TAM 470 Introduction to Computational Mechanics: Assignment 2

Due: Friday, September 23, 2016.

Problem 1 (15 points): Moin. pp. 25-26, Exercise 1(a), (b), (d) [*not* (c)].

For (d), note that the author intends that *two applications of the first-derivative finite difference operator* means repeated application of the 2nd-order centered difference operator $\frac{\delta}{\delta x}$. Thus, if

$$v_j \;:=\; \frac{\delta u_j}{\delta x} \;\approx\; \left.\frac{du}{dx}\right|_{x_j}, \quad \text{then}$$

$$w_j \;:=\; \frac{\delta v_j}{\delta x} \;\approx\; \left.\frac{d^2u}{dx^2}\right|_{x_j}.$$

The question asks you to compare the leading-order truncation error in w_j to the result for the standard $\frac{\delta^2 u_j}{\delta x^2}$ formula.

Problem 2 (15 points): Moin. p. 26, Exercise 2.

Problem 3 (15 points): Consider the first-order backward difference formula,

$$\frac{\delta u}{\delta t} \;:=\; \frac{u^n - u^{n-1}}{h} \;=\; \left.\frac{du}{dt}\right|_{t^n} + \; c_1 h \; + \; c_2 h^2 \; + \; \text{h.o.t.},$$

with uniform spacing, $h := t^n - t^{n-1}$.

(a) Use Richardson extrapolation applied to interval widths h, $2h$, and $3h$ to improve the order of accuracy to $O(h^3)$. Show your derivation and the final result.

(b) Demonstrate that your scheme is 3rd-order accurate by applying it to $u(x) = e^x$ with $t^n = 1$ for $h = 0.1$ and $h = 0.05$.

Problem 4 (15 points): Moin. p. 44-45, Exercise 8 (a) and (b).

(c′): Instead of (c), repeat (a) using Gauss-Lobatto-Legendre (GLL) quadrature with the `zwgll.m` script provided on-line to generate the quadrature points and weights. Plot all of your error results on a single log-log graph for $n = 2$, 4, 8, 16,...,1024. (Number of *points* is $n + 1$.)

(d): Compare your Simpson rule results to the results when Richardson extrapolation is applied to the trapezoidal rule. What do you find about the two methods?

(e): Compare trapezoidal and GLL quadrature for $n = 1, 2, 3, ..., 50$ for the periodic function $y = e^{\cos 2\pi x}$ on the interval $[0, 1]$. Plot the error vs n on a semilogy plot. What do you observe about the two methods in this case?

NOTE: You may use the GLL solution as the "exact" solution when determining the error as a function of n for each of the methods by simply evaluating the GLL solution at a value of n that is larger than used in the experiments. ($n_{\max} + 10$ is probably reasonable.)

Problem 5 (15 points): Moin. p. 45, Exercise 9.

Problem 6 (15 points): Consider the integral,

$$\tilde{I} \;:=\; \int_0^{2\pi}\int_0^{2\pi} \cos(xy)\, dx\, dy.$$

(a): Plot a mesh showing the integrand vs. x and y on the GLL points for $n = 50$. (Use `mesh(X,Y,F)`.) The integrand is of the form $f_{ij} = \cos(x_i y_j)$. While it can be evaluated in matlab with a pair of `for` loops, it will be much faster to define mesh variables X and Y using `[X,Y]=ndgrid(x,y)` and then set `F=cos(X.*Y)`, where x and y are the vectors of $n+1$ GLL points mapped to $[0, 2\pi]$. (Understanding this approach will be important later when we solve PDEs.)

(b): Use GLL quadrature for $n = 1, 2, \ldots, 50$ to compute the approximate integral, $I_n \approx \tilde{I}$. Estimate the error by comparing the difference between your computed integral with that of one on a larger number of points and plot the convergence behavior on a semilogy plot.

(c): Verify your code by computing the integral for an integrand that you can compute exactly (on the same domain). Describe the function of your choosing, it's exact integral, how you computed the integral, and show the convergence plot for your verification test in semilogy form.