Lecture 20: Orthogonal Polynomials for Continuous Least Squares Problems

In the last lecture we saw how to reduce continuous least squares problems to systems of linear algebraic equations. In particular, we could expand polynomials in any basis $\{\phi_k\}_{k=0}^n$ for $\mathcal{P}_n$,

$$p = \sum_{k=0}^{n} c_k \phi_k,$$

and then solve the system

$$\begin{bmatrix} \langle \phi_0, \phi_0\rangle & \langle \phi_0, \phi_1\rangle & \cdots & \langle \phi_0, \phi_n\rangle \\ \langle \phi_1, \phi_0\rangle & \langle \phi_1, \phi_1\rangle & & \vdots \\ \vdots & & \ddots & \vdots \\ \langle \phi_n, \phi_0\rangle & \langle \phi_n, \phi_1\rangle & \cdots & \langle \phi_n, \phi_n\rangle \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} \langle f, \phi_0\rangle \\ \langle f, \phi_1\rangle \\ \vdots \\ \langle f, \phi_n\rangle \end{bmatrix}.$$

The monomial basis $\phi_k(x) = x^k$ can give poor numerical approximations even for fairly small values of n due to the fragility of the Hilbert matrix. Here we show how to construct a basis for $\mathcal{P}_n$ that proves to be more robust.

3.3.5. Orthogonal polynomials.

We say two vectors are orthogonal if their inner product is zero. The same idea leads to the notion of orthogonality of functions in $C[a, b]$. It will prove useful for us to generalize the notion of inner product introduced in §3.3.1. For any function $w \in C[a, b]$ with $w(x) > 0$ (actually, we can allow $w(x) = 0$ only on a set of measure zero), we define

$$\langle f, g\rangle = \int_a^b f(x)g(x)w(x)\,\mathrm{d}x.$$

One can confirm that this definition is consistent with the axioms required of an inner product that were enumerated in the last lecture. This inner product thus motivates the following definition.

Definition. Two functions f and g are *orthogonal* if $\langle f, g\rangle = 0$.

Definition. A set of functions $\{\phi_k\}_{k=0}^n$ is a system of *orthogonal polynomials* provided:
- ϕ_k is a polynomial of exact degree k (with $\phi_0 \neq 0$);
- $\langle \phi_j, \phi_k\rangle = 0$ when $j \neq k$.

Be sure not to overlook the first property, that ϕ_k has exact degree k; it ensures the following result.

Proposition. The system of orthogonal polynomials $\{\phi_k\}_{k=0}^\ell$ is a basis for $\mathcal{P}_\ell$, for all $\ell = 0, \ldots, n$.

This leads immediately to our first key theorem, one we will use repeatedly.

Theorem. Let $\{\phi_j\}_{j=0}^n$ be a system of orthogonal polynomials. Then $\langle p, \phi_n\rangle = 0$ for any $p \in \mathcal{P}_{n-1}$.

Proof. Our previous proposition implies that $\{\phi_k\}_{k=0}^{n-1}$ is a basis for $\mathcal{P}_{n-1}$. Thus for any $p \in \mathcal{P}_{n-1}$,

$$p = \sum_{k=0}^{n-1} c_k \phi_k$$

for some constants $\{c_k\}_{k=0}^{n-1}$. The linearity of the inner product and orthogonality of $\{\phi_k\}_{k=0}^n$ imply that

$$\langle p, \phi_n\rangle = \Big\langle \sum_{k=0}^{n-1} c_k\phi_k, \phi_n \Big\rangle = \sum_{k=0}^{n-1} c_k\langle \phi_k, \phi_n\rangle = \sum_{k=0}^{n-1} 0 = 0. \qquad \blacksquare$$

We need a mechanism for constructing orthogonal polynomials. The Gram–Schmidt process used to orthogonalize vectors in $\mathbb{C}^n$ can easily be generalized to the present setting. Suppose that we have some $(n+1)$-dimensional subspace $\mathcal{S}$ with the basis $p_0, p_1, \ldots, p_n$. Then the classical Gram–Schmidt algorithm takes the following form.

Gram–Schmidt orthogonalization. Given a basis $\{p_0, \ldots, p_n\}$ for some subspace $\mathcal{S}$, the following algorithm will construct an orthogonal basis $\{\phi_0, \ldots, \phi_n\}$ for $\mathcal{S}$:

$\phi_0 := p_0$

for $k = 1, \ldots, n$

$$\phi_k := p_k - \sum_{j=0}^{k-1} \frac{\langle p_k, \phi_j\rangle}{\langle \phi_j, \phi_j\rangle}\phi_j$$

end.

This is a convenient process, but like the vector Gram–Schmidt process, it requires a nontrivial amount of computation. As k gets larger, the work required in the sum at step k grows: the work grows with every step. (Recall that when dealing with functions on $C[a,b]$, each inner product evaluation requires the computation of an integral, potentially a expensive operation.)

To construct a set of orthogonal polynomials, we take some a basis $\{p_k\}_{k=0}^n$ for $\mathcal{P}_n$, and perform Gram–Schmidt orthogonalization. If p_k has exact degree k for $k = 0, \ldots, n$, then ϕ_k will have exact degree k as well, as required for a system of orthogonal polynomials. The simplest basis for $\mathcal{P}_n$ is the monomial basis, $\{x^k\}_{k=0}^n$. One could perform Gram–Schmidt orthogonalization directly on this basis to obtain orthogonal polynomials, but there is a slicker alternative for which most of the terms in the sum for ϕ_k turn out to be zero.

Suppose one has a set of orthogonal polynomials, $\{\phi_k\}_{k=0}^n$, and seeks the next orthogonal polynomial, ϕ_{n+1}. Since ϕ_n has exact degree n, the polynomial $x\phi_n(x)$ has exact degree $n+1$. Thus, we could apply Gram–Schmidt orthogonalization on $\{\phi_0(x), \phi_1(x), \ldots, \phi_n(x), x\phi_n(x)\}$, which forms a basis for $\mathcal{P}_{n+1}$. This will allow us to make an essential simplification to the customary Gram–Schmidt recurrence

$$\phi_{n+1}(x) = x\phi_n(x) - \sum_{j=0}^{n} \frac{\langle x\phi_n(x), \phi_j(x)\rangle}{\langle \phi_j, \phi_j\rangle}\phi_j(x).$$

First notice that

$$\begin{aligned}\langle x\phi_n(x), \phi_k(x)\rangle &= \int_a^b \Big(x\phi_n(x)\Big)\phi_k(x)w(x)\,\mathrm{d}x \\ &= \int_a^b \phi_n(x)\Big(x\phi_k(x)\Big)w(x)\,\mathrm{d}x \\ &= \langle \phi_n(x), x\phi_k(x)\rangle.\end{aligned}$$

Since $x\phi_k(x) \in \mathcal{P}_{k+1}$,

$$\langle x\phi_n(x), \phi_k(x)\rangle = \langle \phi_n(x), x\phi_k(x)\rangle = 0$$

for all $j < n-1$. This eliminates the bulk of the terms from Gram–Schmidt sum:

$$\sum_{k=0}^{n} \frac{\langle x\phi_n(x), \phi_k(x)\rangle}{\langle \phi_k, \phi_k\rangle}\phi_k = \sum_{k=n-1}^{n} \frac{\langle x\phi_n(x), \phi_k(x)\rangle}{\langle \phi_k, \phi_k\rangle}\phi_k.$$

Thus we can compute orthogonal polynomials efficiently, even if the necessary polynomial degree is large.† This fact has vital implications in numerical linear algebra: indeed, it is a reason that the iterative conjugate gradient method for solving $\mathbf{Ax} = \mathbf{b}$ often executes with blazing speed, but that is a story for another class.

Theorem (Three-Term Recurrence for Orthogonal Polynomials). Given a weight function $w(x)$ ($w(x) \ge 0$ for all $x \in (a,b)$, and $w(x) = 0$ only on a set of measure zero), a real interval $[a,b]$, and an associated real inner product

$$\langle f, g\rangle = \int_a^b w(x) f(x) g(x)\,\mathrm{d}x,$$

then a system of (monic) orthogonal polynomials $\{\phi_k\}_{k=0}^n$ can be generated as follows:

$$\begin{aligned}
\phi_0(x) &= 1,\\
\phi_1(x) &= x - \frac{\langle x, 1\rangle}{\langle 1, 1\rangle},\\
\phi_k(x) &= x\phi_{k-1}(x) - \frac{\langle x\phi_{k-1}(x), \phi_{k-1}(x)\rangle}{\langle \phi_{k-1}(x), \phi_{k-1}(x)\rangle}\phi_{k-1}(x) - \frac{\langle x\phi_{k-1}(x), \phi_{k-2}(x)\rangle}{\langle \phi_{k-2}(x), \phi_{k-2}(x)\rangle}\phi_{k-2}(x) \qquad \text{for } k \ge 2.
\end{aligned}$$

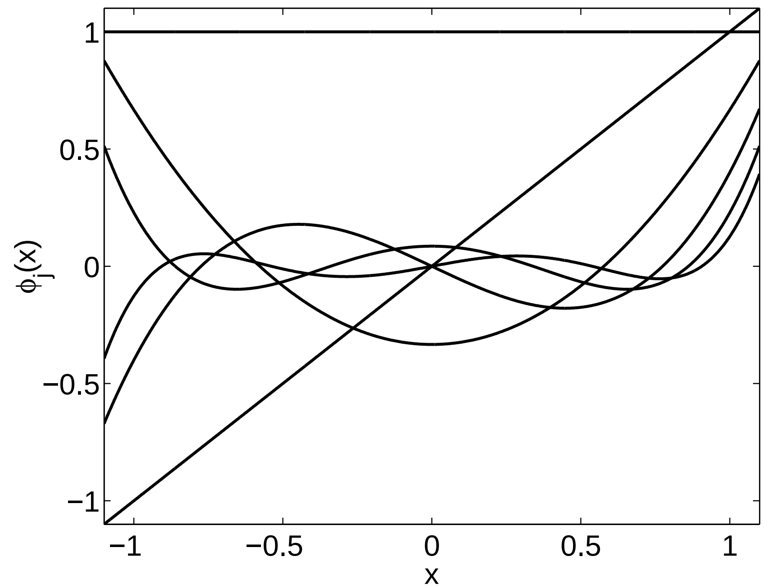
Our definition of orthogonal polynomials made no stipulation about normalization. It is often convenient to work with monic polynomials, i.e., $\phi_k(x) = x^k + \cdots$, as constructed by the three-term recurrence above. Some applications make other normalizations more convenient, e.g., $\langle \phi_k, \phi_k\rangle = 1$ or $\phi(0) = 1$. It is a simple exercise to adapt the three term recurrence to generate such alternative normalizations.

Legendre polynomials. On the interval $[a,b] = [-1,1]$ with weight $w(x) = 1$ for all x, the orthogonal polynomials are known as *Legendre polynomials*:

$$\begin{aligned}
\phi_0(x) &= 1\\
\phi_1(x) &= x\\
\phi_2(x) &= x^2 - \tfrac{1}{3}\\
\phi_3(x) &= x^3 - \tfrac{3}{5}x\\
\phi_4(x) &= x^4 - \tfrac{6}{7}x^2 + \tfrac{3}{35}\\
\phi_5(x) &= x^5 - \tfrac{10}{9}x^3 + \tfrac{5}{21}x\\
\phi_6(x) &= x^6 - \tfrac{15}{11}x^4 + \tfrac{5}{11}x^2 - \tfrac{5}{231}.
\end{aligned}$$

†The Gram–Schmidt process will not reduce to a short recurrence in all settings. We used the key fact $\langle x\phi_n, \phi_k\rangle = \langle \phi_n, x\phi_k\rangle$, *which does not hold in general inner product spaces*, but works perfectly well in our present setting because *our polynomials are real valued on* $[a,b]$. The short recurrence does not hold, for example, if you compute orthogonal polynomials over a general complex domain, instead of the real interval $[a,b]$.

Below we show a plot of $\phi_0, \phi_1, \ldots, \phi_5$. Note how distinct these polynomials are from one another, somewhat reminiscent of the Lagrange basis functions for polynomial interpolation.



Orthogonal polynomials play a key role in a prominent technique for computing integrals known as Gaussian quadrature. In that context, we will see other families of orthogonal polynomials: the Chebyshev, Laguerre, and Hermite polynomials.

3.3.6. Continuous least squares with orthogonal polynomials.

Definition. A system of orthogonal polynomials $\{\psi_k\}_{k=0}^n$ is *orthonormal* provided that $\langle \psi_k, \psi_k \rangle = 1$ for all $k = 0, \ldots, 1$.

Given a any set of orthogonal polynomials $\{\phi_k\}_{k=0}^n$, we obtain orthonormal polynomials by setting

$$\psi_k := \frac{\phi_k}{\langle \phi_k, \phi_k \rangle^{1/2}},$$

giving

$$\langle \psi_k, \psi_k \rangle = \frac{\langle \phi_k, \phi_k \rangle}{\langle \phi_k, \phi_k \rangle} = 1.$$

We seek an expression for the least squares approximation to f as a linear combination of orthonormal polynomials. That is, determine the coefficients $\{c_k\}_{k=0}^n$ in the expansion

$$p(x) = \sum_{k=0}^{n} c_k \psi_k(x)$$

to minimize $\|f - p\|_{L^2}$. The optimal choice of coefficients follows immediately from the linear system derived in the last lecture,

$$\begin{bmatrix} \langle \psi_0, \psi_0 \rangle & \langle \psi_0, \psi_1 \rangle & \cdots & \langle \psi_0, \psi_n \rangle \\ \langle \psi_1, \psi_0 \rangle & \langle \psi_1, \psi_1 \rangle & & \vdots \\ \vdots & & \ddots & \vdots \\ \langle \psi_n, \psi_0 \rangle & \langle \psi_n, \psi_1 \rangle & \cdots & \langle \psi_n, \psi_n \rangle \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} \langle f, \psi_0 \rangle \\ \langle f, \psi_1 \rangle \\ \vdots \\ \langle f, \psi_n \rangle \end{bmatrix}.$$

Since $\{\psi_k\}_{k=0}^n$ is a system of orthonormal polynomials, this matrix equation reduces to

$$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} \langle f, \psi_0 \rangle \\ \langle f, \psi_1 \rangle \\ \vdots \\ \langle f, \psi_n \rangle \end{bmatrix},$$

with the trivial solution $c_k = \langle f, \psi_k \rangle$. As this linear system clearly has a unique solution, the optimal polynomial must be unique.

Theorem. The unique optimal L^2 approximation to $f \in C[a,b]$ on $[a,b]$ is given by

$$p_* = \sum_{k=0}^{n} \langle f, \psi_k \rangle \psi_k,$$

where $\{\psi_k\}_{k=0}^n$ forms a system of orthonormal polynomials on $[a,b]$.

From this expression for the optimal polynomial immediately follows a fundamental property of all least squares approximations.

Theorem (Orthogonality of the optimal L^2 error). Let $p_* \in \mathcal{P}_n$ be the optimal L^2 approximation to $f \in C[a,b]$. Then $f - p_*$ is orthogonal to all $q \in \mathcal{P}_n$, i.e., $\langle f - p_*, q \rangle = 0$.

Proof. Given any $q \in \mathcal{P}_n$, express this polynomial in the basis of orthonormal polynomials,

$$q = \sum_{k=0}^{n} \gamma_k \psi_k.$$

We have just shown that p_* takes the form

$$p_* = \sum_{k=0}^{n} \langle f, \psi_k \rangle \psi_k.$$

Since $\{\psi_k\}_{k=0}^n$ forms a basis for $\mathcal{P}_n$, it suffices to show that $f - p_*$ is orthogonal to each ψ_k. In particular, for $k = 0, \ldots, n$, we have

$$\begin{aligned} \langle f - p_*, \psi_k \rangle &= \langle f - \textstyle\sum_{j=0}^n \langle f, \psi_j \rangle \psi_j, \psi_k \rangle \\ &= \langle f, \psi_k \rangle - \textstyle\sum_{j=0}^n \langle f, \psi_j \rangle \langle \psi_j, \psi_k \rangle \\ &= \langle f, \psi_k \rangle - \langle f, \psi_k \rangle \langle \psi_k, \psi_k \rangle \\ &= \langle f, \psi_k \rangle - \langle f, \psi_k \rangle \\ &= 0. \end{aligned}$$

Since $f - p_*$ is orthogonal to all members of a basis for $\mathcal{P}_n$, it is orthogonal to any member of $\mathcal{P}_n$:

$$\langle f - p_*, q\rangle = \sum_{k=0}^{n} \gamma_k \langle f - p_*, \psi_k\rangle = \sum_{k=0}^{n} 0 = 0. \qquad \blacksquare$$

Example: $f(x) = \mathrm{e}^x$. We repeat our previous example: approximating $f(x) = \mathrm{e}^x$ on $[0,1]$ with a linear polynomial. First, we need to construct orthonormal polynomials for this interval. It is easy to see that $\psi_0(x) = 1$, and a straightforward computation gives $\psi_1(x) = \sqrt{3}(1-2x)$. We then compute

$$\begin{aligned}\langle \mathrm{e}^x, \psi_0(x)\rangle &= \int_0^1 \mathrm{e}^x \,\mathrm{d}x = \mathrm{e} - 1\\ \langle \mathrm{e}^x, \psi_1(x)\rangle &= \sqrt{3}\int_0^1 \mathrm{e}^x(1-2x)\,\mathrm{d}x = \sqrt{3}(\mathrm{e}-3),\end{aligned}$$

giving a formula for p_*:

$$\begin{aligned}p_* &= (\mathrm{e}-1)\psi_0 + \sqrt{3}(\mathrm{e}-3)\psi_1\\ &= (\mathrm{e}-1)1 + \sqrt{3}(\mathrm{e}-3)[\sqrt{3}(1-2x)]\\ &= 4\mathrm{e} - 10 + x(18 - 6\mathrm{e}).\end{aligned}$$

This is exactly the polynomial we obtained using basic calculus techniques.

Note that with this procedure, one can easily to increase the degree of the approximating polynomial. To increase the degree by one, simply add

$$\langle f, \psi_{n+1}\rangle \psi_{n+1}$$

to the old approximation. True, this requires computation of an integral, but the general method we discussed in the last lecture would also require a new integral evaluation to include in the right hand side of the $(n+2)$-by-$(n+2)$ linear system, which then must be solved to get the new approximation.‡ Indeed, an advantage to the new method is that we express the optimal polynomial in a 'good' basis—the basis of orthonormal polynomials—rather than the monic polynomial basis.

```
% Code to demonstrate computation of continuous least squares approximation.
% Uses MATLAB's built-in codes to compute inner products.

% Use the weight function w(x) = 1 on the interval [-1,1].
% Construct the orthogonal polynomials for this weight, interval.
% These are the Legendre polynomials; one can look up their coefficients
% in mathematical tables.  We input them in MATLAB's standard format
% for polynomials.  (We have normalized the standard Legendre polynomials.)

Leg = [[     0     0       0      0      0      0     1]*sqrt(1/2);  % psi_0(x)
       [     0     0       0      0      0      1     0]*sqrt(3/2);  % psi_1(x)
       [     0     0       0      0    3/2      0  -1/2]*sqrt(5/2);  % psi_2(x)
```

‡It is true, however, that both these methods for finding the least squares polynomial will generally be more expensive then simply finding a polynomial interpolant.

```
      [     0     0       0    5/2      0   -3/2     0]*sqrt(7/2);  % psi_3(x)
      [     0     0    35/8      0  -15/4      0   3/8]*sqrt(9/2);  % psi_4(x)
      [     0  63/8       0  -35/4      0   15/8     0]*sqrt(11/2); % psi_5(x)
      [231/16     0 -315/16      0 105/16      0 -5/16]*sqrt(13/2)];% psi_6(x)

% All the necessary integrals have integrands that are the product of our
% target function f(x) = exp(x)*sin(5*x) and some polynomial.
% The following inline function defines this general form of integrand.
 f = inline('sin(pi*x) + 3*exp(-(50*(x-.5)).^2)');
 integrand = inline('feval(f,x).*polyval(p,x)','x','f','p');

% We also include a function to evaluate the 2-norm of the error
 errintegrand = inline('(feval(f,x)-polyval(p,x)).*polyval(q,x)','x','f','p','q');

% compute the expansion coefficients for the optimal polynomial approximation
 x = linspace(-1.1,1.1,1000)';
 figure(1),clf
 plot(x,f(x),'b-','linewidth',3), hold on
 axis([-1.1 1.1 -2 5])
 set(gca,'fontsize',20)
 drawnow
 px = zeros(1,size(Leg,1));

 clear pxplt
 for j=1:size(Leg,1)
     input('press return to continue')
     c(j) = quad(integrand,-1,1,1e-10,[],f,Leg(j,:));
     px = px + c(j)*Leg(j,:);
     fprintf(' c_%d = %10.7f \n', j-1, c(j))
     if exist('pxplt','var'), set(pxplt,'linewidth',1); end
     pxplt = plot(x,polyval(px,x),'r-','linewidth',3);
     quad(errintegrand,-1,1,1e-10,[],f,px,[1])
     title(sprintf('Degree %d Least-Squares Approximation',j),'fontsize',20)
 end
```

Appendix.

We derived the formula

$$p_* = \sum_{k=0}^{n} \langle f, \psi_k \rangle \psi_k$$

based on a simple calculus result from the previous lecture. Here is an alternative derivative-free exposition that mirrors the construction of the discrete least squares solution in §3.1.

For a general $p \in \mathcal{P}_n$, write

$$p = \sum_{k=0}^{n} c_k \psi_k.$$

Using the linearity of the inner product, and the fact that $\{\psi_k\}_{k=0}^{n}$ is a system of orthonormal polynomials, we have

$$\|f - p\|_{L^2}^2 = \langle f - p, f - p \rangle \;=\; \langle f, f \rangle - 2\langle f, p \rangle + \langle p, p \rangle$$

$$\begin{aligned}
&= \|f\|_{L^2}^2 - 2\langle f, \textstyle\sum_{k=0}^n c_k\psi_k\rangle + \langle \textstyle\sum_{k=0}^n c_k\psi_k, \textstyle\sum_{j=0}^n c_j\psi_j\rangle \\
&= \|f\|_{L^2}^2 - 2\textstyle\sum_{k=0}^n c_k\langle f, \psi_k\rangle + \textstyle\sum_{k=0}^n \textstyle\sum_{j=0}^n c_k c_j\langle \psi_k, \psi_j\rangle \\
&= \|f\|_{L^2}^2 - 2\textstyle\sum_{k=0}^n c_k\langle f, \psi_k\rangle + \textstyle\sum_{k=0}^n c_k^2\langle \psi_k, \psi_k\rangle \\
&= \|f\|_{L^2}^2 - 2\textstyle\sum_{k=0}^n c_k\langle f, \psi_k\rangle + \textstyle\sum_{k=0}^n c_k^2.
\end{aligned}$$

Despite these manipulations, it is still not clear how we should choose the c_j to give the least squares approximation. Toward this end, note that

$$(c_k - \langle f, \psi_k\rangle)^2 = c_k^2 - 2c_k\langle f, \psi_k\rangle + \langle f, \psi_k\rangle^2.$$

Rearranging this expression and summing over k, we have

$$-2\textstyle\sum_{k=0}^n c_k\langle f, \psi_k\rangle + \textstyle\sum_{k=0}^n c_k^2 = \textstyle\sum_{k=0}^n \Big[(c_k - \langle f, \psi_k\rangle)^2 - \langle f, \psi_k\rangle^2\Big].$$

Substituting this formula into our expression for the error, we obtain

$$\|f-p\|_{L^2}^2 \;=\; \|f\|_{L^2}^2 + \sum_{k=0}^n (c_k - \langle f, \psi_k\rangle)^2 - \sum_{k=0}^n \langle f, \psi_k\rangle^2.$$

The first term in this key expression, $\|f\|_{L^2}^2$, is independent of our choice of the c_k, as is the last term, $-\sum_{k=0}^n \langle f, \psi_k\rangle^2$. Thus, to minimize $\|f\|_{L^2}^2$, minimize the middle term

$$\sum_{k=0}^n (c_k - \langle f, \psi_k\rangle)^2.$$

As this term is nonnegative, our best hope is to find coefficients c_k that zero out this expression. That is easy:

$$c_k = \langle f, \psi_k\rangle.$$

Moreover, this is the only choice for the c_k that will zero the middle term. Hence, we have constructed the optimal polynomial p_* and shown it to be unique.