Assignment #2

Due Monday, 13 February 2023, at the start of class

Please read sections 1.1–1.4 and 2.1–2.10 from the textbook.[1] The problems on this Assignment especially covers material from 1.1–1.4 and 2.1–2.4.

Problem P7. Suppose this table of "data" gives samples of a function $Z(h)$:

h	1.0	0.5	0.1	0.05	0.01	0.005
$Z(h)$	28.43	10.747	0.5540	0.5555	0.04849	0.005521

This data may be fitted (linear regression) by a function $f(h) = Ch^p$ for some values C and p. Find these values by fitting a straight line to the *logarithms* of the data; in MATLAB you may use `polyfit`. Then graph the data *and* show the fitted line on the same axes, using MATLAB's `loglog` or similar.

Problem P8. Reproduce Figure 1.2 on page 6 of the textbook. In particular, write a MATLAB/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 $\bar{x} = 1$. Then generate the Figure, which has logarithmic scaling on both axes. Make sure to label the axes as shown, or comparably, and also make sure to put the labels ("D_0", etc.) in approximately the right locations; use text in Matlab. The data should be shown as markers, but the lines between can be generated however is convenient.

Turn in both the code and the figure you generate. However, please do *not* waste paper by turning in the numbers! Instead, use Table 1.1 to check that you are getting the right numbers.

Problem P9. *Before doing this exercise, read and understand Example 1.2 in section 1.2.*

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

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

In particular, expand $u(x - 2h), u(x - h), u(x + h), u(x + 2h)$ in Taylor series. Then collect terms on the right side of the above equation to generate a square linear system $A\,c = g$ in unknowns $c_{-2}, c_{-1}, c_0, c_1, c_2$. This system will have numerical (constant) entries in the matrix A, but the entries of vector g will depend on h.

---

[1]R. J. LeVeque, *Finite Difference Methods for Ordinary and Partial Diff. Eqns.*, SIAM Press 2007

(b) Use MATLAB/etc. to solve the linear system from part **a)**. A recommended way to do this is to use $h = 1$ in the vector g and solve the system numerically using the "backslash" method. Then write down the answer in a form like (1.11), inserting the correct power of h. Use $h = 0.5$ to confirm that you have captured the correct powers. (*Feel free to use LeVeque's* `fdstencil` *to check your work, but it is not required.*)

Problem 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). For any integer $m \geq 1$, this method is based on a grid with $h = 1/(m+1)$ and $x_j = jh$. There are m unknowns $U_1, U_2, \ldots, U_m$, located at the interior nodes $x_1, \ldots, x_m$. Note that finite difference approximation D^2 from equation (1.13) is used for the u'' term.

Assume q, x_L, x_R are real numbers with $x_L < x_R$. Similar to the method in section 2.4, create a finite difference approximation for the problem

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

Use the same approximation D^2 for u''. Use the same grid indexing with m unknowns $U_1, \ldots, U_m$, and give the new formulas for x_j and the mesh width h. State, in detail, A and F in $AU = F$. (*Note that entries of A will depend on q as well as h.*) Check that, by choosing appropriate constants, you can reproduce formulas (2.10).

Problem P11. Continuing along the lines of **P10**, set up a finite difference method for the most general linear, second-order Dirichlet boundary value problem:

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

Compared to **P10**, now $p(x), q(x)$ are arbitrary functions like $f(x)$. Use approximation (1.3), namely the centered finite difference D_0, for the u' term. State A and F in the linear system $AU = F$. (*In a future problem I'll ask you to implement this method.*)