Math 620 Comprehensive Exam 8/22/2012

Name:

Instructions: You must show all your work to receive full credit. Partial answers will only receive partial credit. Please choose 3 of the 4 problems to solve. Please indicate which 3 problems you would like graded. You may use scientific calculators; however, programmable calculators, laptops, etc. are not allowed.

[1] The *trisection* method is a modification of the bisection method used to find a root of $f(x) = 0$ over a bracketing interval $[a, b]$, where, instead of dividing the interval into two at each step, we divide it into three:

(T1) Define $c = (2a + b)/3$, $d = (a + 2b)/3$.

(T2) If $\text{sign}(f(b)) \cdot \text{sign}(f(d)) \leq 0$, set $a = d$
else if $\text{sign}(f(d)) \cdot \text{sign}(f(c)) \leq 0$, set $a = c, b = d$
else set $b = c$.

(In other words, we find the first subinterval (from the right) which must contain a root, and take it to be our new bracket)

(T3) If $b-a \leq 2\varepsilon$, accept $\xi = (a+b)/2$ as the approximation to the root and stop. Otherwise, go back to step (T1)

(a) Count the above cycle as one step. How many such steps (call this number n) will be required to ensure that the error is less than or equal to ε?

(b) On an average, do you expect the trisection method to be more computationally efficient than the bisection method? Justify. (Assume that all the computational work is concentrated in functional evaluations, and other steps are negligible.)

[2] Let p_1 be the linear interpolant of f at the points $x_0, x_1 (x_0 < x_1)$. Then by the theorem on interpolation error, if $f \in C^{(2)}[x_0, x_1]$,

$$f(x) - p_1(x) = (x - x_0)(x - x_1)\frac{f^{(2)}(\xi(x))}{2}, \; x_0 < x < x_1$$

for some $\xi(x)$ between x_0 and x_1.

(a) Determine $\xi(x)$ explicitly in the case $f(x) = \frac{1}{x}, x_0 = 1, x_1 = 2$.

(b) Will such a $\xi(x)$ still exist if $x_0 = -1, x_1 = 1$, and f as above? Investigate and explain how this case differs from part (a).

(c) How about if $x_0 = -1, x_1 = 1$, but $f(x) = |x|$ instead? Investigate and explain.

[3] (a) Explain how floating-point numbers are represented in binary computers, that is, how is a block of 32 bits or 64 bits in computer memory is related to the real number being represented. If you wish, you may pick the example of one number system concretely, such as single-precision IEEE standard, double-precision IEEE standard, or the "Marc-32 computer" (a term used in the book by Kincaid and Cheney). If so, state which one you are describing.
(b) Assume a mathematical representation for normalized computer numbers of the abstract form $\tilde{x} = (-1)^s \, (1.b_1 b_2 \ldots b_{p-1})_2 \, 2^E$ with $s \in \{0, 1\}$, $b_j \in \{0, 1\}$, $j = 1, \ldots, p-1$, and $E_{\min} \leq E \leq E_{\max}$. Derive and explain the formulas for the (i) smallest positive normalized number, (ii) largest positive normalized number, and (iii) machine epsilon.
(c) Discuss how there is a trade-off between the range of a number system (the size of the largest number; larger is better) and the relative accuracy of the number system (the size of machine epsilon; smaller is better).

[4] Derive the *local truncation error* $\tau_n(Y)$ of the numerical method

$$y_{n+1} = \frac{1}{2}(y_n + y_{n-1}) + \frac{h}{4}\Big(4f(x_{n+1}, y_{n+1}) - f(x_n, y_n) + 3f(x_{n-1}, y_{n-1})\Big), \quad n \geq 1,$$

for the ordinary initial value problem $y'(x) = f(x, y(x))$, $y(x_0) = Y_0$. Be sure to carry enough terms in the Taylor expansions to get the precise coefficient of the leading order of h and to verify whether the immediately following term cancels or not. The result should read

$$\tau_n(Y) = C\, h^p\, Y^{(q)}(x_n) + \mathcal{O}(h^r)$$

with some non-zero constant C and some integers p, q, and r and where $Y(x)$ denotes the solution to the differential equation. What is the order of accuracy of the method?