CSCC51H

Assignment 2

Jan. 30, 2012 Scarborough Campus **Due:** Feb. 27, 2012

1. Parametric Interpolation:
In computer typography we are faced with the problem of finding an interpolant to points which lie on a curve in the plane, for example the Greek lower case "alpha", (α). Such a shape cannot be represented with y as a fuction of x since $y(x)$ will not generally be single-valued. One approach to resolve this difficulty is to order the data points, $P_0, P_1, ...P_M$ as we traverse the curve (using an approximation to the 'arclength' to parametrise the curve). Let d_i be the (straight-line) distance between P_{i-1} and P_i, $i = 1...M$, and let $t_i = \sum_{j=1}^{i} d_j$, $i = 0...M$. That is, $t_0 = 0, t_1 = d_1, t_2 = d_1 + d_2$, etc. With $P_i \equiv (x_i, y_i)$, consider the two sets of data, (t_i, x_i) and (t_i, y_i), $i = 0...M$. We can interpolate each of these sets independently to generate approximations $R(t)$ and $S(t)$ ($R(t)$ and $S(t)$ are approximations to the respective coordinate functions $x(t)$ and $y(t)$). Then for any $t \in [0, t_M]$, $P(t) \equiv (R(t), S(t))$ is a point in the plane and, as t increases from 0 to t_M, $P(t)$ interpolates the data and, with luck, reproduces the shape of the Greek letter α too.

   (a) Using drafting paper draw an α and measure seven points P_i on your curve (recording the respective measured values d_i, x_i, y_i). Interpolate the parametric data (ie. the $\{t_i, x_i\}$ and the $\{t_i, y_i\}$ $i = 0 \cdots 6$), using the Matlab function *spline* . Plot the results and show how well the shape of this character is represented. For the two extra constraints necessary to define your spline approximations, $R(t), S(t)$; use estimates for the derivative values $x'(0), x'(t_M), y'(0), y'(t_M)$ as discussed below.

   (b) Write a Matlab function That will determine the two cubic Hermite piecewise polynomial approximations (to $x(s)$ and $y(s)$) from tabulated data (t_i, x_i, x'_i) and (t_i, y_i, y'_i), $i = 0...M$ and use this Matlab function (with $M = 6$) to approximate the parametric curve representing your alpha. Note that, to do this, you will have to measure, from your 'hand-drawn' curve representing your alpha, estimated values for dy/dx at the data points. (That is, you will estimate the slope of the tangent to your curve at each point P_i.) These values can then be used to estimate dx/dt and dy/dt at each P_i since t is an approximation to arclength. (ie, at P_i you can estimate dx/dt and dy/dt using your estimate of dy/dx and the observation that $(dt)^2 = (dx)^2 + (dy)^2$.)

   (c) In both cases, using cubic Hermite and cubic spline interpolation, display and comment on the accuracy and smoothness of the approximating curve.

2. Approximating the length of a hand-drawn curve:
   You are to derive and implement (in Matlab) a method to approximate the 'length' of a hand-drawn curve. The method is based on the 'arclength' parametrisation of the coordinate functions, $x(t)$ and $y(t)$ that you worked with and approximated to solve the first problem. Let your approximation to the curve $(x(t), y(t))$, be $P(t) \equiv (R(t), S(t))$, where $R(t)$ and $S(t)$ are piecewise polynomials.

   The method first determines N equally space points on $P(t)$, calculates the straight line distance between each pair of consecutive points, $P(t_{j-1}), P(t_j)\ \ j = 1, 2 \cdots N$ and takes the sum of these distances to be an approximation to the length of the curve.

   Apply this method to estimate the length of the approximation to the hand-drawn α you used to solve the first problem (corresponding to the use of piecewise Hermite interpolation over the seven data points). Report the results for $N = 51, N = 101$ and $N = 201$ and briefly discus which approximation you feel is the more accurate.