Note: You will need some software to solve the first of these problems (and maybe others, depending on how you solve them.) As before, please put your code into a Python Notebook called YOURNAME_code_hw5.ipynb and submit this to your Dropbox.

Textbook Problems:

1. Exercise 3.1 (pg. 91)

Solution:

See "hw5_solutions.ipynb" for relevant code.

(a) see code

(b) dt = .001 is about the point at which further reductions in dt make no visible difference.

(c) With $S_0 = 2$ and $F_0 = 1$, the solutions are constant. With other values, solutions are sometimes periodic, sometimes not.

Other Problems:

1. (B&F, pg 29, #15) Use the 64-bit floating point format to find the decimal equivalent of the following floating-point machine numbers.

(a) 0 10000001010 1001001100000000000000000000000000000000000000000000
(b) 1 10000001010 1001001100000000000000000000000000000000000000000000
(c) 0 0111111111 01010011000000000000000000000000000000000000000000000

Solution:

I wrote some Python code to convert these strings of 1s and 0s into decimal numbers. The code is posted on the website. Using the code, I determined that these numbers were 3224, -3224, and 1.32421875.

2. (B&F, pg 29, #16) Find the next largest and smallest machine numbers in decimal form for the numbers given in the previous exercise.

Solution:

Next largest:

(a) 0 10000001010 1001001011111111111111111111111111111111111111111111
(b) 1 10000001010 1001001100000000000000000000000000000000000000000001
(c) 0 0111111111 01010011000000000000000000000000000000000000000000001

Next smallest:

(a) 0 10000001010 1001001011111111111111111111111111111111111111111111
(b) 1 10000001010 1001001100000000000000000000000000000000000000000001
(c) 0 0111111111 01010011000000000000000000000000000000000000000000001
3. (B&F, pg 185, #28) Derive a method for approximating \( f'''(x_0) \) whose error term is of order \( h^2 \) by expanding the function \( f \) in a fourth Taylor polynomial about \( x_0 \) and evaluating at \( x_0 \pm h \) and \( x_0 \pm 2h \).

**Solution:**

A fourth order taylor series around \( x_0 \) is:

\[
f(x) = f(x_0) + f'(x_0)(x-x_0) + \frac{f''(x_0)(x-x_0)^2}{2} + \frac{f'''(x_0)(x-x_0)^3}{6} + \frac{f''''(x_0)(x-x_0)^4}{24} + \mathcal{O}(h^5)
\]

Evaluating at \( \pm h \) and \( \pm 2h \) yield the following expressions:

\[
\begin{align*}
f(x_0 + h) &= f(x_0) + f'(x_0)h + \frac{f''(x_0)h^2}{2} + \frac{f'''(x_0)h^3}{6} + \frac{f''''(x_0)h^4}{24} + \mathcal{O}(h^5) \\
f(x_0 - h) &= f(x_0) - f'(x_0)h + \frac{f''(x_0)h^2}{2} - \frac{f'''(x_0)h^3}{6} + \frac{f''''(x_0)h^4}{24} + \mathcal{O}(h^5) \\
f(x_0 + 2h) &= f(x_0) + f'(x_0)2h + \frac{f''(x_0)4h^2}{2} + \frac{f'''(x_0)8h^3}{6} + \frac{f''''(x_0)16h^4}{24} + \mathcal{O}(h^5) \\
f(x_0 - 2h) &= f(x_0) - f'(x_0)2h + \frac{f''(x_0)4h^2}{2} - \frac{f'''(x_0)8h^3}{6} + \frac{f''''(x_0)16h^4}{24} + \mathcal{O}(h^5)
\end{align*}
\]

Subtracting the first two of these expressions and dividing by \( 2h \) yields:

\[
\frac{f(x_0 + h) - f(x_0 - h)}{2h} = f'(x_0) + f''(x_0)h^2 + \mathcal{O}(h^4),
\]

while subtracting the second two of these expressions and dividing by \( 4h \) yields:

\[
\frac{f(x_0 + 2h) - f(x_0 - 2h)}{4h} = f'(x_0) + f''(x_0)4h^2 + \mathcal{O}(h^4).
\]

Finally, subtracting the two right hand sides above yields:

\[
\frac{f(x_0 + 2h) - f(x_0 - 2h)}{4h} - \frac{f(x_0 + h) - f(x_0 - h)}{2h} = 3h^2 f'''(x_0) + \mathcal{O}(h^4).
\]

Dividing by \( 3h^2 \) yields our desired formula:

\[
f'''(x_0) = \frac{f(x_0 + 2h) - 2f(x_0 + h) + 2f(x_0 - h) - f(x_0 - 2h)}{12h^3} + \mathcal{O}(h^2)
\]

4. Suppose \( f(x) = xe^x \), and you intend to use the formula

\[
f'(x_0) \approx \frac{f(x_0 + h) - f(x_0)}{h}
\]

to approximate the derivative at \( x_0 = 2 \). Find the (approximate) optimal value of \( h \) for a machine whose roundoff error is \( \epsilon \).
Solution:
By Taylor series, the theoretical error in the formula
\[
f'(x_0) \approx \frac{f(x_0 + h) - f(x_0)}{2h}
\]
is given by \(f''(\zeta)/2h\) for some \(\zeta \in [x_0, x_0 + h]\). Since the second derivative of \(f(x) = xe^x\) is
\[
f''(x) = (2 + x)e^x,
\]
and this quantity is bounded by about 30 in the vicinity of \(x = 2\), it follows that the approximation error is bounded by about 15h. On the other hand, each term in the finite difference approximation has a roundoff error. On a 64-bit machine, the relative round off error for any number within the range of representable numbers is bounded by \(2^{-52}\), so the actual roundoff error is roughly bounded by
\[
f(2) \cdot 2^{-52} \approx 15 \cdot 2^{-52}.
\]
Since the finite difference approximation has two terms, each of them contributes to roundoff error, so total roundoff error is bounded by \(30 \cdot 2^{-52}\). It follows that total error is bounded by
\[
\text{roundoff error + approximation error} \leq \frac{30 \cdot 2^{-52}}{h} + 15h.
\]
Optimize this by differentiating with respect to \(h\) and equating the result to 0. It turns out that the optimal value of \(h\) is about \(2^{-25}\).

Challenge (optional):
• (B&F, pg 184, #22) Derive an \(O(h^4)\) five-point formula to approximate \(f'(x_0)\) that uses \(f(x_0 - h)\), \(f(x_0)\), \(f(x_0 + h)\), \(f(x_0 + 2h)\), and \(f(x_0 + 3h)\). [Hint: Consider the expression \(Af(x_0 - h) + Bf(x_0 + h) + Cf(x_0 + 2h) + Df(x_0 + 3h)\). Expand in fourth Taylor polynomials, and choose \(A\), \(B\), \(C\), and \(D\) appropriately.]