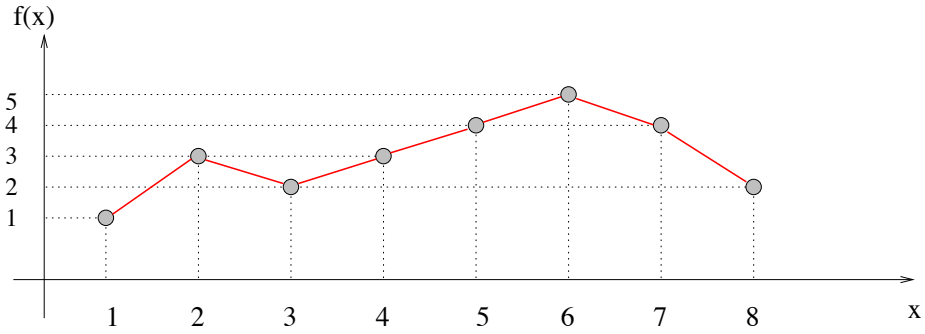
University of Colorado
Department of Computer Science

Numerical Computation CSCI 3656

Spring 2016

Exam 2 Solutions

1. (a) *[4 pts]* Draw a linear interpolation of the points below:



(b) *[4 pts]* Use linear interpolation to estimate the value of $f(x)$ at $x = 2.5$.

$f(2.5) = 2.5$

(c) *[5 pts]* What condition(s) would the points in such a data set have to satisfy for the first derivative of such an interpolation to be continuous?

They would have to be co-linear (all in one line).

2. *[8 pts]* Perform two iterations of the Jacobi method on this system starting from a guess of $[x_1, x_2]^T = [0, 0]^T$:

$$\begin{aligned} -2x_1 + 7x_2 &= 5 \\ 6x_1 - 2x_2 &= 11 \end{aligned}$$

First, rearrange the equations to get the term with the largest magnitude coefficient on the diagonal:

$$\begin{aligned} 6x_1 - 2x_2 &= 11 \\ -2x_1 + 7x_2 &= 5 \end{aligned}$$

Then solve the first equation for x_1 and the second one for x_2:

$$\begin{aligned} x_1 &= \frac{11 + 2x_2}{6} \\ x_2 &= \frac{5 + 2x_2}{7} \end{aligned}$$

Those two equations are what you use to iterate to obtain the $n+1^{st}$ guess from the n^{th}:

$$x_1^{(n+1)} = \frac{11 + 2x_2^{(n)}}{6}$$
$$x_2^{(n+1)} = \frac{5 + 2x_2^{(n)}}{7}$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^{(0)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$
$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^{(1)} = \begin{bmatrix} 11/6 \\ 5/7 \end{bmatrix}$$
$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^{(2)} = \begin{bmatrix} 87/42 \\ 26/21 \end{bmatrix}$$

3. *[5 pts]* What condition(s) must any matrix A satisfy if the Jacobi method is to (eventually) converge to a solution of $A\vec{x} = \vec{b}$?

   It must be *diagonally dominant*.

4. *[9 pts]* In PS7, you wrote a program to compute and draw a cubic spline curve through n points. Discuss how you would parallelize that computation.

   This solution involves a single big matrix solve that has to be done all at once. (Remember: changes made to one point will propagate through *every part* of a cubic natural spline.) That means that you can't pull this problem apart into individual patchwise pieces, one for each set of four points, as you could with Bezier curves or B-splines. However, there's still lots of opportunity for parallelism—in the same ways that you can think about parallelizing *any* $A\vec{x} = \vec{b}$ problem (e.g., allocating one processing unit to each row and doing each column's worth of the forward elimination at once, or going further and having one processing unit for each element in the matrix).

5. *[6 pts]* Compute the Jacobian J of this system:

$$\vec{f}(\vec{x}) = \begin{bmatrix} x_1 x_2 + 2 \\ \sin 2x_1 + x_2^2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$J = \begin{bmatrix} x_2 & x_1 \\ 2\cos 2x_1 & 2x_2 \end{bmatrix}$$

6. *[8 pts]* Explain how you would carry out one step of Newton's method on the system in the previous problem, starting from some guess $\vec{x}_0$. Hint: use that Jacobian J to find the correction $\Delta\vec{x}$. *There is no need to do the math; just explain, in words and symbols, how you would go about it.*

   You would plug in the initial values of x_1 and x_2 to figure out the value of J and $\vec{f}(\vec{x})$ at $x = x_0$, then solve the equation $J\Delta\vec{(x)} = -\vec{f}(\vec{x_0})$ to get the correction $\Delta\vec{(x)}$, and finally add that correction to $\vec{x_0}$ to get the next guess.

   For example, if the guess were $\vec{x_0} = [0.5, 1]^T$, then

$$J = \begin{bmatrix} 1 & 0.5 \\ 2 & 2 \end{bmatrix}$$

   and $\vec{f}(\vec{x_0}) = [2.5, \sin(1) + 1]^T$, so we'd solve

$$\begin{bmatrix} 1 & 0.5 \\ 2 & 2 \end{bmatrix} \begin{bmatrix} \Delta x_0 \\ \Delta x_1 \end{bmatrix} = - \begin{bmatrix} 2.5 \\ \sin(1) + 1 \end{bmatrix}$$

   Then we have

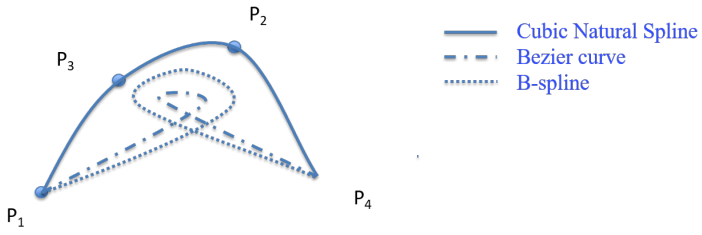
$$\Delta\vec{(x)} = \begin{bmatrix} -4.0793 \\ 3.1585 \end{bmatrix}$$

   Adding the correction to $\vec{x_0}$,

$$\vec{x_1} = \begin{bmatrix} -3.5793 \\ 4.1585 \end{bmatrix}$$

   (We weren't looking for you to include an example; this is just by way of explanation/demonstration.)

7. *[9 pts]* Draw cubic natural splines, cubic Bezier curves & cubic B-spline interpolations of the points below. Label the curves clearly.



8. (a) *[6 pts]* Derive a cubic interpolating polynomial $P_3(x)$ from the data points below using Newton's divided difference method. Here's the formula:

$$P_3(x) = f(x_0) + (x - x_0)f[x_0\, x_1] + (x - x_0)(x - x_1)f[x_0\, x_1\, x_2] + (x - x_0)(x - x_1)(x - x_2)f[x_0\, x_1\, x_2\, x_3]$$

x	$f(x)$
1	-9
2	-1
3	17
4	54

The divided differences triangle is:

-9			
	8		
-1		5	
	18		1.5
17		9.5	
	37		
54			

Using this table, the polynomial is generated as:
$P_3(x) = -9 + (x-1) * 8 + (x-1)(x-2) * 5 + (x-1)(x-2)(x-3) * 1.5$
$= 1.5x^3 - 4x^2 + 9.5x - 16$

(b) *[4 pts]* Use that $P_3(x)$ to estimate $f(5)$.

$P_3(5) = (1.5 * 5^3) - (4 * 5^2) + (9.5 * 5) - 16 = 119$

(c) *[5 pts]* Judging by your results in part (a) of this problem, is $f(x)$ a cubic?

If the $f(x)$ is a polynomial of degree n, the n^{th} column of the divided difference table to the right of the $f(x)$ column will have all the same numbers in it and the $n+1^{st}$ column will have all zeroes in it. I messed up and didn't include enough data points in this table for you to compute enough entries in the third & fourth columns to know whether this $f(x)$ was a cubic, so we accepted all answers that indicated that you knew that the zeroes in a column meant something.

9. *[5 pts]* Name another method for deriving the same $P_n(x)$ from the same data.

   Lagrangian interpolating polynomial (or the "plug'n'chug" method, where you plug each point into the $P_3(x)$ and solve the resulting matrix problem to get the unknown coefficients).

10. *[7 pts]* Are higher-order $P_n(x)$s always better? Explain.

    If the $f(x)$ that's under the hood is **not** a polynomial—and especially if it has flat spots—then using more points exposes you to the Runge effect: the $P_n(x)$ will have to oscillate wildly to go through all the points. (If the $f(x)$ **is** a polynomial, then the coefficients of the higher-order terms in the $P_n(x)$ will automatically zero out, so extra points are not an issue in that situation.)

11. *[4 pts]* What's the difference between extrapolation and interpolation?

In extrapolation, you're estimating the value of $f(x)$ *outside* the range of the points that you used to build the fit; in interpolation, you're estimating its value *inside* that range (i.e., between two of the points that you have).

12. *[4 pts]* List a *geometric* difference between Bezier curves and cubic natural splines.

    There are lots of possible answers here. Bezier curves stay within the convex hull, while cubic natural splines do not. Cubic naturals have to hit all the points, while Beziers only have to hit the patch boundaries. Etc.

13. *[7 pts]* Find the forward and backward errors for this system for the approximate solution $[x_1, x_2]^T = [0, -2]^T$.

$$\begin{bmatrix} 1 & -2 \\ 3 & -4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 7 \end{bmatrix}$$

    The true solution is $[x_1, x_2]^T = [1, -1]^T$.

    The backward error, $||b - Ax_a||_\infty$, is $||[3, 7]^T - [4, 8]^T||_\infty = 1$
    The forward error, $||x - x_a||_\infty$, is $||[1, -1]^T - [0, -2]^T||_\infty = 1$