CS 4210: Midterm Solution Guide

October 23, 2014

(Name)

(Cornell NetID)

Problem 1	25 points	18.8
Problem 2	10 points	5.9
Problem 3	15 points	14.8
Problem 4	25 points	17.6
Problem 5	25 points	19.1
		76.2

Guidelines: $85 \leq A \leq 100$, $65 \leq B \leq 80$, $45 \leq C \leq 60$

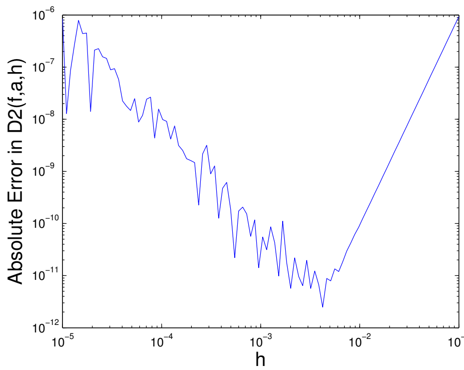
1. The second derivative rule

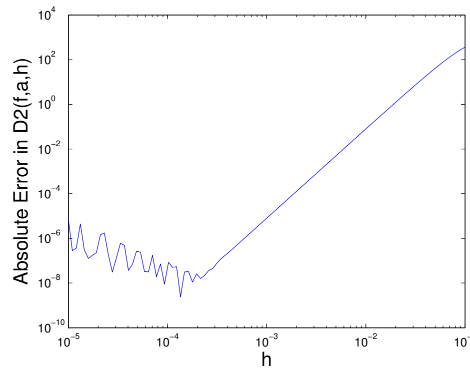
$$D_2(f,a,h) \;=\; \frac{-f(a-2h)+16f(a-h)-30f(a)+16f(a+h)-f(a+2h)}{12h^2}$$

has the property that

$$f^{(2)}(a) \;=\; D_2(f,a,h) \;+\; cf^{(6)}(\eta)h^4 \qquad\qquad a-2h \le \eta \le a+2h$$

where c is a constant. For two separate examples (each with a known second derivative) we plot the value of $|f^{(2)}(a) - D_2(f,a,h)|$ for various values of h and obtain the following:





The total error that is displayed in these plots is a combination of roundoff error and "math error". Model the total error as a function of h and use it to explain the shape of the graphs and what determines the best value of h. Notice that the optimum h values and the minimum error values are rather different in the two examples. Offer a possible explanation. Be brief! No proofs necessary.

You are to "model the total error as a function of h" so anything of the form

$$(\text{something})*h^4 + \text{something}/h^2$$

is fine for 15 points, e.g.

$$err(h) = M_6 h^4 + \frac{\epsilon}{h^2}$$

where M_6 captures some aspect of the size of $f^{(6)}$. Up to eight points for talking qualitatively about the model. Have to show that math error goes down with h and rounding error goes up as h decreases.

5 points for differentiating err and setting the result to zero. Will get something like $h_{opt} \approx (\epsilon/M_6)^{1/6}$.

5 points for explaining why h_{opt} is different: best answer, the sixth derivative of the function in the second example is much bigger.

2. Assume that if the following script is run on a computer that implements 64-bit IEEE floating point arithmetic, then it assigns the value of 53 to `p`

```
x = 1;
p = 0;
y = 1;
z = x +y ;
while z > x
   y = y/2;
   p = p + 1;
   z = x + y;
end
```

(a) What value is assigned to `p` if the first statement is changed to `x = 128`? Briefly justify your answer.

Note that in the given script

$$z = 1 + 1/2^p = (1 + 1/2^p) \times 2^0$$

In the modified script

$$z = 128 + 1/2^p = (1 + 1/2^{p+7}) \times 2^7$$

If the original script stops when $p = 53$, then the modified one stops when $(p + 7) = 53$, i.e., when $p = 46$. 5 points for this. Off-by-one answers OK too.

(b) Does the loop terminate if the first statement is changed to `x = 0`? Why?

$z = 1/2^p = 1.00 \times 2^-p$ Eventually there just are not enough bits in the exponent to represent p. 5 points for this. When this happens the result is set to zero. The loop terminates. It has nothing to do with the machine precision. Many thought if $z < eps = 2^{-53}$ then it is set to zero. Not true. This question is all about the exponent and has nothing to do with the mantissa.

Check out the lecture script `FpFacts`. In parts (a) and (b) minor points if you show some understanding of the finiteness of floating point representation, i.e., (mantissa part)*(power of 2) with limited hardare for each part.

3. Define the piecewise quadratic function $Q(x)$ by

$$Q(x) = \begin{cases} q_1(x) = s_1 + (y_2 - y_1)(x - \frac{1}{2}) + 2(y_1 - 2s_1 + y_2)(x - \frac{1}{2})^2 & 0 \leq x \leq 1 \\ q_2(x) = s_2 + (y_3 - y_2)(x - \frac{3}{2}) + 2(y_2 - 2s_2 + y_3)(x - \frac{3}{2})^2 & 1 \leq x \leq 2 \end{cases}$$

It can be shown that $Q(0) = y_1$, $Q(1) = y_2$, and $Q(2) = y_3$ no matter how we choose the parameters s_1 and s_2. Thus, $Q(x)$ is a continuous interpolant of the points $(0, y_1)$, $(1, y_2)$ and $(2, y_3)$. Show that it is possible to choose s_1 and s_2 so that $Q'(x)$ is also continuous.

8 points for identifying this as the key equation: $q_1'(1) = q_2'(1)$

7 points for spelling it out

$$(y_2 - y_1) + 4(y_1 - 2s_1 + y_2)(1 - \frac{1}{2}) = (y_3 - y_2) + 4(y_2 - 2s_2 + y_3)(1 - \frac{3}{2})$$

i.e.,

$$(y_2 - y_1) + 2(y_1 - 2s_1 + y_2) = (y_3 - y_2) - 2(y_2 - 2s_2 + y_3)$$

i.e.,

$$s_1 + s_2 = \frac{y_1 + 6y_2 + y_3}{4}$$

4. Define the cubic spline

$$B_*(z) \;=\; \begin{cases} 0 & z \le -2 \\ (2+z)^3/6 & -2 \le z \le -1 \\ (-3(1+z)^3 + 3(1+z)^2 + 3(1+z) + 1)/6 & -1 \le z \le 0 \\ (-3(1-z)^3 + 3(1-z)^2 + 3(1-z) + 1)/6 & 0 \le z \le 1 \\ (2-z)^3/6 & 1 \le z \le 2 \\ 0 & 2 \le z \end{cases}$$

and note that

$$B_*'''(z) \;=\; \begin{cases} 0 & z < -2 \\ 1 & -2 < z < -1 \\ -18 & -1 < z < 0 \\ 18 & 0 < z < 1 \\ -1 & 1 < z < 2 \\ 0 & 2 < z \end{cases}$$

Assume that $x_k = a + (k-1)h$ where a and $h > 0$ are given and define

$$B_k(z) \;=\; B_*\left(\frac{z - x_k}{h}\right)$$

for $k = 0, 1, \ldots, n+1$. If

$$s(z) \;=\; \sum_{k=0}^{n+1} \alpha_k B_k(z)$$

and $x_k < z_0 < x_{k+1}$, then what is the value of $s'''(z_0)$? You may assume that $\alpha(0{:}n+1)$ is given and that k (also given) satisfies $1 \le k \le n-1$.

From the local support properties of the basis functions

$$\begin{aligned} s'''(z_0) &= \sum_{j=0}^{n+1} \alpha_j B_j'''(z_0) \\ &= \alpha_{k-1} B_{k-1}'''(z_0) + \alpha_k B_k'''(z_0) + \alpha_{k+1} B_{k+1}'''(z_0) + \alpha_{k+2} B_{k+2}'''(z_0) \end{aligned}$$

8 points for recognizing that.

Using the definition of the basis functions and chain rule:

$$s'''(z_0) = \frac{1}{h^3}\left(\alpha_{k-1} B_*'''\left(\frac{z_0 - x_{k-1}}{h}\right) + \alpha_k B_*'''\left(\frac{z_0 - x_k}{h}\right) + \alpha_{k+1} B_*'''\left(\frac{z_0 - x_{k+1}}{h}\right) + \alpha_{k+2} B_*'''\left(\frac{z_0 - x_{k+2}}{h}\right)\right)$$

8 points for that.

Using the given 3rd derivative facts about B_*:

$$B_*'''\left(\frac{z_0 - x_{k-1}}{h}\right) = -1 \qquad B_*'''\left(\frac{z_0 - x_k}{h}\right) = 18 \qquad B_*'''\left(\frac{z_0 - x_{k+1}}{h}\right) = -18 \qquad B_*'''\left(\frac{z_0 - x_{k+2}}{h}\right) = 1$$

9 points for that. Thus, $s'(z_0) = (-\alpha_{k-1} + 18\alpha_k - 18\alpha_{k+1} + \alpha_k)/h^3$.
Minus 5 for $s'(z_0) = (\alpha_{k-1} - 18\alpha_k + 18\alpha_{k+1} - \alpha_k)/h^3$.

5. Recall that Simpson's rule is the 3-point Newton-Cotes rule. Let $S(a,b,n)$ be the composite Simpson rule with n equal subintervals applied to

$$I(a,b) = \int_a^b f(x)dx.$$

It can be shown that

$$|I(a,b) \; - \; S(a,b,n)| \; \le \; \frac{M_4}{2880\,n^4}(b-a)^5$$

where M_4 is an upper bound on the size of $|f^{(4)}(x)|$.

(a) If the function evaluations associated $S(a,b,2)$ are available, how many new function evaluations are required to compute $S(a,m,2)$ where $m = (a+b)/2$? (Drawing a picture is fine.) Why is this fact important when implementing an adaptive Simpson procedure?

```
a                        m                          b
|------------------------|--------------------------|
x          x             x           x              x      <---- f-eval locations for S(a,b,2)

|------------------------|

x    *     x     *       x       <---- f-eval locations for S(a,m,2)   "*" are the new ones
```

5 points for that. Minus 5 if you replace S(a,b,2) and S(a,m,2) with S(a,b,1) and S(a,m,1) resp.

The re-usable f-evals should be passed through the recursive call to the left half-sized problem. WIll approximately halve the total number of f-evals. 5 points for that.

(b) Suppose the function $f(x)$ satisfies $f(-x) = f(x)$ for all x. Outline how the composite Simpson rule can be used to approximate $I(-a,2a)$ so that the absolute error is bounded by a given positive tolerance δ. Be sure to explain your strategy for minimizing the total number of f-evaluations. (It does not have to be perfect.)

$$I(-a,2a) \;=\; 2I(0,a) + I(a,2a) \;\approx\; 2S(0,a,n_1) + S(a,2a,n_2)$$

5 points for that

Error requirement:

$$2\frac{M_4}{2880\,n_1^4}a^5 \;+\; \frac{M_4}{2880\,n_2^4}a^5 \;\le\; \delta$$

i.e.,

$$\frac{2}{n_1^4} + \frac{1}{n_2^4} \le \frac{\delta}{2880 M_4 a^5} \;\equiv\; \tilde{\delta} \tag{1}$$

5 points for that.

Choosing n_1 and n_2. Simple idea. Choose n_1 so

$$\frac{2}{n_1^4} \le \frac{1}{2}\tilde{\delta}$$

and hoose n_1 so

$$\frac{1}{n_2^4} \le \frac{1}{2}\tilde{\delta}$$

for then (1) is satisfied. 5 points for any idea that works and addresses the idea that we want to minimize $n_1 + n_2$ since the total number of f-evals is about $2n_1 + 1 + 2n_2 + 1$.