Homework Checklist

1. Did you read every problem in its entirety?

    Yes

2. Dud you include the code you used to produce the results and plots?

    Yes

3. Did you write in complete English sentences including correct spelling and punctuation?

    Yes

4. Do your plots have readable labels on the axes?

    Yes

5. Did you use appropriate line styles in your plots?

    Yes

6. Are the lines visible on the plots you submitted?

    Yes

7. Do your plots have appropriate captions?

    Yes

8. Do your plots have appropriate titles?

    Yes

9. Do your plots have appropriate legends?

    Yes

10. Is the scale of the axes’ correct (plain or log10)?

    Yes

11. If you are plotting errors, are all the errors non-negative?

    Yes+

12. If you have any handwritten notes, are these notes legible?

    I have no handwritten notes

13. Do your tables have appropriate column and/or row headings?

    Yes

14. Did you properly cite any sources you used in completing the homework?

    Yes

15. Did you properly identify and acknowledge collaborations with classmates?

    Yes

Signed, *Tanner O’Rourke*

Question 1

Think of a numerical method that depends on a parameter h, where h is set to a “small” value to achieve accuracy. Name the method, and write a precise and complete mathematical statement of the problem that this method addresses.

-One-sided forward difference! The problem being addressed in this method is the numerical approximation of the derivative. Otherwise, given a function f(x), with derivative f’(x), and a parameter h that is “sufficiently small” such that f(x) is continuous on [x, x+h], it aims to approximate the derivative of f’(x) at value x+h.

Question 2

Name another method that solves the same problem as the previous question. Why would you use one method over the other?

-Another problem that solves the same problem wold be Two-sided Central Difference. The reason to use one method instead of another is a difference in time and accuracy of computation between the methods. , While One-sided Forward Difference has error that is asymptotically proportional to h (For a small enough h), Two-sided Central Difference has error asymptotically proportional to h^2. We can use this fact to chose which method will be more accurate in different situations.

Question 3

Think of a particular setting for the problem where neither method would be appropriate. In other words, give an instance of the problem data (the given part of the problem) where both methods would fail to “work.”

-Neither, actually none, of the three methods used to do a numerical approximation of the derivative (one-sided forward, one-sided backward, central) will fail to if the function is not continuous.
For example, if either method was used to calculate:

$f(x) = \frac{1}{x} \leftarrow [-1, 1]$ at $x_0 = 0$

Because the function is not continuous on [-1, 1], neither method will work. This occurs because when we approximate with an $h \longrightarrow 0$, we start to yield increasingly larger round-off error as we use increasingly smaller values of h value to approximate f(0). Because the $\lim_{x \to 0^+} \frac{1}{x} = \infty$ and $\lim_{x \to 0^-} \frac{1}{x} = \infty$, a computer will compute an infinite value at $h \simeq 0$, and result in a division by 0 in both methods.

Question 4

Forget about computers altogether for the next couple of questions. Mathematically, what happens to the output of the method when the parameter h goes to zero? Use precise mathematical notation.

-In One-sided Forward Difference, we compute the approximation of the derivative as $f'(x) = \frac{f(x+h)-f(x)}{h}$, for sufficiently small values of h. Here, h is the discretization from x we use to approximate the derivative.
Lets consider $\lim_{h\to 0^+} \frac{f(x+h)-f(x)}{h}$, which models the approximation parameter h at increasingly smaller values. As h goes to 0, we compute the derivative at a values closer to x, yielding a smaller discretization error. However in the case that h equals 0 (otherwise impossible to represent on a computer), the method's calculation yield $f'(x+h) = \frac{f(x+0)-f(x)}{0} = \frac{0}{0} = 0$, resulting in an error.

Question 5

What is the asymptotic error estimate for the method you picked? What is the mathematical definition of the asymptotic error estimate? Be precise. Use mathematical notation.

One-sided Forward Difference: $O(h)$
The asymptotic error estimate defines an upper bound on the absolute value of a method's error for an approximation parameter h. It is defined as $|e(h)| \le C * h^p$. This means that for an h sufficienctly small and constant C, a method's error estimate (difference between real and computed value) is bounded above by C * h^p.

Question 6

How should the asymptotic error estimate be used to compare methods? In other words, what sort of interpretation does the asymptotics justify? What interpretation does it not justify? Here's another way to phrase this question: what is the value of the asymptotic error estimate? Does it provide a specific number for the error in the method? If not, what is it good for?

-The asymptotic error estimate gives us an upper bound on a method's error computed at parameters h. It gives us basis to make a generalized comparison of two method's worst-case error. Therefore if a method's asymptotic error estimate is of smaller degree than another, then it is more accurate method (i.e. $O(h)$ faster than $O(h^2)$). Because the asymptotic error estimate depends only on the parameter h and defines only an upper bound on error, it can't justify any relationship to time complexity or a specific error estimate.

Question 7

Using the definition of the asymptotic error estimate and assuming that the parameter h is in the asymptotic regime, show (i.e., derive) the connection between the convergence rate and the slope of the line of log(error) versus log(h).

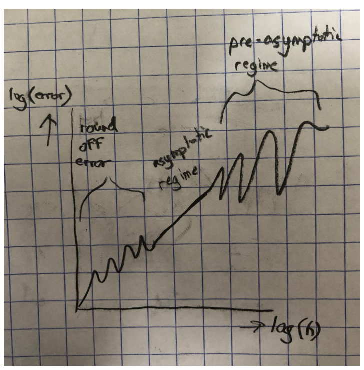
-If a method has an asymptotic error estimate of $O(h^p)$, for a parameter h in the asymptotic regime, we say that error is proportional to asymptotic error estimate such that:

$$\text{error} \simeq O(h^p) \longrightarrow \log(\text{error}) \simeq \log(\, O(h^p)\,) \longrightarrow \log(\text{error}) \simeq p * \log(h)$$

Therefore we can say that for a method with convergence rate $O(h^p)$, it will have an asymptotic regime that consists of the line log(error)= p*log(h), which is a linear line with slope p.

Question 8

Draw a cartoon picture that contains the essential features of the actual error convergence that you would expect to see for your method. Make sure the cartoon includes: the asymptotic regime, the pre-asymptotic regime, and the region dominated by round-off.



Question 9

Connect the asymptotic error estimate from your method to the cartoon picture you just drew. Specifically, how do the pieces of the error estimate appear in the picture? Why does the actual error behavior deviate from the asymptotic analysis?

-The region dominated by round-off error is where h is small enough that it can't that the computer yields a floating point arithmetic error when calculating h, yielding an inaccurate computation of error. The pre-asymptotic regime has the same model, as it is the region where the parameter h is too *large* to result in an accurate value. The asymptotic regime describes range of h's that are sufficiently small to produce an accurate value. Because the graph is on a log-log scale, we know that the *slope of the line* in the asymptotic regime directly correlates to the degree of the method's convergence rate. The actual error behavior in the plot, however, will have a slope p such that p < h. This is because the asymptotic analysis describes the “worst-case” upper-bound for error. Our implementation will therefore have convergence that is *roughly less than* p, but *looks* like h.

Question 10

The error associated with the parameter h is called discretization error, because the method is approximating a continuous mathematical object by a discrete and computable mathematical object. What is the relationship between discretization error and round-off error? How would you know when one dominates the other in an actual calculation?

-The goal of using a smaller parameter h is to decrease the discretization error, in order to get a better approximation for our method. However if $h < \epsilon_{\text{machine}}$, our computer will come across a floating point arithmetic error, thus yielding round-off error in that computation of h. Therefore, when h is sufficiently small such that a round-off error is produced, the round-off error will result in a loss of accuracy in the discretization error. This is easy to spot, because the discretization error of a method will not model a smooth function.

Question 11

In practice, we never know the true solution to the problem—otherwise we wouldn’t need the numerical method. Devise a principled procedure to study the error in the method when the true solution is unknown. Justify your procedure with precise math.

-For any discretization method using a parameter h, we know that a smaller value of h *should* produce smaller discretization error. We also know that the real value of our method is a constant. Therefore we can say:

$|\text{error}(h)| = \lambda$ - computed value, s.t. λ = real value of the function

If we consider that λ is a constant, we can then say conditionally that

|error(h)| = computed value

Therefore what we can do first is model/plot the **difference** in computed values from our method on a log-log scale with values of h:

$h = 2^{-k}$, k=5, 6,...,20

Although the real value is not factored into this conditional “error”, we can still use the slope of the asymptotic regime of this plot to derive an estimated asymptotic convergence. From here, we can compare this convergence rate with the mathematically derived convergence rate of the function we are trying to implement.

Question 12

Suppose someone else implements your chosen method and applies it to a test problem. This person shows you the results and claims they have a good approximation to the problem. How can you verify that their approximation is as good as claimed? You're not allowed to look at the code. But you are allowed to re-run the code with different values for h.

-To verify their approximation, we can re-run the code to compute the approximation for values of h such that:

h = 2^{-k} , k= 4,5,6,... *the h value that produced their best approximation*

After plotting the error approximation vs. h on a log-log scale, we should see that at the value of h that produced their best approximation is **inside** the asymptotic regime of the plot. If that value of h corresponds to the pre-asymptotic regime, that means we could make a better approximation. If this value of h corresponds to the region of round-off error, that means that it is an unreliable estimate.

Question 13

Suppse you write your own code and use it to approximate the solution to a given problem. Your colleague questions your results; it just doesn't look right. How would you defend your results to your colleague? How do you know you got a good approximation?

-FIrst, we could use a proof by Weak Induction to prove the method works as intented. If you prove that your code will yield the correct computed value for an iteration k_i ,starting with k_0, and your code also yields the correct computed value at an iteration k_{i+1}, then you prove that the method must work for all values of k. You can then, with your results, plot the log(error) vs. log(h). We can use this plot to find the asymptotic regime of the function and the convergence rate of our results. If there **is** an asymptotic regime, and the slope of the line in the asymptotic regime corresponds to the degree of the mathematically derived convergence rate, we can justify our results.

Question 14

Suppose you are resource limited. Your computing resources allow you to use h = 10–4 as the smallest possible value for h. How can you know whether this h is small enough for the desired accuracy?

-Assuming that you are able to calculate the real value for your method, first compute the error at values of h = 10^k for $k = 0, 1, 2, 3, 4$.

Next, plot the log(error) vs. log(h). Considering the definition of the asymptotic regime above and the definition of the asymptotic error estimate, the plot should show an asymptotic regime modeled by a smooth line. If you don’t see this regime, that means you aren’t able to calculate your function accurately enough to provide sufficiently small error and a low enough desired accuracy.