Approximation

(See CdB Ch. 6)

Purpose: Approximate a general function by a class of simpler functions.

There are two motivations for this:

1) Decompose a complicated function into its constituent simpler functions, in order to manipulate the function more easily (e.g., differentiation, integration, storage, etc.).

2) Recover a function from partial or noisy information (for example, we may have stored some values of the function in a table, or we may be observing or sensing the function at discrete times).

Applications:

- Signal compression and reconstruction (Fourier techniques)
- Data fitting (best line, best quadratic, ...)
- CAD representations of shapes

(Note: Approximation goes beyond interpolation.)

Uniform approximation by polynomials

Let us start by looking at polynomials again. This time, rather than interpolate at given fixed points, we will seek the best uniform approximation.

Given a function $f: [a,b] \to \mathbb{R}$ and a polynomial p we measure the error between f and p in terms of an L_∞ norm, that is,

$$\|f-p\|_\infty = \max_{a \le x \le b} |f(x) - p(x)|$$

A good uniform approximation is one for which this error is small. Recall from Weierstrass' theorem that (for continuous f) we can make this error arbitrarily small by choosing a polynomial p of high-enough degree.

Let's restrict the degree of the polynomial.

<u>Def</u> Let the set $\underline{\Pi_n}$ consist of all polynomials of degree at most n.

<u>Def</u> The <u>uniform</u> <u>distance</u> of f from Π_n is the smallest error achievable using polynomials in Π_n. We write:

$$d(f, \Pi_n) = \min_{p \in \Pi_n} \|f-p\|_\infty$$

The goal in best uniform approximation is to find a polynomial p that actually achieves this minimum possible error.

The following theorem helps us:

Th$^{\underline{m}}$ A function f which is continuous on $[a,b]$ has exactly one best uniform approximation on $[a,b]$ from Π_n.

The polynomial $p \in \Pi_n$ is the best uniform approximation to f on $[a,b]$

if and only if

(Aside: This phrasing permits even more points satisfying the alternating error condition.)

there are $n+2$ points $a \le x_0 < \cdots < x_{n+1} \le b$ such that

$$(-1)^i \left[f(x_i) - p(x_i) \right] = \varepsilon \|f-p\|_\infty \qquad i=0,\ldots,n+1$$

where $\varepsilon = \text{signum}\left[f(x_0) - p(x_0)\right]$.

(Here $x_0 = a$ and $x_{n+1} = b$ in case $f^{(n+1)}(x)$ does not change sign on $[a,b]$.)

In other words, with alternating sign at $n+2$ points the difference between f and p is precisely equal to the L_∞ distance between f and p.

The idea is to apply this theorem by constructing a polynomial p that satisfies the alternating-error-condition. By the theorem that polynomial p is the best uniform approximation

<u>Ex</u> Consider $f(x) = e^x$ on $[-1, 1]$.

Let us construct the best uniform approximation to e^x by a straight line over the interval $[-1, 1]$.

With $n=1$, the alternating-error-condition of the previous theorem tells us that qualitatively the best uniform approximation will look something like this:

The point is that there are three points

$x_0 = -1$
$x_1 = ?$
$x_2 = 1$

at which the error $f(x) - p(x)$ is greatest, with equal magnitude and alternating sign.

The trick is to figure out what the point x_1 is. To do this, let's try to figure out what the theorem tells us in detail.

Let's write the polynomial as $p(x) = a + bx$.

(since $f''(x)$ does not change sign on $[-1,1]$ we know that $x_0 = -1$ and $x_2 = 1$.)

Computing errors we find that:

$$e(x_0) = f(x_0) - p(x_0) = f(-1) - p(-1) = \frac{1}{e} - a + b$$
$$e(x_1) = f(x_1) - p(x_1) = e^{x_1} - a - bx_1$$
$$e(x_2) = f(x_2) - p(x_2) = f(1) - p(1) = e - a - b$$

The theorem says $e(x_0) = -e(x_1) = e(x_2) = \|f - p\|_\infty$

So
$$e(x_0) = e(x_2)$$
$$\frac{1}{e} - a + b = e - a - b$$
$$2b = e - \frac{1}{e}$$
$$b = \frac{e - \frac{1}{e}}{2} \approx 1.1752$$

This says that the slope of our best line must be the same as the average change of $f(x)$ over $[-1, 1]$

<u>How do we choose a?</u>

Well, by definition $\|f - p\|_\infty$ is the maximum error between f & p over $[-1,1]$. It is equal to $-e(x_1)$. So the error function $e(x)$ must achieve a local extremum at $x = x_1$,

In other words, $e'(x_1) = 0$.
We can use this fact to determine x_1.

$$e(x) = f(x) - p(x) = e^x - a - bx$$

so $e'(x) = e^x - b$
$$0 = e^{x_1} - b$$

$$\therefore \quad x_1 = \ln b$$
$$x_1 \approx .16144$$

Now using $e(x_1) = -e(x_2)$

$$e^{x_1} - a - bx_1 = -e + a + b$$

since $e^{x_1} = b$ (by our previous calculation), we see that

$$-a - bx_1 = -e + a$$

so $$a = \frac{e - bx_1}{2}$$
$$a \approx 1.2643$$

so the best line is $\boxed{p(x) \approx 1.2643 + 1.1752x}$

Just what is the maximum error?
Well, it occurs at $x_0, x_1, \& x_2$.
So, computing, we see

$$\boxed{\|f - p\|_\infty \approx .2788}$$

The previous example shows how one might construct the best uniform approximating polynomial of a certain degree n. But what if we're not sure what degree n we're interested in? We might want to approximately compute the number $d(f, \pi_n)$, in order to find a good n. After all, if $d(f, \pi_n)$ is too big, then we know that even the best approximating polynomial of degree n isn't very good. In that case, we might wish to look at higher degree polynomials.

<u>Can we estimate $d(f, \pi_n)$?</u>

Well, sort of. We <u>can</u> compute a <u>lower bound</u>. That's useful, in that it helps us eliminate degrees n for which $d(f, \pi_n)$ is too big.

Here's the procedure.

I'll only show you how to compute $d(f, \pi_1)$, that is, a lower bound for $d(f, \pi_1)$. But, the basic method generalizes to arbitrary n.

[Recall what $d(f, \pi_1)$ means. It is the smallest L_∞ error possible if one approximates $f(x)$ by linear polynomials.]

We start with a function $f: [a,b] \to \mathbb{R}$.

Recall the notation of divided differences, e.g. $f[x_0, x_1, x_2]$

Suppose $p \in \Pi_1$.

Let x_0, x_1, x_2 be any three points in the interval $[a, b]$
Since p is linear,

$$p[x_0, x_1, x_2] = 0.$$

And we can write

$$f[x_0, x_1, x_2] = \frac{f(x_0)}{(x_0-x_1)(x_0-x_2)} + \frac{f(x_1)}{(x_1-x_0)(x_1-x_2)} + \frac{f(x_2)}{(x_0-x_2)(x_1-x_2)}$$

So

$$\begin{aligned} f[x_0, x_1, x_2] &= f[x_0, x_1, x_2] - p[x_0, x_1, x_2] \\ &= (f-p)[x_0, x_1, x_2] \\ &= \frac{f(x_0)-p(x_0)}{(x_0-x_1)(x_0-x_2)} + \frac{f(x_1)-p(x_1)}{(x_1-x_0)(x_1-x_2)} + \frac{f(x_2)-p(x_2)}{(x_0-x_2)(x_1-x_2)} \\ &= \frac{f(x_0)-p(x_0)}{w'(x_0)} + \frac{f(x_1)-p(x_1)}{w'(x_1)} + \frac{f(x_2)-p(x_2)}{w'(x_2)} \end{aligned}$$

where $w(x) = (x-x_0)(x-x_1)(x-x_2)$

(So $w'(x) = (x-x_1)(x-x_2) + (x-x_0)(x-x_2) + (x-x_0)(x-x_1)$.)

So $$\left|f[x_0,x_1,x_2]\right| \le \|f-p\|_\infty \left(\frac{1}{|w'(x_0)|}+\frac{1}{|w'(x_1)|}+\frac{1}{|w'(x_2)|}\right)$$

So $$\|f-p\|_\infty \ge \frac{\left|f[x_0,x_1,x_2]\right|}{\frac{1}{|w'(x_0)|}+\frac{1}{|w'(x_1)|}+\frac{1}{|w'(x_2)|}}$$

First consider the left hand side of this inequality. The polynomial p is arbitrary (within π_1). None of its coefficients appear in the right hand side. We can therefore conclude that

$$d(f,\pi_1) = \min_{p\in\pi_1}\|f-p\|_\infty \ge \frac{f[x_0,x_1,x_2]}{\frac{1}{|w'(x_0)|}+\frac{1}{|w'(x_1)|}+\frac{1}{|w'(x_2)|}}$$

Now look at the right hand side of this inequality. It depends only on f and on $x_0, x_1, \& x_2$. Furthermore, the points x_0, x_1, x_2 are arbitrary. So, we can conclude that

$$d(f,\pi_1) \ge \max_{x_0,x_1,x_2} \frac{f[x_0,x_1,x_2]}{\frac{1}{|w'(x_0)|}+\frac{1}{|w'(x_1)|}+\frac{1}{|w'(x_2)|}}$$

This is the lower bound we are seeking. Of course, determining the max is usually too difficult. Instead it is often easier simply to pick three points x_0, x_1, x_2 and compute the resulting quotient. This, as we saw, is certainly also a lower bound for $d(f,\pi_1)$, though perhaps not the tightest possible.

<u>Ex</u> let's try this out on our example $f(x) = e^x$, on $[-1,1]$. Taking $x_0 = -1$, $x_1 = 0$, $x_2 = 1$, we get

$$f[x_0, x_1, x_2] = \tfrac{1}{2} f(-1) - f(0) + \tfrac{1}{2} f(1)$$

and $\dfrac{1}{|w'(x_0)|} + \dfrac{1}{|w'(x_1)|} + \dfrac{1}{|w'(x_2)|} = \dfrac{1}{2} + 1 + \dfrac{1}{2} = 2$

So $$d(f, \pi_1) \geq \frac{f(-1) - 2f(0) + f(1)}{4}$$

(Notice this is true for all f whose interval of definition includes $[-1,1]$. Now let's plug in for $f(x) = e^x$.)

So $$d(e^x, \pi_1) \geq \frac{e^{-1} - 2e^0 + e^1}{4}$$

$$\approx .2715$$

<u>Comments:</u>

- Notice that this lower bound is pretty close to the best error possible, namely .2788
- This lower bound tells us that we can't expect to approximate e^x with a line that gives decimal place accuracy. If we want better than about $\pm .3$ accuracy, we need higher-order approximations.

<u>Ex</u> Consider the function $f(x) = x^{n+1}$ on the interval $[-1,1]$. Suppose we wanted to approximate this degree $n+1$ polynomial with polynomials of degree at most n.

<u>Comment:</u> There is no real reason we would want to do this, but it will turn out that the best uniform approximation can actually be obtained by interpolation!

<u>Why is that nice?</u>

Well, computing the best uniform approximation to a function f by polynomials in π_n can be quite difficult.

Interpolation is easy.

So, if we can find interpolation points whose interpolating polynomial is "almost best" (in the uniform sense), then we've made life a lot easier.

Recall the Chebyshev polynomials:

degree k: $\quad T_k(\cos\theta) = \cos k\theta$

and in general we have the recurrence

$$T_{k+1}(x) = 2x\,T_k(x) - T_{k-1}(x), \quad k = 1, 2, \ldots$$

with $\quad T_0(x) = 1$ and $T_1(x) = x$

<u>Observe:</u> (i) $|T_k(x)| \le 1$ for all $x \in [-1,1]$

(ii) The leading coefficient of T_k is 2^{k-1}, $k=1,2,\ldots$

(iii) $T_k(x) = \pm 1$ alternatingly at the $k+1$ points

$$x_j = \cos\frac{k-j}{k}\pi, \quad j=0,\ldots,k$$

[Why? Because $T_k(x_j) = T_k\left(\cos\frac{k-j}{k}\pi\right)$

$$= \cos((k-j)\pi)$$

$$= (-1)^{k-j}.\ ]$$

It follows that $\overset{\text{define } p_n(x)=}{x^{n+1} - 2^{-n}T_{n+1}(x)}$ is a polynomial of degree at most n. Call it $p_n(x)$.

Observe that $$x^{n+1} - p_n(x) = 2^{-n}\underbrace{T_{n+1}(x)}_{=\pm 1 \text{ at } n+2 \text{ points (alternatingly)}}$$

The right hand side satisfies the alternating-error-condition (i.e. alternating error at $n+2$ points, equal to the maximum possible error between f & p_n) of our theorem. ($2^{-n}T_{n+1}(x)$ hits $\pm$ max $n+2$ times)

So, p_n must be the best uniform approximation!

<u>Conclusions:</u> 1) $d(x^{n+1}, \pi_n) = \frac{1}{2^n}$

2) Since $T_{n+1}(x)$ is zero at the $n+1$ Chebyshev points $\xi_{k,n+1} = \cos\frac{2k+1}{2n+2}\pi$, $k=0,\ldots,n$, we see that $p_n(x)$ is the polynomial that interpolates x^{n+1} at $\xi_{0,n+1}, \ldots, \xi_{n,n+1}$.

So, amazingly, for the specific function $f(x) = x^{n+1}$ we can obtain the best uniform approximation by polynomials of degree at most n simply by interpolating!

⋮

As we mentioned back when we were studying interpolation, it turns out that for arbitrary f, interpolating at the Chebyshev points or the expanded Chebyshev points comes very close to producing the best uniform approximation. Amazing.

See CdB, pp. 242-244 for further details.

Data Fitting

Now imagine that we have taken measurements of some unknown function f at points $x_1, \ldots, x_n$. We would like to "reconstruct" this function as well as we can.

(Some applications should pop into your head: surface reconstruction, signal recovery, and even learning, say of observed robot dynamics.)

Now, let's refer to our n measurements as f_i, $i = 1, \ldots, n$. If the measurements were perfect, then for each i we would have $f_i = f(x_i)$. Generally, however, the measurements have been corrupted with noise or other errors, so that

$$f_i = f(x_i) + \varepsilon_i \quad \text{, where } \varepsilon_i \text{ is some unknown error.}$$

Nonetheless, we would like to try to recover $f(x)$. Were it not for the measurement errors, we might consider using interpolation. But there's no point in forcing an interpolating polynomial to pass exactly through points (x_i, f_i) that are errorful. Indeed, the polynomial may just wiggle around a lot to pass through all the points (x_i, f_i) and thus have an order much higher than $f(x)$ really has.

Ex Let's look again at our old friend, the function $f(x) = (x-1)^2$. Suppose we have taken measurements at 7 points, evenly spaced over the interval $[-1, 2]$. The following table lists the points x_i, the true values $f(x_i)$, and our (errorful) measurements.

x_i	$f(x_i)$	f_i
-1	4	4.1
$-\frac{1}{2}$	2.25	2.3
0	1	1.05
$\frac{1}{2}$	.25	.20
1	0	.05
$\frac{3}{2}$	.25	.26
2	1	.90

The function $f(x)$ is a quadratic. If we interpolated through the points $\{(x_i, f_i)\}$ we would get a 6th order polynomial. No doubt that polynomial would contain some twists and turns that $f(x)$ does not (in other words, even if we get lucky and the interpolating polynomial matches $f(x)$ well, its derivatives may not).

Note: Of course, there are many functions $\{g(x)\}$ that could produce the measured values f_i above. How could we possibly hope to recover the correct function (in this case $(x-1)^2$) without further information we can't expect to recover

the correct one. Suppose, however, that we know the underlying function is quadratic. (Or maybe we just decide that we will only look at quadratic functions.)

Then we could pick as <u>basis functions</u> the functions

$$1 \quad , \quad x \quad , \quad x^2$$

and seek to find coefficients a, b, c such that

$$f(x) = a \cdot 1 + b \cdot x + c \cdot x^2$$

(Note: other basis functions are possible and indeed sometimes desirable. The key is that they are independent and span the vector space of quadratic functions. Later we'll also look at orthogonal bases.)

<u>If</u> our measurements f_i were perfect, then we would know

$$(*) \qquad f_i = a + b x_i + c x_i^2 \quad , \quad i = 1, \ldots, n$$

This $n \times 3$ system of equations in the variables a, b, c has a unique solution if the f_i are perfect.

If the f_i are imperfect, then the system $(*)$ is generally overconstrained. — But we could still obtain a least-squares solution, using for example SVD!

That's exactly the approach we will take. We will now look at the general formulation of this simple idea.

Suppose we are given n measurements (x_i, f_i).
We would like to reconstruct the function $f(x)$.

In practice we have some parameterized family of functions

$$F(x) = F(x; c_1, \dots, c_k).$$

We then choose the parameters $c_1, \dots, c_k$ based on the observations $\{(x_i, f_i)\}$ in such a way that

$$F(x) \text{ is "close to" } f(x).$$

Often, for mathematical simplicity we write $F(x)$ as a linear combination of some k basis functions:

$$F(x) = c_1 \phi_1(x) + \dots + c_k \phi_k(x).$$

This is what we did in the example before.
Again, the objective is to choose the $\{c_i\}$ well.

<u>Comment:</u> Why do we do things this way?

The point is that k will generally be much smaller than n. So rather than retain all n pieces of data (as with interpolation), most of which really contains little information (due to errors), we try to extract the important or useful information. That information is encoded in the k parameters $\{c_j\}$.

Next question: How does one choose the $\{c_j\}$?

Ideally we would like to minimize the difference between $f(x)$ and $F(x)$. Unfortunately, we don't know $f(x)$, so instead we simply minimize the difference between f & F at the data points $x_1, \ldots, x_n$.

Even then, we could use a variety of norms by which to measure the error:

continuous version

∞-norm: $$\|f - F\|_\infty = \max_{1 \le i \le n} |f_i - F(x_i)| \qquad \max_{a \le x \le b} |f(x) - F(x)|$$

1-norm: $$\|f - F\|_1 = \sum_{i=1}^{n} |f_i - F(x_i)|$$

p-norm: $$\|f - F\|_p = \sqrt[p]{\sum_{i=1}^{n} |f_i - F(x_i)|^p} \qquad \sqrt[p]{\int_a^b |f(x) - F(x)|^p dx}$$

Recall that the parameters $\{c_j\}$ are hidden in this notation. In fact, $F(x_i) = F(x_i; c_1, \ldots c_k)$. For each norm the goal would be to choose the $\{c_j\}$ so as to make the error a minimum. — This minimization process tends to lead to non-linear equations in $c_1, \ldots, c_k$, even when $F(x)$ has the simple linear form $F(x) = c_1\phi_1(x) + \cdots + c_k\phi_k(x)$.

However, as we shall see, in the case of a 2-norm, the error minimization leads to linear equations that determine $c_1, \ldots, c_k$ (assuming $F(x)$ itself has a simple linear form). For this practical reason 2-norms, that is, least squares are

so popular. In other words, we want to choose the $\{c_j\}$ to minimize the quantity

$$\|f - F\|_2 = \sqrt{\sum_{i=1}^{n} \left| f_i - F(x_i; c_1, \dots, c_k) \right|^2}$$

<u>Least Squares</u>

It is enough to choose $c = (c_1, \dots, c_k)$ to minimize the function

$$E(c) = \sum_{i=1}^{n} \left(f_i - F(x_i; c) \right)^2$$

At the point $c \in \mathbb{R}^k$ where $E(c)$ attains a minimum it must be the case that each of the partials vanishes, that is,

$$\frac{\partial}{\partial c_j} E(c) = 0 \quad , \; j = 1, \dots, k$$

Computing, we see that

$$\frac{\partial}{\partial c_j} E(c) = -2 \sum_{i=1}^{n} \left(f_i - F(x_i; c) \right) \frac{\partial}{\partial c_j} F(x_i; c)$$

Given that $F(x; c) = \sum_{j=1}^{k} c_j \, \phi_j(x)$

we have $\frac{\partial}{\partial c_j} F(x_i; c) = \phi_j(x_i)$ (a number)

Therefore $\frac{\partial}{\partial c_j} E(c) = -2 \sum_{i=1}^{n} \left(f_i - F(x_i; c) \right) \phi_j(x_i)$

When $E(c)$ is a minimum we therefore have the system:

$$\sum_{i=1}^{n} \left(f_i - F(x_i; c) \right) \phi_j(x_i) = 0 \qquad j = 1, \dots, k.$$

This system of equations is called the system of <u>normal equations</u>.

Why this name?
Well, consider the error vector

$$\vec{e} = \begin{pmatrix} f_1 - F(x_1; c) \\ \vdots \\ f_n - F(x_n; c) \end{pmatrix}$$

and consider the k vectors

$$\vec{\phi}_j = \begin{pmatrix} \phi_j(x_1) \\ \vdots \\ \phi_j(x_n) \end{pmatrix} \qquad j = 1, \dots, k$$

$\vec{e}$ measures the error at the n data points.
$\vec{\phi}_j$ encodes the value of the basis function ϕ_j at the n data points

The normal equations can be written compactly as:

$$(*) \qquad \vec{e}^{\,T}\vec{\phi}_j = 0 \qquad j = 1, \dots, k$$

In other words, at the minimizing c, the error vector $\vec{e}$ is perpendicular to each of the vectors $\vec{\phi}_j$ (all these vectors reside in $\mathbb{R}^n$, the space of possible data values at the points $x_1, \dots, x_n$).

This is classic least squares. Remember SVD? If you can't get an exact solution, you get as close as possible, meaning the error vector is perpendicular to the space of possible solutions.

Let's write

$$\vec{f} = \begin{pmatrix} f_1 \\ \vdots \\ f_n \end{pmatrix} \quad \& \quad \vec{F} = \begin{pmatrix} F(x_1; c) \\ \vdots \\ F(x_n; c) \end{pmatrix}$$

so $\vec{e} = \vec{f} - \vec{F}$. Then we have the following picture (this is supposed to be in $\mathbb{R}^n$):

This is a k-dimensional subspace of $\mathbb{R}^n$, spanned by the vectors $\vec{\phi}_1, \dots, \vec{\phi}_k$

(We assume $\vec{\phi}_1, \dots, \vec{\phi}_k$ are linearly independent.)

$$\vec{F} = c_1\vec{\phi}_1 + \dots + c_k\vec{\phi}_k$$

In other words, $\vec{F}$ (our solution vector) is simply the <u>orthogonal projection</u> of $\vec{f}$ (our data vector) onto the hyperplane spanned by the vectors $\vec{\phi}_1, \dots, \vec{\phi}_k$ (our space of permissible function vectors).

That's what the normal equations say. (See (*) again.)

<u>Purpose:</u> If we have picked our basis functions $\phi_1, \dots, \phi_k$ well, then we hope that the effect of this orthogonal projection is to remove noise and reconstruct the function $f(x)$.

<u>So, how do we solve the normal equations?</u>

1) If we expand (*) using $F(x_i; c) = c_1 \phi_1(x_i) + \dots + c_k \phi_k(x_i)$ we get the linear system of equations

$$Pc = q$$

where P is a $k \times k$ matrix with entries

$$P_{\ell j} = \vec{\phi}_\ell^{\,T} \vec{\phi}_j = \sum_{i=1}^{n} \phi_\ell(x_i)\, \phi_j(x_i),$$

q is the $k \times 1$ vector with entries

$$q_\ell = \vec{f}^{\,T} \vec{\phi}_\ell = \sum_{i=1}^{n} f_i\, \phi_\ell(x_i),$$

and c is the $k \times 1$ vector

$$c = \begin{pmatrix} c_1 \\ \vdots \\ c_k \end{pmatrix}.$$

It turns out that the system $Pc = q$ always has a solution. However, the matrix P is often very ill-conditioned, leading to unreliable results.

So, despite being able to set up a solvable system, it is often better to proceed as we did with our example on page 16, leading to the following method:

2) Recall that we concluded the normal equations were telling us to perform an orthogonal projection in $\mathbb{R}^n$.

Consider the $n \times k$ matrix A with entries

$$A_{ij} = \phi_j(x_i)$$

In other words, the jth column of A is the vector $\vec{\phi}_j$.

So, $\vec{F} = Ac$, $c = \begin{pmatrix} c_1 \\ \vdots \\ c_k \end{pmatrix}$.

Now consider the (overconstrained) system

$$Ac = \vec{f}$$

If we solve this system using SVD we will indeed be projecting $\vec{f}$ orthogonally onto $\text{span}(\vec{\phi}_1, \ldots, \vec{\phi}_k)$. The resulting vector c determines our best approximation.

<u>Ex</u> Let's return to our example of pages 15 & 16.

Our basis functions are $\phi_1(x) = 1$

$\phi_2(x) = x$

$\phi_3(x) = x^2$

We seek coefficients c_1, c_2, c_3 (we used a, b, c earlier) such that $F(x) = c_1 + c_2 x + c_3 x^2$ is the best least squares approximation to the data $\{(x_i, f_i)\}$. We have

$$\vec{f} = \begin{pmatrix} 4.1 \\ 2.3 \\ 1.05 \\ 0.2 \\ 0.05 \\ 0.26 \\ 0.9 \end{pmatrix}, \quad \vec{\phi}_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \quad \vec{\phi}_2 = \begin{pmatrix} -1.0 \\ -0.5 \\ 0 \\ 0.5 \\ 1.0 \\ 1.5 \\ 2.0 \end{pmatrix}, \quad \vec{\phi}_3 = \begin{pmatrix} 1.0 \\ 0.25 \\ 0 \\ 0.25 \\ 1.0 \\ 2.25 \\ 4.0 \end{pmatrix}$$

1) If we use the method $Pc = q$, we would solve

$$\begin{pmatrix} 7 & 3.5 & 8.75 \\ 3.5 & 8.75 & 11.375 \\ 8.75 & 11.375 & 23.1875 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 8.86 \\ -2.91 \\ 8.96 \end{pmatrix}$$

2) If we use the SVD method, $Ac = \vec{f}$, we would solve

$$\begin{pmatrix} 1 & -1.0 & 1.0 \\ 1 & -0.5 & 0.25 \\ 1 & 0 & 0 \\ 1 & 0.5 & 0.25 \\ 1 & 1.0 & 1.0 \\ 1 & 1.5 & 2.25 \\ 1 & 2.0 & 4.0 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 4.1 \\ 2.3 \\ 1.05 \\ 0.2 \\ 0.05 \\ 0.26 \\ 0.9 \end{pmatrix}$$

By either means we find that

$$c_1 \approx 1.044$$
$$c_2 \approx -2.044$$
$$c_3 \approx 0.995$$

So, our estimated polynomial is

$$p(x) = 1.044 - 2.044x + 0.995x^2$$

This is not too far from the true

$$f(x) = 1 - 2x + x^2$$

Page 26 compares the graphs of $p(x)$ and $f(x)$
Notice that there is a small persistent error between $p(x)$ & $f(x)$
However, the shape of p & f agree well, since we restricted ourselves to a quadratic subspace of the space of functions.
Thus the derivatives will be reasonable as well.

In contrast, suppose we constructed the interpolating polynomial $p_6(x)$ that matches the f_i values at the x_i, $i=1,\dots,7$.
See page 27.
While we get agreement between $p_6(x)$ and $f(x)$ more often, the maximum error is substantially larger than it was for $p(x)$ and $f(x)$. Furthermore, being of higher order and by matching the spurious data f_i exactly, $p_6(x)$ wiggles around a lot more than do $p(x)$ or $f(x)$. Thus the general "shape" is not as good.

f(x) – true function

p(x) – least-squares solution. (best quadratic fit to measurements

f(x) – true function

P6(x) – interpolating polynomial that passes through measured data exactly.

Orthogonality

We have seen how important orthogonal projection and orthogonal decomposition are. Indeed, orthogonality arises over and over, particularly in function approximation and decomposition. Let's take a closer look.

First, recall from linear algebra:

Suppose you have a vector space V with an inner product defined on it, denoted by $\langle \cdot, \cdot \rangle$.

Suppose you have an orthonormal basis for V:

$$\mathcal{B} = \{ b_1, \ldots, b_n \}$$ ($\mathcal{B}$ could be infinite if V is infinite-dimensional)

[Recall what this means: (i) The $\{b_i\}$ are linearly independent.
(ii) The $\{b_i\}$ span V.
(iii) $\langle b_i, b_j \rangle = \delta_{ij} = \begin{cases} 1 & \text{if } i=j \\ 0 & \text{if } i \neq j \end{cases}$.
]

Now let $v \in V$ be an arbitrary vector. Then $v = \sum_{i=1}^{n} \langle v, b_i \rangle b_i$. In other words, it is very easy to determine the coefficients of v relative to the basis $\mathcal{B}$; just take inner products.

Furthermore, if $\mathcal{U}$ is the subspace of V spanned by $b_1, \ldots, b_k$, then $w = \sum_{i=1}^{k} \langle v, b_i \rangle b_i$ is the vector in $\mathcal{U}$ that is "closest" to v, in that it minimizes $\|v-w\| = \sqrt{\langle v-w, v-w \rangle}$. This is of course the whole motivation for using least-squares; orthogonality makes life **easy**.

These ideas generalize to functions.
For instance, we might let V be the vector space of continuous integrable functions defined on some interval $[a,b]$.
(This is an infinite-dimensional vector space.)

<u>What is a good inner product for V?</u>

There are many, generally determined by applications.
We can describe these inner products by the rule

$$\langle g,h\rangle = \int_a^b g(x)\,h(x)\,w(x)\,dx$$

Here g & h are the two functions whose inner product we are taking.
$w(x)$ is a fixed positive function, called a weighting function.

Unfortunately, we often don't have information about a function at all points x. Instead, we may have information only at n discrete points $x_1, \dots, x_n$.
In that case one defines the inner product

$$\langle g,h\rangle = \sum_{i=1}^{n} g(x_i)\,h(x_i)\,w(x_i)$$

Strictly speaking this is not an inner product on V (the space of continuous integrable functions), but rather on $\mathbb{R}^n$ (the space of possible function values at the points $x_1, \dots, x_n$).
In practice one often forgets that distinction. (The bug is that a function may seem to be zero or seem to be orthogonal when in reality it is not.)

Why do we care?

Because there are two parts to good data fitting:

1. Selecting a good basis of functions $\mathcal{B}$.
2. Approximating a given function or observed data using the basis functions.

We have seen how to do 2, but we haven't yet talked about 1.

The discussion of the last two pages suggests that we find an orthogonal basis $\mathcal{B}$, for then we can approximate or decompose a function f by the rule

$$f \cong \sum_{g \in \mathcal{B}} \langle f, g \rangle g$$

(This is an equality if $f \in \text{span}(\mathcal{B})$. Otherwise it is an orthogonal projection of f onto $\text{span}(\mathcal{B})$.)

With this in mind, let us take a look at two important classes of bases, those given by orthogonal polynomials and those given by trigonometric polynomials.

Orthogonal polynomials

Caution: This definition depends implicitly on $[a,b]$.

Def An <u>orthogonal sequence of polynomials</u> is a sequence

$$p_0(x),\ p_1(x),\ p_2(x),\ \ldots \quad \text{(finite or infinite)}$$

such that $p_i(x)$ is a polynomial of degree i

and $\langle p_i, p_j \rangle = 0$ for all i & j with $i \neq j$.

If $\langle p_i, p_i \rangle = 1$ then we say the sequence is <u>orthonormal</u>.

(Here $\langle \cdot, \cdot \rangle$ is an inner product on functions, as discussed earlier.)

Some important facts:

Suppose $p_0(x), \ldots, p_k(x)$ is a finite sequence of orthogonal polynomials. Then:

1. If $p(x)$ is any polynomial of degree at most k then one can write $p(x) = d_0 p_0(x) + \cdots + d_k p_k(x)$, with the d_i uniquely determined. In particular

meaning coefficient of x^k, even if 0, here.

$$d_k = \frac{\text{leading coefficient of } p(x)}{\text{leading coefficient of } p_k(x)}$$

More generally,

$$d_i = \frac{\langle p, p_i \rangle}{\langle p_i, p_i \rangle}$$

2. If $p(x)$ is any polynomial of degree less than k, then

$$\langle p, p_k \rangle = 0.$$

3. If the inner product $\langle \cdot, \cdot \rangle$ is given by an integral of the form on page 29, then $p_k(x)$ has k simple real zeros in the interval (a,b).

This property also holds for the discrete inner product of page 29, so long as there are at least k distinct points among the $\{x_i\}$.

4. The orthogonal polynomials satisfy a three-term recurrence relation:

$$p_{i+1}(x) = A_i (x - B_i) p_i(x) - C_i p_{i-1}(x), \quad i = 0, 1, \ldots, k-1$$

where $A_i = \dfrac{\text{leading coefficient of } p_{i+1}}{\text{leading coefficient of } p_i}$, $\quad p_{-1}(x) \equiv 0$

$$S_i = \langle p_i, p_i \rangle$$

$$B_i = \frac{\langle x p_i, p_i \rangle}{S_i}$$

$$C_i = \begin{cases} \text{arbitrary} & \text{if } i = 0 \\ \dfrac{A_i S_i}{A_{i-1} S_{i-1}} & \text{if } i > 0 \end{cases}$$

We assume here that none of the S_i are zero.

If the polynomials are monic (leading coefficient is 1), then we can write the recurrence in the form

$$p_{i+1}(x) = \left[x - \frac{\langle x p_i, p_i \rangle}{\langle p_i, p_i \rangle} \right] p_i(x) - \frac{\langle p_i, p_i \rangle}{\langle p_{i-1}, p_{i-1} \rangle} p_{i-1}(x)$$

$i = 0, \ldots$

with $p_0(x) \equiv 1$

$p_{-1}(x) \equiv 0$ ⊛

The nice thing about this recurrence is that it lets us build the orthogonal sequence of polynomials iteratively.

For further details and some proofs of these properties, see CdB, § 6.3.

⊛ Really I mean $\dfrac{p_{-1}(x)}{\langle p_{-1}, p_{-1} \rangle}$ should be taken to be 0.

The only purpose in this definition is to be able to start the recurrence from $p_0(x) \equiv 1$, i.e., to be able to define $p_1(x)$ without worrying about $\dfrac{\langle p_0, p_0 \rangle}{\langle p_{-1}, p_{-1} \rangle} p_{-1}(x)$. After that one can define $p_2(x)$ based on $p_1(x)$ & $p_0(x)$, $p_3(x)$ based on $p_2(x)$ & $p_1(x)$, and so forth.

<u>Ex</u> Suppose the inner product is given by the rule

$$\langle g, h \rangle = \int_{-1}^{1} g(x)\, h(x)\, dx$$

This inner product yields the famous Legendre polynomials.

(Aside: This and many other similar inner products arise naturally in the solution of differential equations.)

Let's construct the orthogonal sequence of polynomials using the recurrence relation on page 33.

(0) Define $p_0(x) \equiv 1$. So $\langle p_0, p_0 \rangle = \int_{-1}^{1} 1 \cdot 1\, dx = 2$

$$\langle x p_0, p_0 \rangle = \int_{-1}^{1} x\, dx = 0$$

(1) So $p_1(x) = [x - 0]\, p_0(x) = x$

$$\langle p_1, p_1 \rangle = \int_{-1}^{1} x^2 = \frac{2}{3}$$

$$\langle p_1, p_1 \rangle = \int_{-1}^{1} x^3 = 0$$

(2) So $p_2(x) = [x - 0]\, p_1(x) - \frac{\frac{2}{3}}{2} p_0(x) = x^2 - \frac{1}{3}$

<u>Continuing this process we would get:</u>

$p_0(x) = 1$

$p_1(x) = x$

$p_2(x) = x^2 - \frac{1}{3}$

$p_3(x) = x^3 - \frac{3}{5}x$

$p_4(x) = x^4 - \frac{6}{7}x^2 + \frac{3}{35}$

<u>Ex</u> Calculate the polynomial of degree at most 3 that best approximates e^x over the interval $[-1,1]$ in the least-squares sense.

We want to minimize the error $\int_{-1}^{1} (e^x - p(x))^2 dx$, where $p(x)$ ranges over all polynomials of degree at most 3.

This error integral corresponds to the inner product of page 34. So, we will use the Legendre polynomials $p_0, \ldots, p_3$ as a basis. We obtain a best approximation $p(x)$ by orthogonally projecting e^x onto the subspace of functions spanned by $p_0, \ldots, p_3$.

In other words, $p(x) = \sum_{i=0}^{3} \frac{\langle e^x, p_i \rangle}{\langle p_i, p_i \rangle} p_i(x)$

We compute:

$$\langle p_0, p_0 \rangle = \int_{-1}^{1} 1 \, dx = 2$$

$$\langle p_1, p_1 \rangle = \int_{-1}^{1} x^2 dx = \frac{2}{3}$$

$$\langle p_2, p_2 \rangle = \int_{-1}^{1} \left(x^4 - \tfrac{2}{3}x^2 + \tfrac{1}{9}\right) dx = \frac{8}{45}$$

$$\langle p_3, p_3 \rangle = \int_{-1}^{1} \left(x^6 - \tfrac{6}{5}x^4 + \tfrac{9}{25}x^2\right) dx = \frac{8}{175}$$

$$\langle e^x, p_0 \rangle = \int_{-1}^{1} e^x \, dx = e - \frac{1}{e}$$

$$\langle e^x, p_1 \rangle = \int_{-1}^{1} e^x x \, dx = \frac{2}{e}$$

$$\langle e^x, p_2 \rangle = \int_{-1}^{1} e^x \left(x^2 - \frac{1}{3}\right) dx = \frac{2}{3} e - \frac{14}{3} \frac{1}{e}$$

$$\langle e^x, p_3 \rangle = \int_{-1}^{1} e^x \left(x^3 - \frac{3}{5} x\right) dx = -2e + \frac{74}{5} \frac{1}{e}$$

Let's write $p(x) = \sum_{i=0}^{3} d_i p_i(x)$ with $d_i = \dfrac{\langle e^x, p_i \rangle}{\langle p_i, p_i \rangle}$

Then $d_0 = \frac{1}{2}\left(e - \frac{1}{e}\right) \approx 1.175201194$

$d_1 = \frac{3}{2} \frac{2}{e} \approx 1.103638324$

$d_2 = \frac{45}{8}\left(\frac{2}{3} e - \frac{14}{3} \frac{1}{e}\right) \approx .536721526$

$d_3 = \frac{175}{8}\left(-2e + \frac{74}{5} \frac{1}{e}\right) \approx .1761390844$

So the best least squares polynomial is

$$p(x) = 1.175201194 p_0(x) + 1.103638324 p_1(x) + .536721526 p_2(x) + .1761390844 p_3(x)$$

$$= .9962940187 + .9979548734x + .536721526x^2 + .1761390844x^3$$

$p(x)$ differs from e^x by at most about $.0112$ on $[-1, 1]$.

Ex. Suppose we wanted the best linear approximation to e^x over $[-1,1]$ in the least-squares sense.

The previous calculation shows that this would be the line

$$p_{l.s.}(x) = d_0 p_0(x) + d_1 p_1(x)$$

$$= 1.175201194 + 1.103638324x$$

Compare this with the best uniform approximation

$$p_u(x) = 1.2643 + 1.1752x$$

(see page 6)

$p_{l.s.}$ and p_u are clearly different.

After all, they minimize two different error measurements. Indeed, the errors are:

	polynomial $p(x)$ \ error	Uniform error: $\max_{-1\le x\le 1} \lvert e^x - p(x)\rvert$	Least Squares Error: $\int_{-1}^{1} (e^x - p(x))^2 dx$
linear	p_u	0.2788	0.0719
linear	$p_{l.s.}$	0.4394	0.0527
cubic	p from page 36	0.0112	$\sim 2\cdot 10^{-5}$

$f(x) = e^x$

P_u is the best uniform approximation by a line over $[-1, 1]$.

$P_{l.s.}$ is the least squares approximation by a line over $[-1, 1]$.

Over the interval $[-1,1]$ $f(x) = e^x$ and its least-squares approximation by a cubic are nearly indistinguishable.

$f(x) = e^x$

$p(x)$ is the best (least-squares) approximation by a cubic over the interval $[-1, 1]$. Outside this interval the two functions diverge.

Trigonometric Polynomials

<u>Applications</u>: signal decomposition into constituent frequencies
signal recovery from constituent frequencies

Many functions in nature are periodic, that is,

$$f(x+\tau) = f(x) \quad \text{for some fixed } \tau,$$

called the <u>period</u> of f.

There is only one kind of periodic polynomial, and that is a constant. So, polynomials are not very good for approximating periodic functions. However, trigonometric polynomials are useful.

<u>Def</u> A <u>trigonometric polynomial of order n</u> is any function of the form

$$p(x) = \frac{a_0}{2} + \sum_{j=1}^{n} \left[a_j \cos \frac{2\pi j x}{\tau} + b_j \sin \frac{2\pi j x}{\tau} \right],$$

with the $\{a_i\}$ and $\{b_i\}$ either real or complex.
Note $p(x)$ has period τ.

<u>Note</u>: Often it is more convenient to write

$$p(x) = \sum_{j=-n}^{n} c_j e^{i \frac{2\pi j x}{\tau}} \qquad \left(i = \sqrt{-1} \right)$$

The connection is: $(j = 0, \dots, n)$

$$\begin{cases} a_j = c_j + c_{-j} & b_j = i(c_j - c_{-j}) \\ c_j = \frac{1}{2}(a_j - i b_j) & c_{-j} = \frac{1}{2}(a_j + i b_j) \end{cases}$$

The functions $1, e^{\pm i \frac{2\pi x}{\tau}}, e^{\pm i \frac{4\pi x}{\tau}}, e^{\pm i \frac{6\pi x}{\tau}}, \ldots$

form an orthonormal basis with respect to the inner product

$$\langle g, h \rangle = \frac{1}{\tau} \int_0^{\tau} g(x) \overline{h(x)} \, dx$$

[$\overline{h(x)}$ is the complex conjugate of $h(x)$.]

In other words, for "every" f we can write

$$(\ddagger) \qquad f(x) = \sum_{j=-\infty}^{\infty} \hat{f}(j) e^{ijx \frac{2\pi}{\tau}}$$ <u>Fourier Series</u>

where $\hat{f}(j) = \langle f, e^{ijx \frac{2\pi}{\tau}} \rangle$

$$= \frac{1}{\tau} \int_0^{\tau} f(x) e^{-ijx \frac{2\pi}{\tau}} \, dx$$

<u>Caution:</u> We have to be a little careful when we speak of a basis. We haven't yet said what the vector space is. The point is that the series in $(\ddagger)$ does not converge to $f(x)$ in all cases. But it does converge under "mild" conditions. It converges uniformly if f is continuous with a piecewise continuous first derivative. So our vector space includes those functio

In our general discussion about orthogonality on page 28, we described least-squares approximations as orthogonal projections onto vector subspaces.

In the context of trigonometric polynomials those projections are described by the following theorem:

$\boxed{\text{Th}^{\underline{m}}\ \text{The partial sum } \sum_{j=-n}^{n} \hat{f}(j)\, e^{ijx\frac{2\pi}{T}} \text{ of the Fourier series } (\ddagger) \text{ for } f(x) \text{ is the best approximation to } f(x) \text{ by trigonometric polynomials of order } n \text{ with respect to the norm } \|g\| = \sqrt{\frac{1}{T}\int_0^T |g(x)|^2 dx}.}$

Some of you may also remember

<u>Parseval's relation:</u> $\sum_j |\hat{f}(j)|^2 = \frac{1}{T}\int_0^T |f(x)|^2 dx$

$\uparrow$

power spectrum

Discretization

Generally it is difficult to compute the Fourier coefficients $\hat{f}(j)$, since one cannot compute the integral involved exactly. For instance, if $f(x)$ is not known exactly, or only at discrete points $\{x_i\}$, then one cannot compute the continuous integral. Even if $f(x)$ is known exactly, one may not be able to obtain a closed form solution, but instead be forced to use a numerical integrator.

Suppose, in fact, that we compute an integral of the form $\int_0^\pi g(x)\,dx$ using the composite trapezoid rule.

So
$$\int_0^\pi g(x)\,dx \approx \frac{\pi}{N}\sum_{n=0}^{N-1} g\left(\frac{\pi}{N}n\right)$$

given that $g(\pi) = g(0)$.

We might similarly compute integrals by discrete sums if the value of f is only known at the discrete points $x_n = \frac{\pi}{N}n$, $n = 0, \ldots, N-1$.

<u>Def</u> The points $\{x_n\}$ are called the <u>sampling points</u>, the values $\{f(x_n)\}$ are called the <u>sampling values</u>, and the number $\frac{\pi}{N}$ is called the <u>sampling interval</u>, $2\pi\frac{N}{\pi}$ is called the <u>sampling frequency</u>.

Discrete sampling leads to a discrete inner product:

<u>Def</u>

$$\langle g, h \rangle_N = \frac{1}{N} \sum_{n=0}^{N-1} g(x_n)\, \overline{h(x_n)} \quad , \text{where } x_n = \frac{\pi}{N} n.$$

- Observe that the functions $1, e^{\pm i \frac{2\pi x}{\pi}}, e^{\pm i \frac{4\pi x}{\pi}}, \ldots$ still satisfy a certain orthogonality property, namely

$$\left\langle e^{ikx\frac{2\pi}{\pi}}, e^{ijx\frac{2\pi}{\pi}} \right\rangle_N = \begin{cases} 1 & \text{if } k \equiv j \bmod N \\ 0 & \text{otherwise} \end{cases}$$

- Observe also that if we approximate $\hat{f}(j)$ by a numerical integration $\hat{f}_N(j)$ then we have

$$\hat{f}(j) = \langle f, e^{ijx\frac{2\pi}{\pi}} \rangle = \frac{1}{\pi} \int_0^{\pi} f(x) e^{-ijx\frac{2\pi}{\pi}} dx$$

$$\hat{f}_N(j) = \langle f, e^{ijx\frac{2\pi}{\pi}} \rangle_N = \frac{1}{N} \sum_{n=0}^{N-1} f(x_n) e^{-ijx_n\frac{2\pi}{\pi}}$$

The big question is:

<u>How well does $\hat{f}_N(j)$ approximate $\hat{f}(j)$?</u>

The answer to this question leads directly to the <u>Sampling Theorem</u> and the notion of <u>aliasing</u> (we see a hint of it already in the first observation above.)

Well, let's compute.

Let's suppose that the full Fourier series $\sum \hat{f}(j) e^{ijx\frac{2\pi}{T}}$ converges absolutely to $f(x)$ (this is not that strong a condition; it only requires the existence of the limit $\sum |\hat{f}(j)|$

Then:

$$\hat{f}_N(j) = \left\langle f, e^{ijx\frac{2\pi}{T}} \right\rangle_N$$

$$= \left\langle \sum_{k=-\infty}^{\infty} \hat{f}(k) e^{ikx\frac{2\pi}{T}}, e^{ijx\frac{2\pi}{T}} \right\rangle_N$$

$$= \sum_{k=-\infty}^{\infty} \hat{f}(k) \left\langle e^{ikx\frac{2\pi}{T}}, e^{ijx\frac{2\pi}{T}} \right\rangle_N$$

$$= \sum_{\substack{\text{all } k \text{ such that} \\ k = j \bmod N}} \hat{f}(k)$$

This is <u>aliasing</u>.

We cannot tell the difference between two basis functions $e^{ikx\frac{2\pi}{T}}$ and $e^{ijx\frac{2\pi}{T}}$ at the sampling points $\{x_n\}$ if $k = j \bmod N$. Therefore the Fourier coefficients at the frequencies for $k \neq j$ get mashed together and show up indistinguishably in the discrete Fourier coefficients.

In other words, the discrete Fourier coefficient $\hat{f}_N(j)$ is made up of all the exact Fourier coefficients $\hat{f}(k)$ whose corresponding functions $e^{ikx\frac{2\pi}{T}}$ cannot be distinguished from the function $e^{ijx\frac{2\pi}{T}}$ by the discrete inner product $\langle \cdot, \cdot \rangle_N$.

It is customary therefore to identify the function $e^{ijx\frac{2\pi}{T}}$ with the function $e^{i\tilde{j}x\frac{2\pi}{T}}$ where $\tilde{j}$ is chosen so that $j = \tilde{j} \bmod N$ and $|\tilde{j}|$ is as small as possible. This means that a given (angular) frequency $j\frac{2\pi}{T}$ is identified with the smallest frequency indistinguishable from it.

Consequently, in reconstructing a function $f(x)$ we must be careful not to count any Fourier coefficients $\hat{f}(k)$ more than once. Indeed, given observed values $f(x_0), \dots, f(x_{N-1})$ at the sampling points $x_n = \frac{T}{N}n$, $n = 0, \dots, N-1$, we compute the trigonometric polynomial

$$\boxed{p(x) = \sum_{\substack{j \text{ such that} \\ |j| < \frac{N}{2}}} \hat{f}_N(j)\, e^{ijx\frac{2\pi}{T}} + \mathrm{Re}\left[\hat{f}_N\left(\frac{N}{2}\right) e^{i\frac{N}{2}x\frac{2\pi}{T}}\right]}$$

this term is present only if N is even.

In other words, we only reconstruct frequencies $\frac{2\pi}{T}j$ with j in the range $|j| \le \frac{N}{2}$.

Consequently, we have derived a statement of the <u>Sampling Theorem</u>:

> If we wish to observe a certain periodic phenomenon of frequency ν, then we must sample at a frequency greater than 2ν.

Now suppose for simplicity that N is odd, so $N = 2k+1$, with k some positive integer. Observe that the functions $1, e^{\pm\frac{2\pi x}{\tau}}, e^{\pm\frac{4\pi x}{\tau}}, \ldots, e^{\pm\frac{2k\pi x}{\tau}}$ are orthonormal, with respect to the discrete inner product $\langle \cdot, \cdot \rangle_N$.

Following the reasoning on pages 19–23 (but replacing dot products with inner products), we obtain the following theorem:

> Th$^{\underline{m}}$ If $m \le k$, then $p_m(x) = \sum_{j=-m}^{m} \hat{f}_N(j)\, e^{ijx\frac{2\pi}{\tau}}$
>
> is the best approximation to $f(x)$ by trigonometric polynomials of order m, with respect to the discrete mean-square norm
>
> $$\|g\| = \sqrt{\frac{1}{N}\sum_{j=0}^{N-1} \left| g\left(\tfrac{\tau}{N} j\right)\right|^2}$$

<u>Corollary</u> If $m = k$, then $p_m(x) = p(x)$ as defined on p. 44.
A consequence of the theorem is:

<u>$p(x)$ interpolates $f(x)$ at the sampling points $\{x_i\}$.</u>

(This is the discrete analogue to the continuous Fourier series convergence