CS 322: Assignment P2

Due: Wednesday, February 20, 2002 (In Lecture)

You may work in pairs. Follow the course rules for the submission of assignments. Do not submit work unless you have adhered to the principles of academic integrity as descibed on the course website. Points will be deducted for poorly commented code, redundant computation that seriously effects efficiency, and failure to use features of MATLAB that are part of the course syllabus.

Part A (8 pts) Local Max's and Min's of a Cubic Spline

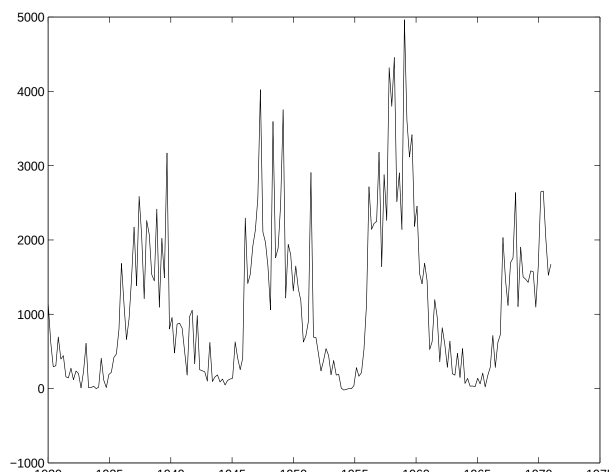
Suppose S is the MATLAB cubic spline interpolant of the data (t_i, y_i), $i = 1{:}n$. Assume $t_1 < \cdots < t_n$ and that $t_1 \leq t_* \leq t_n$. Note that S has a local maximum at t_* if $S'(t_*) = 0$ and $S''(t_*) < 0$. Likewise, S has a local minimum at t_* if $S'(t_*) = 0$ and $S''(t_*) > 0$. Write a MATLAB function `[tmax,tmin] = MaxMin(S)` that returns in the column vectors `tmax` and `tmin` all the points in $[t_1, t_n]$ where S has a local maximum and a local minimum respectively. The values in these vectors should be in ascending order; e.g., `tmax = [1;3;4]`. Return the empty matrix in `tmax` and/or `tmin` if there are no local maxima or minima.

This problem involves examining each local cubic for critical values. Does the ith local cubic have a local max or min in the ith subinterval? Make sure that double copies of a critical point are not returned, something that could easily happen if a critical point coincides with an interior knot of S.

Test your implementation by writing a script `P2A`. It should apply `MaxMin` to the spline interpolant of the $\sin(t)\exp^{-t/10}$ at `t = linspace(0,8*pi,100)`. The spline interpolant should be plotted across $[0, 8\pi]$. Place an '`*`' on the graph of the spline at all the local minima and an '`o`' on the graph of the spline at all the local maxima. Also print out the vectors `tmax` and `tmin` to full machine precision. Submit all the output and a listing of `P2A` and `MaxMin`. For background, be sure to read §3.3.6.

Part B (8 pts) Interpolating Sunspot Data [1]

Download the file `SunSpotArea.txt` from the website and also the function `GetSunSpotData`. Each row in this file houses three numbers: a year, a month index, and a measure of sunspot activity for that year and month. The data covers 1875 through 2001. Make sure `SunSpotArea.txt` is in whatever directory houses the `.m` files associated with this problem. The command `z = GetSunSpotData` establishes `z` as a column vector of length $n = (2001 - 1875 + 1) * 12$ with the property that `z(12*(y-1875)+j)` is the total area of sunspots during month j of year y for all $y = 1875{:}2001$ and $j = 1{:}12$. We would like to deduce the approximate period of the sunspot cycle from this data. If we plot the cubic spline interpolatant of the values in the 1930-1970 range we get something like this:



As you can see, it is pretty hard to infer anything precise about the sunspot period from the raw data. To rectify this we introduce the notion of "smoothing" a vector. To smooth a vector we replace the value in each

[1] Our data is from `http://www.nasa.gov/ssl/pad/solar/greenwich.htm`. Scroll down and click on the `FullSun` link.

component by the average of it and its left and right neighbor, leaving the first and last components alone. Thus,

$$\left[\begin{array}{ccccc} z_1 & (z_1+z_2+z_3)/3 & (z_2+z_3+z_4)/3 & (z_3+z_4+z_5)/3 & z_5 \end{array}\right]$$

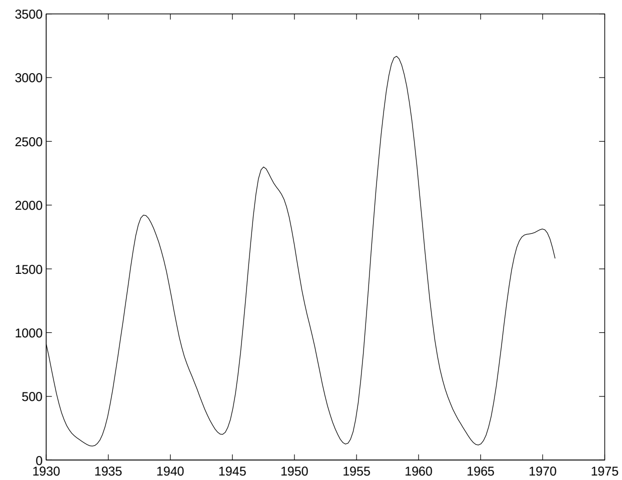
is the smoothed version of $[z_1\ z_2\ z_3\ z_4\ z_5\ ]$. Repeatedly smoothing a vector has the effect of "evening out" oscillations among the components. For example, after two smoothings

$$z = \left[\begin{array}{cccccc} 1.0000 & 4.0000 & 2.0000 & 5.0000 & 3.0000 & 7.0000 \end{array}\right]$$

turns into

$$z = \left[\begin{array}{cccccc} 1.0000 & 2.3333 & 3.1111 & 4.0000 & 5.1111 & 7.0000 \end{array}\right]$$

If we smooth the 1930-1970 sunpot data enough before generating the spline interpolant we get something like this:



Notice that the we're not saying that this interpolates/approximates the original data. (Look at the vertical scaling ranges in the two figures.) However, we now have a chance of discovering the (approximate) period by looking for local maxima and minima.

Write a function `S = SunSpotSpline(Y1,Y2,nSmooth)` that computes the spline interpolant of the sunspot data from year `Y1` through and including year `Y2` which has been smoothed `nSmooth` times. Thus, the spline will be based upon `12(Y2-Y1+1)` data points. Your function should assume that `Y1` $<=$ `Y2` and `nSmooth` $>=$ `0`. Write a script `p2B` that plots the splines defined by

```
Y1 = 1930;
Y2 = 1970;
for nSmooth = [0 10 50 100];
   S = SunSpotSpline(Y1,Y2,nSmooth);
end
```

Make sure that the x-axis is labeled nicely as in the above figures (or better). Indicate the value of `nSmooth` in the title. Submit output which consists of four figures and a listing of `SunSpotSpline` and `p2B`

Part C (4 pts) Estimating the Sunspot Cycle Period

Write a script `P2C` that prints tables indicating the time of the sunspot maximums and minimums during the interval [1900,2000]. Do this by applying `MaxMin` to `S = SunSpotSpline(1900,2000,200)` Print two tables, one for the maximums and one for the minimums. The tables should have two columns. In the first column report the time of the maximum (or minimum). In the second column begining in row 2, print the "time gap" that has passed since the previous maximum (or minimum). Numbers should be displayed through the second decimal place. Submit a listing of `P2C` and output.