, \ldots, n$. Also if a degree at most n polynomial P_n equioscillates, it is minimax; i.e., $P_n = M_n$.

(d) If f is a polynomial of degree at most $n+1$ then the u_i are the Chebycheff points in $[a, b]$ and M_n is the Chebycheff interploation (see Topic 6, below).

The Runge example shows there is a disastrous way to choose collocation points. Property (b) shows there is an optimal way to choose them and no other polynomial approximation - interpolating or not - is as good.

- **Example 1:** Let $f(x) = \sin x$, $0 \le x \le \pi$ and take $n = 1$. The claim is that $M_1(x) = 1/2$. This line is above f by $1/2$ at $x = 0$, below f by $1/2$ at $x = \pi/2$, and above f by $1/2$ at $x = \pi$. Every other line will deviate from $\sin x$ by more than $1/2$ at one or more of these points. Checking the statements of Chebycheff's theorem for this example: (b) $u_0 = \pi/6$ and $u_1 = 5\pi/6$; (c) $y_0 = 0$, $y_1 = \pi/2$, $y_2 = \pi$ and $d(sinx, 1/2) = 1/2$.
- **Example 2:** Let $f(x) = x^2$, $-1 \le x \le 1$ and take $n = 1$. Again, $M_1(x) = 1/2$ is the minimax straight line approximation. Note that: (b) $f(x) = M_1(x)$ when $x^2 = 1/2$, or at $u_0 = -\sqrt{2}/2$ and $u_1 = \sqrt{2}/2$; (c) $y_0 = -1$, $y_1 = 0$, $y_2 - 1$ and $d(x^2, 1/2) = 1/2$; (d) M_1 IS the degree 1 Chebycheff interpolation for f (see Topic 6, below).
- **Example 3:** Let $f(x) = x^4$, $-1 \le x \le 1$ and take $n = 1$. Once again, $M_1(x) = 1/2$ is the minimax straight line approximation. Note that: (b) $f(x) = M_1(x)$ when $x^4 = 1/2$, or at $u_0 = -2^{-1/4}$ and $u_1 = 2^{-1/4} = .8408964153...$; (c) $y_0 = -1$, $y_1 = 0$, $y_2 - 1$ and $d(x^4, 1/2) = 1/2$.
- **Example 4:** Let $f(x) = x^2$, $0 \le x \le 3$ and take $n = 1$. Because f is increasing and its slope is increasing, we will be able to use (c) to find M_1. First note that the line $y = 3x$ interpolates f at $x = 0$ and at $x = 3$. Next observe that the difference $h(x) = 3x - x^2$ has a unique maximum when $x = 3/2$, and that $h(3/2) = 9/4$. Therefore the line $M_1(x) = 3x - 9/8$ has property (c): it is below f by $9/8$ at $y_0 = 0$, above f by $9/8$ at $y_1 = 3/2$, and below f by $9/8$ at $y_2 = 3$ (we dropped $y = 3x$ "halfway" down towards its point of greatest deviation from f). Also $d(f, M_1) = 9/8$.

6. **Chebycheff Interpolation:** We want to approximate a function $f(x)$ defined on $[-1, 1]$. The $(n+1)$ Chebycheff points for $[-1, 1]$ are defined by

$$u_j = \cos\left[\left(\frac{2j+1}{n+1}\right)\frac{\pi}{2}\right],\ j = 0, 1, \ldots, n. \tag{14}$$

The polynomial $C_n(x)$ that interpolates f at the Chebycheff points is called the n^{th} Chebycheff interpolation of f. Plotting $f(x) = 1/(1+9x^2)$ and $C_n(x)$ on $[-1,1]$ suggests that $d(f, C_n) \to 0$ (see handout 8).

To do Chebycheff interpolation on an arbitrary interval $[a,b]$, we map $[-1,1]$ linearly onto $[a,b]$ by $x = a + \frac{(y+1)}{2}(b-a)$, $y \in [-1,1]$. The Chebycheff points on $[a,b]$ are

$$t_j = a + \frac{(u_j+1)}{2}(b-a), \tag{15}$$

the u_j being the Chebycheff points in $[-1,1]$ defined by (14). The polynomial C_n that interpolates f at these collocation points is the n^{th} Chebycheff interpolation.

(a) **An Example:** Let $f(x) = x^2$, $0 \le x \le 3$ and take $n = 1$ (i.e., we seek the linear Chebycheff interpolation). To find the Chebycheff points for $[0,3]$, (14) gives $u_0 = \cos(\pi/4) = \sqrt{(2)}/2$ and $u_1 = \cos(3\pi/4) = -\sqrt{(2)}/2$; from (15), $t_0 = 3/2 + 3\sqrt{(2)}/4$ and $t_1 = 3/2 - 3\sqrt{(2)}/4$. Therefore $C_1(x) = c_0 + c_1(x - t_0)$ (from (13)), where $c_0 = f(t_0)$ and $c_1 = (f(t_1) - f(t_0))/(t_1 - t_0)$ (from (12)) and

$$C_1(x) = (\frac{3}{2} + \frac{3\sqrt{2}}{4})^2 + 3(x - (\frac{3}{2} + \frac{3\sqrt{2}}{4})) = 3x - 9/8.$$

Referring to Example 4, above, we see that $C_1 = M_1$ in agreement with condition (d) in Chebycheff's theorem.

(b) **Convergence:** Chebycheff interpolation is always "good" because

$$d(f, C_n) \to 0\,,\ n \to \infty$$

by a theorem of M. Powell. Actually Powell proved that there is a constant α_n for which $d(f, C_n) < \alpha_n d(f, P_n)$ for any polynomial P_n of degree at most n; the coefficient α_n may be taken to be about $\log n$, and we can take $P_n = M_n$ to be the minimax, or best approximation to f. It is known that $d(f, M_n) < \beta/n^2$ for some $\beta > 0$, under quite general conditions, so we can say $d(f, C_n) < \beta \log n/n^2$ which converges to zero. (Again, its good to look at handout 8.)

7. **Least Squares Approximation:** We want to approximate a continuous function $f(x)$, defined on the interval $[a,b]$, by the "best" polynomial P_k of degree at most k. To express P_k we have $k+1$ basis functions $\phi_0(x), \phi_1(x), \ldots, \phi_k(x)$, where $\phi_i(x)$ is a polynomial of degree exactly $= i$. The most familiar case is the monomial basis where, for each $i = 0, 1, \ldots, k$,

$$\phi_i(x) = x^i.$$

Given a basis, *any* polynomial P_k of degree at most k has a unique representation

$$P_k(x) = a_0\phi_0(x) + a_1\phi_1(x) + \cdots + a_k\phi_k(x) = \sum_{i=0}^{k} a_i\phi_i(x). \tag{16}$$

Note that the standard form of the interpolating polynomial uses the monomial basis and Newton's form uses the basis where $\phi_0 = 1$ and

$$\phi_i(x) = \prod_{j=o}^{i-1}(x - u_j).$$

(a) **Continuous LSQ:** We want to choose coefficients in (16) to minimize

$$d(f,P_k) \equiv \int_a^b [f(x)-P_k(x)]^2 dx = \int_a^b [f(x)-\sum_{i=0}^{k} a_i\phi_i(x)]^2 dx. \tag{17}$$

The integral is a function $E(a_0,a_1,\ldots,a_k)$ of the $k+1$ unknown coefficients. Setting the partial derivative of E w.r.t. a_i equal to zero (a necessary condition for the min) we get

$$a_0\int_a^b \phi_0(x)\phi_i(x)dx + a_1\int_a^b \phi_1(x)\phi_i(x)dx + \cdots + a_k\int_a^b \phi_k(x)\phi_i(x)dx = \int_a^b f(x)\phi_i(x)dx.$$

Call this equation $(*)_i$, *a linear equation in the unknowns* $a_0,a_1,\ldots,a_k$. The equations in $(*)_i$ for $i=0,1,\ldots,k$, form the system called *the normal equations* for continuous least squares:

$$C\underline{a} = \underline{d},$$

where, from $(*)_i$, $\boxed{c_{ij} = \int_a^b \phi_i(x)\phi_j(x)dx}$ and $\boxed{d_i = \int_a^b f(x)\phi_i(x)dx}$, $i,j=0,1,\ldots,k$. The solution gives the minimizing choice of coefficients $a_0,a_1,\ldots,a_k$ in (17). With this choice, P_k in (16) is the continuous least-squares approximation to f on $[a,b]$, expanded in the basis $\phi_0,\phi_1,\ldots,\phi_k$. Note that C is symmetric; i.e., $c_{ij}=c_{ji}$. Also note that when $[a,b]=[0,1]$ and we are in the monomial basis,

$$c_{ij} = \int_0^1 \phi_i(x)\phi_j(x)dx = \int_0^1 x^i x^j dx = \frac{1}{i+j+1}\,,\ i,j=0,1,\ldots,k\,; \tag{18}$$

i.e., $C=H_k$, the Hilbert matrix of size $k+1$.

- **An Example:** Let $f(x)=x^2$, $0\le x\le 3$ and take $k=1$; i.e., we seek the continuous linear least-squares approximation to f. We will use the momonial basis $\phi_i(x)=x^i$. Since $c_{ij}=\int_0^3 x^i x^j dx$ and $d_i=\int_0^3 x^2 x^i dx$, the augmented coefficient matrix of the normal equations is

$$\left(\begin{array}{cc|c} 3 & 9/2 & 9 \\ 9/2 & 9 & 81/4 \end{array}\right).$$

The solution, $a_0=-3/2$, $a_1=3$, gives $P_1(x)=3x-3/2$ as the continuous least-squares straight line approximation to x^2 on $[0,3]$. (Compare to $C_1(x)=3x-9/8$, the Chebycheff Interpolation of f, which is also minimax).

(b) **Discrete LSQ:** Given $n+1$ data points $u_0,u_1,\ldots,u_n$ in $[a,b]$ and $k<n$ we want to choose coefficients $a_0,a_1,\ldots,a_k$ in (16) to minimize

$$d(f,P_k) \equiv \sum_{\ell=0}^{n}[f(x_\ell)-P_k(x_\ell)]^2 = \sum_{\ell=0}^{n}[f(x_\ell)-\sum_{i=0}^{k} a_i\phi_i(x_\ell)]^2. \tag{19}$$

The sum in (19) is a function $E(a_0,a_1,\ldots,a_k)$ of the coefficients. Setting the partial derivative of E w.r.t. a_i equal to zero (a necessary condition for the min) we get

$$a_0\sum_{\ell=0}^{n} \phi_0(x_\ell)\phi_i(x_\ell) + a_1\sum_{\ell=0}^{n} \phi_1(x_\ell)\phi_i(x_\ell) + \cdots + a_k\sum_{\ell=0}^{n} \phi_k(x_\ell)\phi_i(x_\ell) = \sum_{\ell=0}^{n} f(x_\ell)\phi_i(x_\ell).$$

Call this equation $(*)_i*$, a linear equation in the unknown $a_0,a_1,\ldots,a_k$. The equations $(*)_i$, $i=0,1,\ldots,k$, form the system called *the normal equations* for discrete least squares:

$$C\underline{a} = \underline{d},$$

where $\boxed{c_{ij} = \sum_{\ell=0}^{n} \phi_i(x_\ell)\phi_j(x_\ell)}$ and $\boxed{d_i = \sum_{\ell=0}^{n} f(x_\ell)\phi_i(x_\ell)}$, $i, j, = 0, 1, \ldots, k$. The solution gives the minimizing choice of coefficients $a_0, a_1, \ldots, a_k$ in (19). With this choice, P_k in (16) is the discrete least-squares approximation to f on $[a, b]$, expanded in the basis $\phi_0, \phi_1, .$ Note that C is symmetric; i.e., $c_{ij} = c_{ji}$.

- **An Example:** Let $f(x) = x^2$, take $k = 1$ and use points $x_0 = 0$, $x_1 = 1$, $x_2 = 2$, $x_3 = 3$. We seek P_1, the discrete LSQ straight line approximation to f, based on these 4 data points, and in the monomial basis. From the definition of c_{ij} and d_i, the augmented coefficient matrix of the normal equations is

$$\left(\begin{array}{cc|c} 4 & \sum_{\ell=0}^{3} x_\ell & \sum_{\ell=0}^{3} x_\ell^2 \\ \sum_{\ell=0}^{3} x_\ell & \sum_{\ell=0}^{3} x_\ell^2 & \sum_{\ell=0}^{3} x_\ell^3 \end{array} \right) = \left(\begin{array}{cc|c} 4 & 6 & 14 \\ 6 & 14 & 36 \end{array} \right).$$

  The solution, $a_0 = -1$, $a_1 = 3$, gives $P_1(x) = 3x - 1$ as the discrete least-squares straight line approximation to x^2 on $[0, 3]$, based on the 4 given data points.

(c) **Weighted (Discrete) LSQ:** In discrete least squares we may not have equal confidence in the values of f at the different data points (think of $f(t)$ as a "measurement" dependent on t, some measurements being more reliable than others). To reflect our desire to have the reliable points influence the approximation more than the unreliable ones, we give weight $w_i \geq 0$ to the point x_i (if $w_i = c$, for all $i = 0, 1, \ldots, n$, then all points are equally reliable). Given $k < n$ we want to choose coefficients $a_0, a_1, \ldots, a_k$ in (16) to minimize

$$d(f, P_k) \equiv \sum_{\ell=0}^{n} w_\ell \left[f(x_\ell) - P_k(x_\ell)\right]^2 = \sum_{\ell=0}^{n} w_\ell \left[f(x_\ell) - \sum_{i=0}^{k} a_i \phi_i(x_\ell)\right]^2. \tag{20}$$

As in the unweighted case, the optimal coefficients can be shown to satisfy *normal equations for weighted least squares*, namely

$$C\underline{a} = \underline{d},$$

where $\boxed{c_{ij} = \sum_{\ell=0}^{n} w_\ell \phi_i(x_\ell)\phi_j(x_\ell)}$ and $\boxed{d_i = \sum_{\ell=0}^{n} w_\ell f(x_\ell)\phi_i(x_\ell)}$, $i, j, = 0, 1, \ldots, k$; notice that if all weights are equal, we get the solution to the (unweighted) least squares. In the previous discrete least squares example, suppose we regard the data at $x_3 = 3$ to have one third the reliability of the other data. This corresponds to weights $w_0 = w_1 = w_2 = 3$ and $w_3 = 1$. Setting up, then solving the normal equations to minimize (20) gives $P_1(t) = -3/4 + 2\frac{5}{8}t$ as the best line. Observe that it "respects" the data at $x_3 = 3$ less than in the unweighted case in the sense that P_1 permits a larger error at that point.

8. **Orthogonal Bases:** To solve the $k+1$ normal equations for continuous least squares, $C\underline{a} = \underline{d}$, we expect to do $(k + 1)^3/3$ multiplications. Moreover, if we are in the monomial basis on $[0, 1]$, (18) shows that C is the Hilbert matrix, so we can expect very large roundoff errors in computing $a_0, a_1, \ldots, a_k$, the coefficients of P_k.

   Suppose our basis functions satisfied

$$\int_a^b \phi_i(x)\phi_j(x)dx = 0\,,\ i \neq j.$$

   A basis with this property is called orthogonal. In this case the matrix C in the normal equations is diagonal. The advantages are:

- It is easy to compute the solution ($k+1$ divisions).
- There is no propagation of roundoff error.

(a) **Legendre Basis on** $[-1,1]$**:** Let $L_0(x)=1$ and $L_1(x)=x$ and define

$$L_{k+1}(x)=\frac{2k+1}{k+1}xL_k(x)-\frac{k}{k+1}L_{k-1}(x)\,,\ k\geq 1. \tag{21}$$

L_i is the i^{th} Legendre function. The first few are (using (21)): $L_2(x)=(3x^2-1)/2$; $L_3(x)=(5x^3-3x)/2$; $L_4(x)=(35x^4-30x^2+3)/8$. Note that for $i\leq 4$, L_i is a polynomial of degree $=i$. Therefore, (20) implies that for any k, L_k is a polynomial of degree $=k$. It can be shown by induction that $\int_{-1}^{1}L_i(x)L_j(x)dx=0$, $i\neq j$ (verify it for the specific cases, above). Therefore the Legendre functions form an orthogonal basis on $[-1,1]$.

(b) **Arbitrary** $[a,b]$**:** To get an orthogonal basis on $[a,b]$ we just map $[-1,1]$ linearly onto $[a,b]$ and take the image of the Legendre functions: Specifically, letting

$$y=-1+2(x-a)/(b-a)$$

be the linear transformation, define

$$\phi_i(x)=L_i(y)=L_i(-1+\frac{2(x-a)}{b-a}). \tag{22}$$

Checking orthogonality,

$$\int_a^b\phi_i(x)\phi_j(x)dx=\int_a^b L_i(-1+\frac{2(x-a)}{b-a})L_j(-1+\frac{2(x-a)}{b-a})dx=\frac{b-a}{2}\int_{-1}^{1}L_i(y)L_j(y)dy,$$

which is zero when $i\neq j$.

(c) **The example again:** As before, $f(x)=x^2$, $0\leq x\leq 3$. From (22) the first few basis functions are $\phi_0(x)=L_0(y)=1$, $\phi_1(x)=L_1(-1+2x/3)=-1+2x/3$, $\boxed{\phi_2(x)}$ $=L_2(-1+2x/3)=3(-1+2x/3)^2/2-1/2=\boxed{1-2x+2x^2/3}$.

The normal equations for $P_1(x)$ are

$$\left(\begin{array}{cc|c}\int_0^3\phi_0(x)\phi_0(x)dx & \int_0^3\phi_0(x)\phi_1(x)dx & \int_0^3x^2\phi_0(x)dx\\ \int_0^3\phi_1(x)\phi_0(x)dx & \int_0^3\phi_1(x)\phi_1(x)dx & \int_0^3x^2\phi_1(x)dx\end{array}\right)=\left(\begin{array}{cc|c}3&0&9\\0&1&9/2\end{array}\right).$$

Reading off the solution $a_0=3$, $a_1=9/2$, the least-squares straight line approximation to x^2, in the orthogonal basis, is

$$P_1(x)=3\phi_0(x)+9/2\phi_1(x)=3+9/2(-1+2x/3).$$

To express P_1 in in the monomial basis, simplify the above to $3x-3/2$, in agreement with the earlier, direct computation of P_1 in the monomial basis. Also note that once we know P_1 in the monomial basis, there is no need to set up and solve the normal equations for the orthogonal basis. Just write

$$P_1(x)=3x-3/2=a_0\phi_0(x)+a_1\phi_1(x)=a_0+a_1(-1+2x/3).$$

Equating the coefficients of x, $2a_1/3=3$. Equating the constant terms, $a_0-a_1=-3/2$. Therefore $a_1=9/2$ and $a_0=3$, as we discovered from the normal equations, above.