HW #6

1. In class we discussed both the clamped and natural spline interpolants through N data points. The clamped spline is determined by prescribing the derivatives at the endpoints, while the natural spline is determined by prescribing zero, or otherwise fixed, curvatures (second derivatives) at the endpoints. For example, the clamped spline interpolant of the data $\{(x_j, y_j)\}_{j=1}^{N}$ is determined by the equations

$$C_1'(x_1) = \alpha, \quad C_{N-1}'(x_N) = \beta, \quad C_i''(x_{i+1}) = C_{i+1}''(x_{i+1}) \quad \text{for } i = 1, \ldots, N-2, \tag{1}$$

where

$$\begin{aligned} C_i(x) &= y_i + y[x_i, x_{i+1}](x - x_i) \\ &+ \frac{y[x_i, x_{i+1}] - s_i}{\Delta x_i}(x - x_i)(x - x_{i+1}) \\ &+ \frac{s_{i+1} + s_i - 2y[x_i, x_{i+1}]}{\Delta x_i^2}(x - x_i)^2(x - x_{i+1}). \end{aligned} \tag{2}$$

The unknowns in the system (1) are the components of the vector $\boldsymbol{s}$, and more explicitly this *tridiagonal* system is $s_1 = \alpha$, $s_N = \beta$, and for $i = 1, \ldots, N-2$

$$\Delta x_{i+1} s_i + 2(\Delta x_i + \Delta x_{i+1}) s_{i+1} + \Delta x_i s_{i+2} = 3\left(\Delta x_{i+1} y[x_i, x_{i+1}] + \Delta x_i y[x_{i+1}, x_{i+2}]\right). \tag{3}$$

Write a MATLAB® `function s=GetSplineSlopes(x,y,alpha,beta,type)` that returns the slopes `s` which determine the clamped/natural spline, based on `type = C/N`, through the data `x,y` with first/second derivative information `alpha,beta` at the left and right endpoints respectively.

2. The functions `pwchermite_coeffs` and `eval_pwpoly` were discussed in class and are available from the web-site. Use these functions and `GetSplineSlopes` (with necessary modifications) to compute and plot the following splines through the data `x=transpose(linspace(0,1,n))`, `y=f(x)` for $f(x) = e^{-2x}\sin(10\pi x)$.

1. The natural spline for $n = 9$ and $n = 17$ (two separate plots). In each plot show the spline interpolant (solid line), the data (stars), and the function $y = f(x)$ (dashed line).

2. Repeat, using the clamped spline with the correct end conditions

$$\alpha = f'(0), \beta = f'(1).$$

3. Repeat, using the clamped spline with the the end conditions

$$\alpha = -1, \beta = -1.$$

3. Derive the equations for the coefficients of the clamped spline given in Problem 1. Credit will only be given for a clear and complete derivation.

4. Suppose you are given a set of data points (x_j, y_j) in the plane that describe a curve that is not a function of x, and we want to interpolate the data. The basic idea is to think of the curve as parametrized by arclength λ: $(x(\lambda), y(\lambda))$. Here we use λ to denote arclength in lieu of the typical s, since s was earlier used for slope (derivative) information. Thus the discrete data correspond to discrete values of arclength λ_j. That is, $x_j = x(\lambda_j)$ and $y_j = y(\lambda_j)$. To determine λ_j we use the arclength of the piecewise linear interpolant. For example, the data $(x_1, y_1) = (1, 1)$ $(x_2, y_2) = (2, 2)$ $(x_3, y_3) = (1, 4)$ is viewed as corresponding to arclength values $\lambda_1 = 0$, $\lambda_2 = \sqrt{2}$, $\lambda_3 = \sqrt{2} + \sqrt{5}$. Once we have determined the λ_j, we can fit a cubic spline through $\{(\lambda_j, x_j)\}_{j=1}^{N}$ and a second cubic spline through $\{(\lambda_j, y_j)\}_{j=1}^{N}$, and then plot the resulting curve. An interesting example of the technique is the need to approximate curves for the automatic control of sewing machines. Consider the following data.

```
x=[4.7 4.0 4.6 2.3 2.3 4.7 4.1 8.3 6.0 8.6 7.8 5.4 5.4 4.7];
y=[1.3 1.9 2.5 3.1 3.9 3.1 5.8 5.8 3.0 2.9 2.0 2.4 1.5 1.3];
```

Fit the data (as a function of arclength) by with two natural cubic splines and then plot $(x(\lambda), y(\lambda))$ as well as the data points. For this you will need the functions `eval_pwpoly` and `pwchermite_coeffs` available on the web-site.

Problem 1. Computer problem 8, Section 4.1, page 200 of Sauer's textbook. (Hint: Hilbert matrix is given by $H_{ij} = 1/(i + j - 1)$ for $i, j = 1, 2, 3, \ldots$)

Problem 2. Find (by hand) a QR factorization of the matrix given in problem 2(b) from Section 4.3, page 224 of Sauer's textbook. Use the Gram-Schmidt procedure described in Sauer's textbook, and compare your answer with the factorization returned by `qr` in MATLAB® .

Problem 3. Use your results from **Problem 2** to solve (by hand) the least squares problem given in problem 7(b) from Section 4.3, page 224 of Sauer's textbook. Also report the length of the minimum residual.