16-811: Math Fundamentals for Robotics, Fall 2018

Assignment 2

DUE: Thursday, September 27, 2018

1. Implement a procedure that interpolates $f(x)$ based on a divided difference approach.

   The procedure should take as input the following parameters:

$$x, x_0, \ldots, x_n, f(x_0), \ldots, f(x_n).$$

   The procedure should compute an interpolated value for $f(x)$ based on the given data points.

   Note: The procedure should use all the data points $(x_i, f(x_i))$, $i = 0, \ldots, n$, effectively implementing an interpolating polynomial of degree n (or less, depending on the data).

   (a) Use your procedure to interpolate $(\log_6 x)^{3/2}$ at $x = 2.25$, based on known values of $(\log_6 x)^{3/2}$ at $x = 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0$.

   (b) Now consider the function

$$f(x) = \frac{6}{1 + 25\,x^2},$$

   with input data given at the points

$$x_i = i\,\frac{2}{n} - 1, \qquad i = 0, \ldots, n.$$

   Use your procedure to estimate $f(x)$ at $x = 0.05$, with $n = 2$.
   Use your procedure to estimate $f(x)$ at $x = 0.05$, with $n = 4$.
   Use your procedure to estimate $f(x)$ at $x = 0.05$, with $n = 40$.
   What is the actual value of $f(0.05)$?

   (c) In this part, you are to (numerically) estimate the maximum interpolation error

$$E_n = \max_{-1 \le x \le 1} |f(x) - p_n(x)|.$$

   (You don't need to do anything fancy; simply discretize the interval $[-1, 1]$ finely and compute errors at the resulting discrete points.)
   Estimate E_n for $n = 2, 4, 6, 8, 10, 12, 14, 16, 18, 20$, and 40, for the function $f(x) = 6/(1 + 25\,x^2)$ given above.
   Do the error estimates make sense? Explain your results.

2. Suppose you wish to build an interpolation table with entries of the form $(x, f(x))$ for the function $f(x) = \sin x$ over the interval $[0, 2\pi]$. Please use uniform spacing between points. How fine must the table spacing be in order to ensure 6 decimal digit accuracy, assuming that you will use linear interpolation between adjacent points in the table? How fine must it be if you will use quadratic interpolation? In each case, how many entries do you need in the table?

3. Implement Newton's Method. Consider the following equation:

$$x = \tan x.$$

There are an infinite number of solutions x to this equation. Use Newton's method (and any techniques you need to start Newton in regions of convergence) to find the two solutions on either side of 11. (Said differently: Find one solution x_{low} less than 11 and one solution x_{high} greater than 11 such that the interval $[x_{low}, x_{high}]$ contains no other solutions.)

4. If ξ is a root of $f(x)$ of order 2, then $f(\xi) = 0$, $f'(\xi) = 0$ and $f''(\xi) \neq 0$.

Show that in this case Newton's method no longer converges quadratically. Do so by showing that the method now converges linearly. (You may assume that $f''(x)$ is continuous in a neighborhood of ξ.)

Now suppose that $f'''(x)$ is continuous in a neighborhood of ξ. Show that the iteration

$$x_{n+1} = x_n - 2\,\frac{f(x_n)}{f'(x_n)}$$

does converge quadratically.

5. (a) Implement Müller's method.

(b) Use Müller's method to find all the real and complex roots of the polynomial

$$p(x) = x^3 - 4x^2 + 6x - 4.$$

6. Consider the two univariate polynomials

$$\begin{aligned} p(x) &= x^3 - 9x^2 + 26x - 24 \\ q(x) &= x^2 + 3x - 10 \end{aligned}$$

(a) Using resultants decide whether $p(x)$ and $q(x)$ share a common root.

(b) If the two polynomials share a common root, use the ratio method discussed in class to find that root.

7. Consider the two bivariate polynomials

$$\begin{aligned} p(x,y) &= 2x^2 + 2y^2 - 1 \\ q(x,y) &= x^2 + y^2 + 2xy - x + y \end{aligned}$$

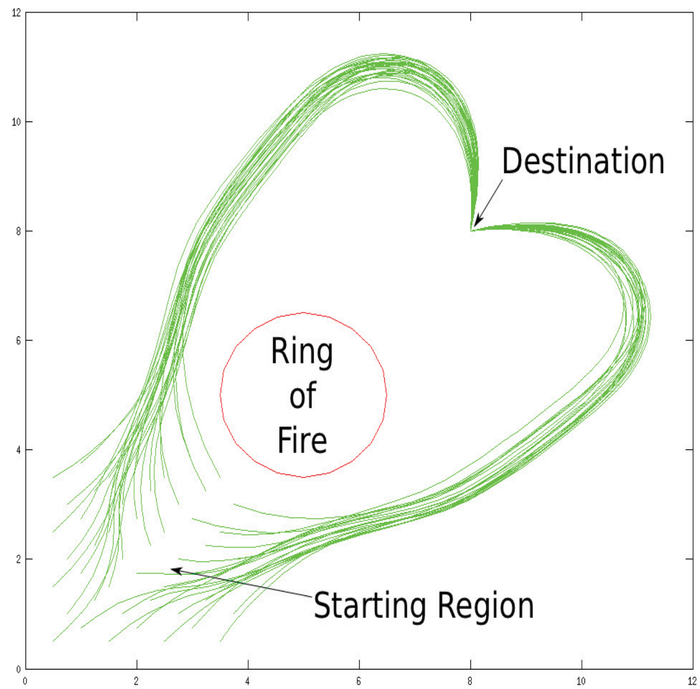
(a) Sketch the two zero contours $p(x,y) = 0$ and $q(x,y) = 0$, for real vales of x and y.

(b) Using resultants, find the intersection points of those two contours (consider only real values of x and y). Do so by eliminating y and constructing a resultant that is a function of x.

(c) Mark the roots of the resultant constructed in part (b) on the x-axis of your sketch from part (a).

8. You are preparing a robotic unicycle to take part in a circus act. For most of the show, a talented acrobat rides the unicycle. But at one point, the acrobat jumps off the unicycle onto a trapeze. After a short trapeze act, the acrobat leaps through the ring of fire in the center of the stage to land on the waiting unicycle, which has moved autonomously to the other side. Your job is to plan a path for the unicycle to take around the ring of fire to the acrobat's landing point.

   You are given a precomputed set of paths which all begin at different points, avoid the circle of fire, and end at the destination (shown below). You must mimic these paths as closely as possible, since they are precisely choreographed for the circus act. However, the acrobat is only human, and does not position the unicycle precisely at any of the paths' starting points. You will need to interpolate a new path from other paths with nearby starting points.



   The destination point is $(8, 8)$, and the ring of fire, a circle of radius 1.5, is centered at $(5, 5)$.

   The precomputed paths are given in the text file `paths.txt`. Every pair of lines in the text file represents a path, which is a sequence of 50 points. The first line contains the x coordinates, and the second contains the y coordinates for one path. The file format is:

$$
\begin{array}{ccccc}
x_1^{(1)} & x_2^{(1)} & \dots & x_{49}^{(1)} & x_{50}^{(1)} \\
y_1^{(1)} & y_2^{(1)} & \dots & y_{49}^{(1)} & y_{50}^{(1)} \\
x_1^{(2)} & x_2^{(2)} & \dots & x_{49}^{(2)} & x_{50}^{(2)} \\
y_1^{(2)} & y_2^{(2)} & \dots & y_{49}^{(2)} & y_{50}^{(2)} \\
\dots & \dots & \dots & \dots & \dots
\end{array}
$$

(a) Write a system of linear equations (in the form $Av = b$, with v representing variables of some sort, appropriately chosen) and constraints (for instance, $v_1 \geq 0$) that will help you determine whether a 2D point (x, y) falls within the triangle formed by three 2D points $(x^{(i)}, y^{(i)})$, $(x^{(j)}, y^{(j)})$, and $(x^{(k)}, y^{(k)})$. (Part of the problem is to think about how you might do this.)

(b) Implement an algorithm to interpolate three paths. Assume the algorithm is given the unicycle's starting location at time $t = 0$.

Your algorithm should first pick three paths $p^{(i)}$, $p^{(j)}$, and $p^{(k)}$ (from the file `paths.txt`) to interpolate.

Then your algorithm should construct a new path p as a weighted sum of these three paths. So, at each time t, $p(t) = \alpha_i\, p^{(i)}(t) + \alpha_j\, p^{(j)}(t) + \alpha_k\, p^{(k)}(t)$, with $p(0)$ the given starting location of the unicycle and with the weights α_i, α_j, α_k fixed throughout.

[Notation: For a path p, $p(t)$ is the 2D point $(x(t), y(t))$ that describes the location of the path at time t.

Your algorithm should be able to produce a value $p(t)$ for all relevant times t prior to reaching the Destination, not just for the discrete time snapshots given in `paths.txt`. These points $p(t)$ should avoid the ring of fire.]

Hint: The unicycle's starting position should fall within the triangle formed by the three paths' starting points, but there may be many valid sets of such starting points. So you should develop criteria to choose one set.

Discuss any decisions you made in your implementation. How did you pick a set of paths? How did you choose the weights $\alpha_i, \alpha_j, \alpha_k$? How did you decide on a time scale for t?

Specificity: You only need to write code to solve the particular problem (with the particular destination, ring of fire, and precomputed paths) described here, not a general purpose algorithm.

(c) Interpolate paths for the starting points (0.8, 1.8), (2.2, 1.0) and (2.7, 1.4). For each starting point, on the same graph, plot the ring of fire, the three paths being interpolated, and the interpolated path. Your paths should not intersect the ring of fire.

(d) Discuss how your algorithm would need to be modified (if at all) if more obstacles were to be introduced.