Assignment #2

Due Monday, 6 February at the start of class

Please read the Chapter 1 and Chapter 2 of the textbook R. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations*.

P7. Reproduce Figure 1.2 on page 6 of the textbook. In particular, write a MATLAB/OCTAVE/etc. code which generates the data shown in Table 1.1, by doing the calculations described by Example 1.1 with $u(x) = \sin x$ and $\bar{x} = 1$. Turn in both the code and the figure you generate. (*Do not waste paper by turning in these numbers, but please do use Table 1.1 to* check *that you are getting the right numbers.*)

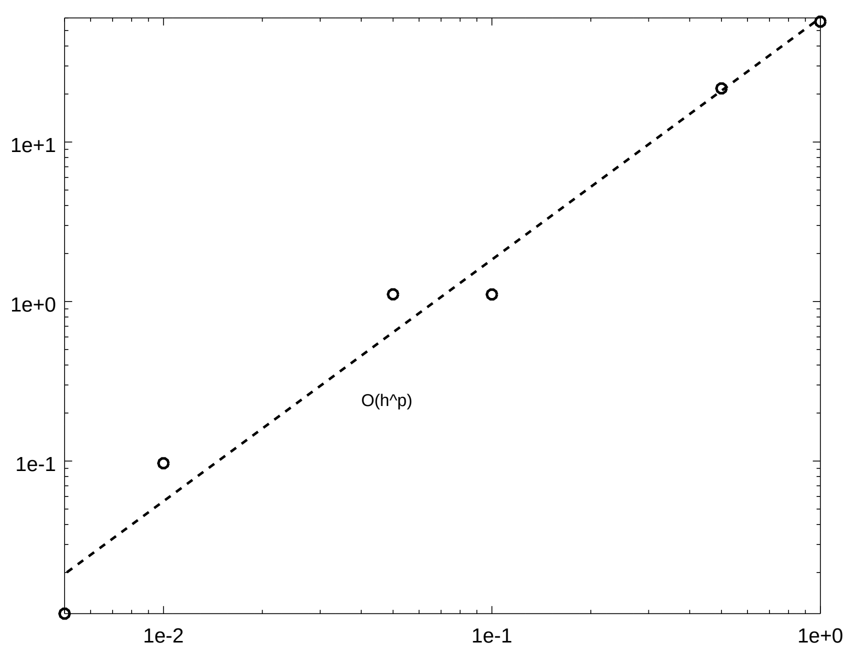
In particular, in MATLAB/OCTAVE use `loglog` to generate the graph, make sure to label the axes as shown (or in a comparable manner), and use `text` to put "D_0" and such labels in approximately the right locations. Note that the data should be shown as markers, but the lines between can be generated however is convenient.

P8. Suppose this table of "data" is samples of an $O(h^p)$ function:

h	1.0	0.5	0.1	0.05	0.01	0.005
Z	56.859	21.694	1.1081	1.1101	0.096909	0.011051

This data may be fitted (linear regression) by a function $f(h) = Mh^p$ for some values M and p, as in the following figure. (Note $M > 0$ and $p > 0$ are clear.) Find p by fitting a straight line to the data, and reproduce the figure. Your version should have the value of p filled in.

In MATLAB/OCTAVE you may use `polyfit` to find p, plus `loglog` and `text` as in **P7** above to generate the graph.



P9. **a)** Use the method of undetermined coefficients (section 1.2) to set up the 5×5 Vandermonde system that determines the fourth-order centered finite difference approximation to $u''(x)$ based on 5 equally-spaced points, namely

(1) $u''(x) = c_{-2}u(x-2h) + c_{-1}u(x-h) + c_0u(x) + c_1u(x+h) + c_2u(x+2h) + O(h^4).$

In particular, expand $u(x-2h), u(x-h), u(x+h), u(x+2h)$ in Taylor series and then collect terms on the right side of (1) to generate 5 linear equations in the 5 unknowns $c_{-2}, c_{-1}, c_0, c_1, c_2$. This linear system $A\,c = b$ will have numerical (constant) entries in the matrix A and entries of b which depend only on h.

b) Use MATLAB/OCTAVE/etc. to solve the linear system from part **a)**. The recommended way to do this is to use $h = 1$ in determining the right-hand side vector b, then solve the system numerically using the "backslash" method, and then write down the answer in a form like equation (1.11), with the correct power of h.

(*Feel free to use LeVeque's* `fdstencil` *to check your work, but it is not required.*)

P10. In section 2.4 the textbook uses finite differences to convert the boundary value problem

$$u''(x) = f(x), \quad u(0) = \alpha, \quad u(1) = \beta$$

into matrix equation $AU = F$, with A and F given in (2.10). Note that finite difference approximation D^2 from equation (1.13) is used for the u'' term, and that there are m unknowns $U_1, U_2, \dots, U_m$ based on a grid with $x_j = jh$ and $h = 1/(m+1)$.

Assuming p, q, x_L, x_R are real numbers with $x_L < x_R$, do a similar finite difference approximation for the more general problem

(2) $$u''(x) + p\,u'(x) + q\,u(x) = f(x), \quad u(x_L) = \alpha, \quad u(x_R) = \beta.$$

Use the same approximation D^2 for u'', and use approximation D_0 for u' as in equation (1.3). Use the same grid indexing with m unknowns $U_1, \dots, U_m$, and give the new formulas for x_j and the mesh width h. Compute A and F in $AU = F$. (*Note that entries of A and F will depend on p and q.*) Check your work by confirming that you can reproduce (2.10) by choosing appropriate constants in your method.

(*In a future problem I'll ask you to implement this as a code.*)