

Chapter 1 Exercises

From: *Finite Difference Methods for Ordinary and Partial Differential Equations*
by R. J. LeVeque, SIAM, 2007. <http://www.amath.washington.edu/~rjl/fdmbook>

Exercise 1.1 (*derivation of finite difference formula*)

Determine the interpolating polynomial $p(x)$ discussed in Example 1.3 and verify that evaluation $p'(\bar{x})$ gives equation (1.11).

Exercise 1.2 (*use of `fdstencil`*)

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

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

- (b) Compute the coefficients using the matlab code `fdstencil.m` available from the website, and check that they satisfy the system you determined in part (a).
- (c) Test this finite difference formula to approximate $u''(1)$ for $u(x) = \sin(2x)$ with values of h from the array `hvals = logspace(-1, -4, 13)`. Make a table of the error vs. h for several values of h and compare against the predicted error from the leading term of the expression printed by `fdstencil`. You may want to look at the m-file `chap1example1.m` for guidance on how to make such a table.

Also produce a log-log plot of the absolute value of the error vs. h .

You should observe the predicted accuracy for larger values of h . For smaller values, numerical cancellation in computing the linear combination of u values impacts the accuracy observed.

Exercise 1.3 (*Generation of finite difference formula for a nonuniform grid*)

- (a) Use the book matlab codes `fdcoeffF.m` and `fdstencil.m` to write the subroutine `SubrGeneralFD(f, fkp, k, xb, xpts)` that computes the coefficients and produces as an output the finite difference formula corresponding to the grid points x_1, x_2, \dots, x_n (not necessarily uniform). The parameters of the subroutine are f and fkp which represent the function f and its k th derivative to be approximated, respectively; the letter k represent the order k of the derivative, xb is the point where the derivative is evaluated, and $xpts$ is a vector whose components are the grid points.
- (b) Your subroutine `SubrGeneralFD.m` should also compute the coefficients of the leading order terms of the error and should include in his output these error terms. The book code `fdstencil.m` can be adapted for this purpose.

- (c) From a main program `MainSubrGeneralFD.m`, where you supply the parameter values call `SubrGeneralFD.m`.

For the function $f(x) = e^{\frac{x}{3}}$, approximate its second derivative at $\bar{x} = 1/2$ by using the following grid points:

- (i) $xpts = [xb - 1e - 1; xb; xb + 1e - 1]$
- (ii) $xpts = [xb - 1e - 1; xb - 1e - 3; xb; xb + 1e - 2]$
- (iii) $xpts = [-3; 0; 1/2; 1; 4]$
- (iv) $xpts = [-3; 0; 1/3; 1; 4]$

For each experiment compute the actual error comparing the approximation and the exact value of the derivative. Then, compute the error from the leading order term of the previous item. Try to produce an output similar to `OutputSubrGeneralFD.pdf` found in our class web page.

Write a paragraph of at least four lines making comments about your results.

Hint: To call the subroutine use:

`SubrGeneralFD('f', 'fkp', k, xb, xpts).`

For computation of $f(x)$, use `feval(f, x)` in the subroutine.