CS 4210: Final Exam Solution Guide

December 12, 2014

Problem 1	20 points	12.7
Problem 2	15 points	10.8
Problem 3	20 points	13.1
Problem 4	15 points	10.9
Problem 5	10 points	7.3
Problem 6	20 points	18.3

73.1

Total score distribution based on 50% for assignments (factorings best 6 of 7), 20% for midterm, and 30% for final:

```
90-95 : xxxxx
85-89 : xxxxxxxx
80-84 : xxxxxx
75-79 : xxxxxxxx
70-74 : xx
65-69 : xxxxx
50-64 : xxx
```

1.(a) Consider the initial value problem (IVP) $\dot{y} = f(y, t)$, $y(t_0) = y_0$. The local truncation error (LTE) of the AB2 method

$$y_{k+1} = y_k + \frac{h}{2}(3f_k - f_{k-1})$$

is $O(h^3)$. What does this mean?

It means that if $y_k = y(t_k)$ and $y_{k-1} = y(t_{k-1})$ [2 points] then

$$|y(t_{k+1}) - y_{k+1}| = O(h^3) \qquad \text{[3 points]}$$

(b) The LTE of the AB4 method

$$y_{k+1} = y_k + \frac{h}{24}\left(55f(y_k, t_k) - 59f(y_{k-1}, t_{k-1}) + 37f(y_{k-2}, t_{k-2}) - 9f(y_{k-3}, t_{k-3})\right)$$

is $O(h^5)$. To achieve a given level of accuracy, does it follow that one can take much longer time steps with the AB4 method compared to the AB2 method?

No. The LTE for AB2 involves $y^{(3)}(t)$ while that for AB4 involves $y^{(5)}(t)$. The latter derivative may be much more ill-behaved nullifying the h^5-to-h^3 advantage.

[5 points] You have to talk about higher derivatives

(c) What are the implications if at some point during the AB4 solution process it is decided that the step length h must be halved?

The AB4 method requires an approximate solution at the current time t_c and also at $t_c - h$, $t_c - 2h$, and $t_c - 3h$. If step size is halved then it is necessary to generate an approximations to both $y(t_c - h/2)$ and $y(t_c - 3h/2)$. The corresponding f-evaluations are also required. [5 points]

The problem is not about what happens in future time steps or reasons why it was necessary to halve h. That is why points off if you just say "smaller steps means more f-evals but a reduction in error". No one said we live with the reduced h forever. But some points were given for intelligent chat.

(d) Some IVP solvers work better on stiff problems than others. Explain by briefly discussing constraints on the selection of h for the Euler and backwards Euler methods when they are applied to the problem the $\dot{y} = -100y$. FYI,

$$\text{Euler:} \quad y_{k+1} = y_k + hf(y_k, t_k)$$

$$\text{Backwards Euler:} \quad y_{k+1} = y_k + hf(y_{k+1}, t_{k+1})$$

Euler: $y_{k+1} = y_k - 100hy_k = (1 - 100h)y_k$ To capture the required decay, must have $h < 1/100$.

Backwards Euler: $y_{k+1} = y_k - 100hy_{k+1} = y_k/(1+100h)$. Regardless of h, we get the required decay. 2 points for writing down the Euler and backwards Euler recipes and 3 points for what they say about h selection.

2. Consider the boundary value problem

$$\frac{d^2}{dx^2}u(x) + q(x)u(x) = r(x), \ u(a) = u_a, \ u(b) = u_b$$

and assume that $\{B_0(x), \ldots, B_{n+1}(x)\}$ is a B-spline basis with breakpoints at $a + (k-1)h$, $h = (b-a)/(n-1)$, $k = 0{:}n+1$. In the method of collocation we seek an approximate solution of the form

$$\tilde{u}(x) = \sum_{j=0}^{n+1} \alpha_j B_j(x)$$

Explain how this leads to a linear system of equations with the property that each equation involves only a few of the unknowns $\alpha_0, \ldots, \alpha_{n+1}$.

At a collocation point x_c, we have the linear equation

$$\sum_{=0}^{n+1} \left(B_j''(x_c) + q(x_c)B_j(x_c) \right) \alpha_j = r(x_c)$$

At most 4 of the $B_j(x_c)$ values are nonzero because of local support. Same comment for boundary value equations. (Actually just 3 unknowns per equation if the collaction points are breakpoints.)

10 points for an equation that clearly show that the underlying equations are linear in the α's. You can't just say "substitute $\tilde{u}$ into the BVP

5 points for connecting the local support of the B_j to the sparsity of the system.

-3 if you have some h^2's in your equations. They only show up when you relate B_j'' to B_*'' so if your answer was not in terms of B_*, this was another chain rule misunderstanding.

3(a) Is it possible to interpolate the data $(x_1, y_1), \ldots, (x_n, y_n)$ with a piecewise degree-5 polynomial $q(x)$ with the property that $q^{(k)}(x)$ is continuous for $k = 0, 1, 2, 3, 4$? Hint. How many "degrees of freedom" are there and how many constraints?

Each local quintic has 6 coefficients so there are $6(n-1)$ degrees of freedom. [5 points for this]

At $x_2, \ldots, x_{n-1}$ we require continuity in $q^{(k)}(x)$ for $k = 0, 1, 2, 3, 4$. Thus, there are $5(n-2)$ constraints arising from continuity requirements and n interpolation requirements. Thus, $6n - 10$ constraints. [5 points for this]

2-3 points if you act as if there is a single quintic.

Note, you can't look at a single local quintic and conclude "no" because $6 < 5 + 5$.

(b) If the polynomial $p_{n-1}(x)$ has degree $(n-1)$ and interpolates a function $f(x)$ at $a = x_1 < x_2 < \cdots < x_n = b$, then

$$f(x) = p_{n-1}(x) + \frac{f^{(n)}(\eta)}{n!}(x - x_1) \cdots (x - x_n) \qquad a \leq \eta \leq b$$

If n is large and the x_i are equally-spaced, then the error $|f(x) - p_{n-1}(x)|$ can be large particularly near the endpoints of $[a, b]$. (This is the "Runge phenomena".) Is there a better way to locate the interpolation points? Is there a better way to interpolate with polynomials that does not involve high-order derivatives in the error bound?

Cluster the x_i towards the endpoints. (The Chebychev distribution) [5 points]
Piecewise polynomial interpolation. [5 points]

4(a) Suppose we have a composite quadrature rule Q_n that approximates

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

based on equally spaced subintervals of length $h = (b-a)/n$. Assume that

$$I = Q_n + ch^p$$

and

$$I = Q_{2n} + c\left(\frac{h}{2}\right)^p$$

Express $|I - Q_{2n}|$ as a multiple of $|Q_n - Q_{2n}|$. The multiple should not involve c.

$$0 = Q_n - Q_{2n} + (1 - 1/2^p)ch^p$$

$$I - Q_{2n} = ch^p/2^p = (Q_{2n} - Q_n)(1/(2^p - 1))$$

[10 points]

(b) Adaptive quadrature procedures are recursive and effectively make use of the idea in part (a) even though the parameter c is NOT a constant. Explain

In practice c will depend on some higher derivative of f. If the interval $[a, b]$ is small enough, it is reasonable to assume that that derivative is constant. And with repeated subdivisions, this will be the case in trouble zones. [5 points]

5. Suppose $\tilde{x}$ is a floating point approximation to a nonzero real number x and that

$$\frac{|\tilde{x} - x|}{|x|} \approx \text{eps}$$

where eps is the unit roundoff. Explain why this means that $\tilde{x}$ is correct to "full machine precision."

For full credit you need to bring up three points.

eps is roughly the distance from 1 to the next largest floating point number. I.e., it is the floating point spacing near 1.

Consideration of the mantissa-exponent role in the floating point representation, the spacing of the floating point numbers in the vicinity of $|x|$ is roughly $|x|$eps.

$|\tilde{x} - x| \approx |x|$eps says that the difference between x and $\tilde{x}$ is roughly the spacing of the floating point numbers around $\tilde{x}$. That means that $\tilde{x}$ and x agree to full mantissa prescision.

There are different ways of weaving these facts together and partial credit was given in accordance with how much your answer reflected knowledge of these basics. There were lots of answers that were just paraphrases of the question, i.e., "full machine precision means all the bits are correct".

6. We wish to solve the Poisson problem

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \tag{1}$$

on a rectangle R with the value of $u(x, y)$ prescribed on the boundary of R.

(a) Using divided differences, show how (1) can be approximated at $P = (x_c, y_c)$ with a linear equation that involves the value of u at P and at its four "compass point" neighbors $E = (x_c + h, y_c)$, $W = (x_c - h, y_c)$, $N = (x_c, y_c + h)$, and $S = (x_c, y_c - h)$.

$$\frac{\frac{u(E) - u(P)}{h} - \frac{u(P) - u(W)}{h}}{h} + \frac{\frac{u(N) - u(P)}{h} - \frac{u(P) - u(S)}{h}}{h} = 0$$

i.e.

$$4u(P) - u(N) - u(E) - u(S) - u(W) = 0$$

5 points

For the set-up on the following page, 7 points for the matrix and 8 for the right hand side *assuming* that your part (a) was correct.

(b) A discrete solution procedure for (1) can be obtained if we apply the divided difference ideas on a grid of regularly spaced meshpoints. Here is a an example:

At each "+" we have an unknown u_i that is to be an approximation to $u(x,y)$. In the example we have 12 unknowns and their indices are displayed next to each "+". The boundary values at the red "o points are also displayed. Assuming that meshpoint spacing in both the x and y directions is the same, fill in the matrix and right hand side values of the linear system that defines the solution vector $u(1:12)$. Entries that are zero can be left blank.

$$
\begin{bmatrix}
4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 4 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 4 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4
\end{bmatrix}
*
\begin{bmatrix}
u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \\ u_{10} \\ u_{11} \\ u_{12}
\end{bmatrix}
=
\begin{bmatrix}
76 \\ 35 \\ 25 \\ 27 \\ 21 \\ 0 \\ 0 \\ 22 \\ 21 \\ 20 \\ 30 \\ 72
\end{bmatrix}
$$