(1) (25 points) Suppose we have \(m\) data points \(\{(t_i, y_i)\}_{i=1}^m\), where the \(t\)-values all occur in some interval \([x_0, x_n]\). We subdivide the interval \([x_0, x_n]\) into \(n\) subintervals \(\{[x_k, x_{k+1}]\}_{k=0}^{n-1}\) of equal length “\(h\)” and attempt to choose a spline function \(s(x)\) with nodes at \(\{x_k\}_{k=0}^n\) in such a way so that

\[
\sum_{i=1}^m |y_i - s(t_i)|^2
\]

is minimized.

This can be formulated as a linear least squares problem with a coefficient matrix that has a banded structure:

The key to the structure of the data matrix for spline data fitting is the fact that cubic splines may be represented as linear combinations of “B-splines” which have only local support on the interval:

Note that \(B_{-1}(t)\) and \(B_{n+1}(t)\) are not shown, but they need to be included. So, written in terms of the \(\{B_i(t)\}\), the cubic spline \(s(t)\) appears as

\[
s(t) = \sum_{i=0}^n \alpha_i B_i(t)
\]
with
\[
    h^3 B_i(t) = \begin{cases} 
        (t - x_{i-2})^3 & \text{for } t \in [x_{i-2}, x_{i-1}] \\
        h^3 + 3h^2(t - x_{i-1}) + 3h(t - x_{i-1})^2 - 3(t - x_{i-1})^3 & \text{for } t \in [x_{i-1}, x_i] \\
        h^3 + 3h^2(x_{i+1} - t) + 3h(x_{i+1} - t)^2 - 3(x_{i+1} - t)^3 & \text{for } t \in [x_i, x_{i+1}] \\
        (x_{i+2} - t)^3 & \text{for } t \in [x_{i+1}, x_{i+2}] \\
        0 & \text{for any other } t
    \end{cases}
\]

Note that if \( t_k \in [x_i, x_{i+1}] \) then
\[
    s(t_k) = \sum_{i=0}^{n} c_i B_i(t_k) = c_{i-1} B_{i-1}(t_k) + c_i B_i(t_k) + c_{i+1} B_{i+1}(t_k) + c_{i+2} B_{i+2}(t_k)
\]

The data matrix will never have more than four nonzero entries per row !!

The indexing of the \( B_i(t) \) is different than the indexing given in Prof. Embree’s notes. However, this will give the same results. Also note that \( B_{n+2}(t) \) will not be needed since it vanishes for any \( t \in [x_0, x_n] \).

Acknowledgement: This problem is suggested by Prof. Chris Beattie, VA Tech. Prof.

i. (10 points) Write a program that given an interval \([a, b]\) will construct \( n \) equally spaced grid points \( a = x_0 < x_1 < x_2 < \ldots < x_n = b \) and data points \( \{(t_i, f_i) : i = 1 : m\} \) with \( t_i \in [a, b] \) and with \( f_i = f(t_i) \) for some given function \( f \). The routine should then set up the matrix \( A \) of coefficients and the right hand side \( b \) with \( b(i) = f_i \).

Note: Your \( A \) should be an \( m \times (n + 3) \) matrix with \( m > n + 3 \), where \( m \) is the number of data points and \( n \) is the number of intervals \( [x_j, x_{j+1}] \). It is probably best if you take the points \( t_i \) to be equally spaced in the interval \( (x_0, x_n) \) but with a much smaller spacing and then process these points in increasing order to construct the rows of \( A \) as you need them.

The code should then solve the least squares problem
\[
    \min_c \| b - Ac \|_2.
\]

You may use Matlab’s \( \text{qr} \) and “backslash” for this. The entries of \( c \) will be the coefficients \( \{c_i : i = -1 : n + 1\} \).

The input to the code should be the interval \([a, b]\) (i.e. the values \( a \) and \( b \)), the number of intervals \( n \) (to specify \( [x_{i-1}, x_i], i = 1 : n \)) and a function handle \( f \) for a code to evaluate \( f(t) \) where \( f \) is the function you wish to approximate (see documentation for Matlab’s \( \text{fzero} \) for an example if you have forgotten how to do this).

The output of the code should be the coefficients \( c \) and the vector \( x \) of grid points \( x_i \).

**Soln:** See HW5_prob2.m.

ii. (10 points) Test your routine on data points generated from a variety of functions of a single variable plus “noise” (small random perturbations). Include the following example.
Example: $h_1 = \text{a stepsize smaller than } x(j) - x(j-1)$

$t = [0:h_1:2\pi];$

$y = \sin(t)';$

$m = \text{length}(y);$

$b = y + (.05) * \text{randn}(m,1);$

Turn in the plots for this sine function and also select two additional interesting ones to turn in. Plot the spline fit with the data points and the original (noiseless) function. (legends, title, x and y axis labels please)

You will need to write a routine to evaluate the spline at any argument $t \in [a, b]$ to make these plots.

**Soln:** Setting the noise to be 0.03 results in Figure 0.1.

![Figure 0.1. Plots of the B-spline and the error caused by the introduction of noise.](image)

iii.(5 points) Show a **spy** plot of the matrix $A$ and also for the matrix $A^T A$. Notice the structure. We shall learn later in the course about how to take advantage of this structure (do **help spy** in Matlab to see how to use **spy**).

**Soln:**

![Figure 0.2. Plots of the matrix sparsity patterns for A and A^T A.](image)
Let \( \{ \phi_n(x) \}_{n=0}^{\infty} \) be a family of orthonormal polynomials with respect to the weight function \( w(x) \). Show that the \( n \)th degree polynomial that minimizes
\[
\int_{-1}^{1} w(x) (f(x) - p_n(x))^2 \, dx
\]
is given by
\[
p_n(x) = \sum_{k=0}^{n} c_k \phi_k(x)
\]
where \( c_k = \int_{-1}^{1} w(x) f(x) \phi_k(x) \).

**Hint:** Expand the expression you want to minimize.

**Soln:** We want to minimize \( \| f - p_n \|_{L^2} \). Note \( \| f \|_{L^2} = \langle f, f \rangle = \int_{-1}^{1} w(x) (f(x))^2 \, dx \).

For simplicity of presentation, I choose to use the inner product notation.

\[
\| f - p_n \|_{L^2}^2 = \langle f - p_n, f - p_n \rangle = \sum_{k=0}^{n} c_k \phi_k, f - \sum_{k=0}^{n} c_k \phi_k >
\]
\[
= \| f \|^2 - 2 \sum_{k=0}^{n} c_k \langle f, \phi_k \rangle + \sum_{k=0}^{n} \sum_{l=0}^{n} c_k c_l \langle \phi_k, \phi_l \rangle
\]
\[
= \| f \|^2 - 2 \sum_{k=0}^{n} c_k \langle f, \phi_k \rangle + \sum_{k=0}^{n} c_k^2 \langle \phi_k, \phi_k \rangle \quad \text{since} \quad \langle \phi_k, \phi_l \rangle = 0 \text{ when } l \neq k
\]
\[
= \| f \|^2 - 2 \sum_{k=0}^{n} c_k \langle f, \phi_k \rangle + \sum_{k=0}^{n} c_k^2 \quad \text{since} \quad \langle \phi_k, \phi_k \rangle = 1
\]
\[
= \| f \|^2 + \sum_{k=0}^{n} (-2c_k \langle f, \phi_k \rangle + c_k^2)
\]
\[
= \| f \|^2 - \sum_{k=0}^{n} \langle f, \phi_k \rangle^2 + \sum_{k=0}^{n} \langle f, \phi_k \rangle^2 - 2c_k \langle f, \phi_k \rangle + c_k^2
\]
\[
= \| f \|^2 - \sum_{k=0}^{n} \langle f, \phi_k \rangle^2 + \sum_{k=0}^{n} \langle f, \phi_k \rangle - c_k
\]

Since \( f \) and all the basis functions \( \phi_k \) are prescribed the only thing that we have control over is \( c_k \). Hence the only way to minimize this expression is if \( \langle f, \phi_k \rangle - c_k = 0 \) for all \( k \). Hence \( c_k = \langle f, \phi_k \rangle \) is the minimal choice of coefficient.