CSE 4345 / CSE 5315 - *Computational Methods*

Homework 3- Fall 2011

Due Date: Nov. 6 2011

Problems marked with * are required only for students of CSE 5315 but will be graded for extra credit for students of CSE 4345.

Interpolation

1. Polynomial Interpolation
   For many applications, interpolating complex functions using polynomials is efficient for subsequent calculations. In many cases it is possible to sample particular points of the original function and to use the results to interpolate them using a polynomial to obtain a polynomial representation.

   In this problem you are to perform polynomial interpolation for such a system for an original data function $f(x) = \sin^4(x)$ and an interval of interest between 0 and 2π. For each of parts a) and b), and c) you are to perform sampling (i.e. selecting and calculating a set of data points to interpolate) and interpolation for 3, 9, and 15 data points where the two most extreme data points should be at $x = 0$ and $x = 2\pi$, respectively.

   a) Lagrange interpolation using uniformly spaced data points.
   Perform the interpolation using Lagrange basis polynomials and data points that are sampled uniformly over the interval $[0, 2\pi]$ (with the two outside points being at $x = 0$ and $x = 2\pi$).

   For each of the number of points above (3, 9, 15), list the sampled data points, show the formula of the polynomial, and plot the resulting interpolating polynomial.

   b) Newton interpolation using uniformly spaced data points.
   Perform the interpolation using Newton basis polynomials and data points that are sampled uniformly over the interval $[0, 2\pi]$ (with the two outside points being at $x = 0$ and $x = 2\pi$).

   For each of the number of points above, list the sampled data points, show the formula of the polynomial, and plot the resulting interpolating polynomial.

   c) Newton interpolation using Chebyshev points
   Perform the same interpolation with Newton basis polynomails using Chebyshev data points over the interval $[0, 2\pi]$ (with the two outside points again being at $x = 0$ and $x = 2\pi$).

   For each of the number of points above, again list the sampled data points, show the formula of the polynomial, and plot the resulting interpolating polynomial.

d)* Interpolation error
For both polynomial interpolations using Newton basis polynomials, compute and list the maximum interpolation error for each of the number of data points (3, 9, 15) and data point selection criteria (uniform and Chebyshev).

Compare the maximum interpolation errors for the the two data point selection methods at the different data point numbers and discuss your findings.

Also discuss for each of the data point selection criteria whether (and why) it would be possible to perform the interpolations with Newton basis polynomials incrementally, when moving from 3 to 9, and then to 15 data points. Being able to do this incrementally would allow to build an algorithm that could use the interpolation error to determine whether to further increase the number of data points (and thus the degree of the polynomial) without having to redo the interpolation from scratch).

2. Piecewise Polynomial Interpolation

Piecewise interpolation allows to interpolate a larger number of data points with a set of simpler functions. To ensure that the resulting piecewise polynomial function is smooth, the polynomials have to be selected such as to have continuous differentials.

a) Cubic Hermite Interpolation
Use cubic Hermite Interpolation to interpolation the following data points into a piecewise polynomial function.
$(0, 2), (1, 4), (2, 3), (3, 6), (4, 5), (5, 3)$
List the piecewise interpolation function and plot it,

b) Multi-Dimensional Cubic Hermite Interpolation
To store and represent computer graphics it is often useful to construct shapes by representing them in terms of a set of data points and a piecewise interpolation method so that the shape can be generated uniquely from the set of data points (and can be scaled by simply scaling the data points).

Use cubic Hermite Interpolation to interpolate the following sequence of data points into a 2D curve.
$(0, 0), (0, 1.9), (0, 2), (0.1, 2), (0.65, 2), (0.75, 2), (1.25, 1.5), (0.75, 1), (0.65, 1), (0.1, 1), (0, 1), (0.125, 0.9), (1.25, 0)$
List the piecewise interpolation functions for these data points and plot the resulting curve.

Note: Since these data points do not represent a function you are to interpolate them as a curve. This means that you have to derive a secondary variable (e.g. the distance between the points) as a parameter for your interpolation functions and derive a separate interpolation function for the x and y coordinates.

c)* Data point transformations
For the data points in b), apply a scaling of 2 and a rotation by $\frac{2\pi}{3}$ around the origin $(0, 0)$ to all the data points and repeat the interpolation.

Again, list the piecewise interpolation functions and plot the resulting curve.

Least Squares Approximation

3) Linear Least Squares Solution for CPU Capacitance

Green computing has become an increasingly important topic and the power consumption of the CPU has become a significant consideration for PC makers. In general, the power consumption of the CPU depends linearly on the clock rate and capacitance of the circuits on the chip and quadratically on the supply CPU voltage. This leads to the general formula $P = C * f * V^2$. To minimize power, chip makers can employ three basic mechanisms: i) reduce supply voltage, ii) reduce clock rate, and iii) turn off parts of the chip to reduce the capacitance. For all of them to be successful it is important to determine the capacitance of the chip (or of individual cores on the chip). One way to do this is to experimentally determine it by measuring power consumption for different voltage and clock rate settings. Since such measurements generally contain noise, a least squares solution can solve for the best approximate value of the capacitance.

Determine the best capacitance estimate, C, (in a least squares sense) for the following measurements using Normal equations. Document the intermediate steps to the solution (i.e. the linear system, the Pseudoinverse, and the solution).

Note: For this problem power can be assumed to be the only measured (and thus noisy) value. The frequency and voltage are set and can be assumed to be accurate, allowing the power dissipation equation to be interpreted as a linear equation with one variable (capacitance).

Frequency f	Voltage V	Power P
10^9	2	10
10^9	3	17
$2 * 10^9$	3	45
$2 * 10^9$	4	85
$3 * 10^9$	4	125
$3 * 10^9$	5	170

4) Nonlinear Least Squares Approximation

Consider the following GPS-like indoor localization system for a large arena. In this system, 5 (or more) time-of flight radio senors are mounted in different (known) locations on the ceiling. These sensors are equipped with the ability to measure the point in time at which they receive a signal from a transmitter with a synchronized, high precision clock. Using this information, the system that links theses sensors can now for any transmission (caused, e.g. by the cell-phone of one of the persons in the arena, or by a special purpose transmitter) determine the relative difference in time that the signal required to reach that sensor, and thus the relative difference in distance tot hat sensor. Using this, it is possible to estimate the location in the arena where the transmission originated (and thus to determine where the person is).

Specifically, we can formulate this problem as follows. Once a transmission is received by each of the sensors, we can for each sensor determine what the data receive time difference, Δt_i, between this sensor and a reference sensor (we beforehand select one of the sensors for this) is

by comparing the points in time where the signal was received for the two sensors. Using this time difference and knowing the speed of light, we can from this determine the difference, Δd_i, of the distance of the origin of the transmission from sensor i as opposed to the reference sensor. Setting the distance to the reference sensor to d we then end up with one distance equation for each of the sensors (with one unknown, d). To determine a location, we can also write the distances using the cartesian coordinates of the origin of the transmission and of the sensor. Combining this with the previous equations, this results in one equation for each of the sensors with a total of 4 unknowns (d and the location of the transmission, (x_t, y_t, z_t)).

Mathematically, given sensor locations (x_i, y_i, z_i), and a time stamp on each of the transmission receipts, t_i, we can, making sensor 0 the reference sensor, compute time differences, $Deltat_i = t_i - t_0$. From these (and the speed of light, $c = 299,792,458$) we know that $d_i = d_0 + c\Delta t_i$. We also know that $\sqrt{(x_i = x_t)^2 + (y_i - y_t)^2 + (z_i - z_t)^2} = d_i$, and thus $f_i(d_0, x_t, y_t, z_t) = d_0 + c\Delta t_i - \sqrt{(x_i = x_t)^2 + (y_i - y_t)^2 + (z_i - z_t)^2} = 0$.

Consider the situation where the five sensors are mounted in the following locations in the arena: $\{(0, 50, 25), (0, 0, 40), (0, -50, 25), (35, 0, 30), (-35, 0, 30)\}$.

a) Formulate the complete system of equations for the sensor system with x_t, y_t, z_t, d_0 as variables and t_i as the function values (i.e. derive one equation for each t_i)

b) Implement the Gauss Newton Method to compute the least squares solution (i.e. the best possible guess at the location of the original transmission for the following two sets of time stamps. (you can set up your code specifically for the data set below - and thus derive the corresponding Jacobian beforehand). In your implementation you can use an existing implementation for QR factorization to solve the linear system in each Gauss Newton step.

- First transmission time stamps:

$$t = \begin{pmatrix} 12347.5 \\ 12347.50000001839 \\ 12347.50000011571 \\ 12347.50000001189 \\ 12347.50000005788 \end{pmatrix}$$

- Second transmission timestamps:

$$t = \begin{pmatrix} 13472.4567 \\ 13472.45670007545 \\ 13472.45670017961 \\ 13472.45670016778 \\ 13472.45669998171 \end{pmatrix}$$