Midterm #1 – Practice

MATH 436/565 — Numerical Analysis
Midterm: Wednesday, Oct 4, 2023

Please print your name:

Bonus challenge. Let me know about any typos you spot in the posted solutions (or lecture sketches). Any mathematical typo, that is not yet fixed by the time you send it to me, is worth a bonus point.

Reminder. No notes, calculators or tools of any kind will be permitted on the midterm exam.

Problem 1. Determine all values C such that fixed-point iteration of $f(x) = \frac{2x^2}{1+3x}$ converges locally to C. In each case, determine the exact order of convergence as well as the rate.

Solution. Such values of C are necessarily fixed points of $f(x)$. To find these, we solve $\frac{2x^2}{1+3x} = x$. This equation has the obvious solution $x = 0$.

To find the nonzero solutions, we divide both sides by x to get $\frac{2x}{1+3x} = 1$ or, equivalently, $2x = 1 + 3x$, resulting in $x = -1$.

Hence, the fixed points are $x^* = 0$ and $x^* = -1$.

$f'(x) = \frac{4x(1+3x) - 6x^2}{(1+3x)^2} = \frac{4x + 6x^2}{(1+3x)^2}$

- $f'(-1) = \frac{-4+6}{(1-3)^2} = \frac{1}{2}$

  Since $|f'(-1)| < 1$, fixed-point iteration converges locally to -1. Convergence is linear with rate $|f'(-1)| = \frac{1}{2}$.

- $f'(0) = 0$

  Since $|f'(0)| < 1$, fixed-point iteration converges locally to 0. Convergence is at least quadratic. To be more precise, we need to compute $f''(0)$.

  $f''(x) = \frac{(4+12x)(1+3x)^2 - 6(4x+6x^2)(1+3x)}{(1+3x)^4} = \frac{4(1+3x)^2 - 6(4x+6x^2)}{(1+3x)^3} = \frac{4}{(1+3x)^3}$

  [Actually, there is no need to simplify $f''(x)$ since all we need is $f''(0)$.]

  $f''(0) = 4$

  Hence, fixed-point iteration converges locally to 0 with order 2 and rate $\frac{1}{2}|f''(0)| = 2$.

Problem 2. Consider $f(x) = (rx+1)(x^2-4)$ where r is some constant. Suppose we want to use Newton's method to calculate the root $x^* = 2$.

(a) For which values of r is Newton's method guaranteed to converge (at least) quadratically to $x^* = 2$?

(b) Analyze the cases in which Newton's method does not converge quadratically to $x^* = 2$. Does it still converge? If so, what can we say about the order and rate of convergence?

(c) For which values of r does Newton's method converge to $x^* = 2$ faster than quadratically?

Solution. Recall that we showed that, if $f(x^*) = 0$ and $f'(x^*) \neq 0$, then Newton's method (locally) converges to x^* quadratically with rate $\frac{1}{2}|f''(x^*)/f'(x^*)|$.

(a) Newton's method is guaranteed to converge to 2 provided that $f'(2) \neq 0$.

Since $f(2) = 0$, we have $f'(2) = 0$ if and only if 2 is a repeated root which happens if and only if $r = -1/2$.

Alternatively, we could compute $f'(x) = r(x^2-4) + 2x(rx+1)$ so that $f'(2) = 4(2r+1)$. Thus $f'(2) = 0$ if and only if $r = -1/2$.

In either case, we conclude that Newton's method converges (at least) quadratically to $x^* = 2$ if $r \neq -1/2$.

(b) We need to analyze the case $r = -1/2$. In that case $f(x) = \left(-\frac{1}{2}x + 1\right)(x-2)(x+2) = -\frac{1}{2}(x-2)^2(x+2)$.

Moreover, using the product rule, we compute that $f'(x) = -\frac{1}{2}(2(x-2)(x+2) + (x-2)^2) = -\frac{1}{2}(x-2)(3x+2)$.

Newton's method applied to $f(x)$ is equivalent to fixed-point iteration of

$$g(x) = x - \frac{f(x)}{f'(x)} = x - \frac{-\frac{1}{2}(x-2)^2(x+2)}{-\frac{1}{2}(x-2)(3x+2)} = x - \frac{(x-2)(x+2)}{3x+2} = x - \frac{x^2-4}{3x+2}.$$

Hence, $g'(x) = 1 - \frac{2x(3x+2) - 3(x^2-4)}{(3x+2)^2}$ so that, in particular, $g'(2) = 1 - \frac{4 \cdot 8}{8^2} = \frac{1}{2}$.

Since $0 \neq |g'(2)| < 1$ we conclude that Newton's method (locally) converges to $x^* = 2$. Moreover, the convergence is linear with rate $|g'(2)| = \frac{1}{2}$.

(c) Newton's method converges to 2 faster than quadratic if $f'(2) \neq 0$ (i.e. $r \neq -1/2$) and $f''(2) = 0$.

Continuing the computation from the first part, we calculate $f''(x) = 6rx + 2$.

Thus $f''(2) = 12r + 2 = 0$ if and only if $r = -1/6$.

Hence, Newton's method converges to 2 faster than quadratic if $r = -1/6$.

Problem 3.

(a) Express 123 in base 5.

(b) Which number is represented by $(1101.011)_2$?

(c) Express $31/14$ in base 2. If necessary, indicate which digits repeat.

(d) Represent -6.5 as a single precision floating-point number according to IEEE 754.

(e) Represent 2^{-4} as a single precision floating-point number according to IEEE 754.

(f) Express -27 in binary using the two's complement representation with 6 bits.

Solution.

(a) $123 = 24 \cdot 5 + \boxed{3}$, $\quad 24 = 4 \cdot 5 + \boxed{4}$, $\quad 4 = 0 \cdot 5 + \boxed{4}$

Hence, $123 = (443)_5$.

(b) $(1101.011)_2 = 2^3 + 2^2 + 2^0 + 2^{-2} + 2^{-3} = 13 + \frac{3}{8} = 13.375$

(c) Note that $31/14 = 2 + 3/14$ so that $31/14 = (10.\cdots)_2$ with $3/14$ to be accounted for.

- $2 \cdot 3/14 = \boxed{0} + 3/7$
- $2 \cdot 3/7 = \boxed{0} + 6/7$
- $2 \cdot 6/7 = \boxed{1} + 5/7$
- $2 \cdot 5/7 = \boxed{1} + 3/7$ and now things repeat…

Hence, $31/14 = (10.0011\cdots)_2$ and the final three digits 011 repeat: $31/14 = (10.0011011011\cdots)_2$

(d) $-6.5 = -(110.1)_2 = -(1.101)_2 \cdot 2^2 = \boxed{-}\underbrace{1.\boxed{101}}_{\text{binary}} \cdot 2^2$

The exponent 2 gets stored as $2 + 127 = \boxed{1000{,}0001}$.

Overall, -6.5 is stored as $\boxed{1}\boxed{1000{,}0001}\boxed{1010{,}0000{,}0000{,}0000{,}0000{,}000}$.

(e) $2^{-4} = 1 \cdot 2^{-4} = \boxed{+}\underbrace{1.\boxed{0}}_{\text{binary}} \cdot 2^{-4}$

The exponent -4 gets stored as $-4 + 127 = \boxed{0111{,}1011}$.

Overall, 2^{-4} is stored as $\boxed{0}\boxed{0111{,}1011}\boxed{0000{,}0000{,}0000{,}0000{,}0000{,}000}$.

(f) Since $27 = (011011)_2$, -27 is represented by 100101 (invert all bits, then add 1).

Alternatively, note that $-27 = -2^5 + 5$ and $5 = (101)_2$ to arrive at the same representation.

Problem 4.

(a) What is IEEE 754? Describe two popular choices that it offers. How many bits are used for what purpose?

(b) Give two reasons why floating-point numbers are used rather than fixed-point numbers.

(c) Give an example of a situation where one should not use floating-point numbers for nonintegers. Offer an alternative.

Solution.

(a) IEEE 754 is the most widely used standard for floating-point arithmetic. IEEE 754 offers several choices but the two most common are:

- *single precision*: 32 bit (1 bit for sign, 23 bit for significand, 8 bit for exponent)
- *double precision*: 64 bit (1 bit for sign, 52 bit for significand, 11 bit for exponent)

(b) Fixed-point numbers have some serious issues for scientific computation. Most notably:

- Scaling a number typically results in a loss of precision.

  For instance, dividing a number by 2^r and then multiplying it with 2^r loses r digits of precision (in particular, this means that it is computationally dangerous to change units).

- The range of numbers is limited.

  For instance, the largest number is on the order of 2^N where N is the number of bits used for the integer part. On the other hand, a floating-point number can be of the order of 2^{2^M} where M is the number of bits used for the exponent.

(c) One should not use floats when dealing with money. That is because, as we saw in class, an amount such as 0.10 dollars cannot be represented exactly using a float (when using base 2, as is the default in many circumstances) and thus will get rounded. This is very problematic when working with money.

In the case of money, the easiest way to avoid these issues is to store dollar amounts as cents. For the latter we can then simply use integers and work with exact numbers (no rounding).

Problem 5. We wish to compute the root of $f(x) = x^3 - 3$.

(a) Starting with the interval $[1, 2]$, apply two iterations of the bisection method. What exactly does it provide? What is the final resulting approximation of $\sqrt[3]{3}$?

(b) After how many iterations can we guarantee that the absolute error is less than 0.001?

(c) Describe in words how the regula falsi method proceeds different from the bisection method.

(d) After n iterations of the regula falsi method, is it a good idea to use the midpoint of the final interval as an approximation of the root?

(e) How does the secant method relate to the regula falsi method?

(f) Give one advantage of the secant method over the regula falsi method, as well as one advantage of the regula falsi method over the secant method.

(g) Starting with the initial approximation 1, apply two iterations of the Newton method. Write down (but do not compute) the absolute and relative errors.

(h) Determine whether the Newton method converges locally to $\sqrt[3]{3}$. If so, determine the exact order and rate of convergence.

(i) Give one advantage of the Newton method over the secant method, as well as one advantage of the secant method over the Newton method.

Solution. For comparison, $\sqrt[3]{3} \approx 1.44225$.

(a) Note that $f(1) = -2 < 0$ while $f(2) = 5 > 0$. Hence, $f(x)$ must indeed have a root in the interval $[1, 2]$.

- The midpoint of $[1, 2]$ is $c = \frac{a+b}{2} = \frac{1+2}{2} = \frac{3}{2}$. We compute $f(c) = c^3 - 3 = \frac{27}{8} - 3 = \frac{3}{8} > 0$.

  Hence, $\left[1, \frac{3}{2}\right]$ must contain a root of $f(x)$.

- The midpoint of $\left[1, \frac{3}{2}\right]$ is $c = \frac{a+b}{2} = \frac{1+3/2}{2} = \frac{5}{4}$. We compute $f(c) = -\frac{67}{64} < 0$.

  Hence, $\left[\frac{5}{4}, \frac{3}{2}\right]$ must contain a root of $f(x)$.

After 2 steps of the bisection method, we know that $\sqrt[3]{3}$ must lie in the interval $\left[\frac{5}{4}, \frac{3}{2}\right] = [1.25, 1.5]$.

The best approximation of $\sqrt[3]{3}$ at this point is the midpoint of the final interval: 1.375

(b) The width of the interval after n steps will be exactly $\ell = \frac{2-1}{2^n} = \frac{1}{2^n}$. Since $\sqrt[3]{3}$ is contained in this interval, the absolute error of approximating it with the midpoint is at most $\ell/2 = \frac{1}{2^{n+1}}$. We need to select n so that $\frac{1}{2^{n+1}} < 10^{-3}$. Knowing that $2^{10} = 1024$ (and $\frac{1}{1024} < \frac{1}{1000}$), we conclude that we need 9 iterations.

[Alternatively, especially with the help of a calculator, we can solve $\frac{1}{2^{n+1}} < 10^{-3}$ for n, we find $n > -\log_2(10^{-3}) - 1 = 3\log_2(10) - 1 \approx 8.97$. Hence, we need 9 iterations.]

(c) The regula falsi method proceeds like the bisection method. However, instead of using the midpoint $\frac{a+b}{2}$ of the interval $[a, b]$, it uses the root of the secant line of $f(x)$ through $(a, f(a))$ and $(b, f(b))$.

(d) No, we should not do that. After several iterations of the regula falsi method, one endpoint of the interval typically does not get updated anymore while the other endpoint converges to the desired root. Therefore, we need to use the most recently updated endpoint as the approximation.

(e) In each iteration, the regula falsi starts with an interval $[a, b]$ that is guaranteed to contain a root and replaces it with either the interval $[a, c]$ or the interval $[c, b]$.

Similarly, in each iteration, the secant method starts with two approximations a, b; however, it always replaces the “older” approximation with c (which is computed as in the regula falsi method).

(f) An advantage of the secant method is that, if it converges, it typically converges faster than linear. (Also, it does not require an initial interval that is guaranteed to contain a root.)

An advantage of the regula falsi method is that it is guaranteed to converge.

(g) We compute that $f'(x) = 3x^2$. Starting with $x_0 = 1$, we therefore have:

- $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = 1 - \frac{-2}{3} = \frac{5}{3}$
- $x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} = \frac{5}{3} - \frac{\frac{125}{27} - 3}{\frac{75}{9}} = \frac{331}{225}$

After 2 steps of Newton’s method, our approximation for $\sqrt[3]{3}$ is $\frac{331}{225} \approx 1.47111$.

The absolute error is $\left|\frac{331}{225} - \sqrt[3]{3}\right| \approx 0.02886$ and the relative error is $\left|\frac{\frac{331}{225} - \sqrt[3]{3}}{\sqrt[3]{3}}\right| \approx 0.02001$.

(h) Recall that we showed that, if $f(x^*) = 0$ and $f'(x^*) \neq 0$, then Newton’s method (locally) converges to x^* quadratically with rate $\frac{1}{2}|f''(x^*) / f'(x^*)|$.

Here, $f'(\sqrt[3]{3}) = 3 \cdot 3^{2/3} = 3^{5/3} \neq 0$. Moreover, $f''(x) = 6x$ so that $f''(\sqrt[3]{3}) = 6 \cdot 3^{1/3} = 2 \cdot 3^{4/3} \neq 0$.

Hence, Newton’s method (locally) converges to $\sqrt[3]{3}$ quadratically with rate $\frac{1}{2} \frac{2 \cdot 3^{4/3}}{3^{5/3}} = 3^{-1/3}$.

(i) An advantage of the Newton method is that it typically converges with order 2, which is faster than the secant method if $f(x_n)$ and $f'(x_n)$ can be computed for the cost of one function evaluation.

An advantage of the secant method is that it does not require the derivative $f'(x)$. Also, if evaluating the derivative requires an additional function evaluation (or worse), than the secant method actually converges to higher order than the Newton method ($\frac{1+\sqrt{5}}{2} \approx 1.618$ versus $\sqrt{2} \approx 1.414$ per function evaluation).

Problem 6. Let $g(x) = \frac{3}{4}\left(x + \frac{1}{x^3}\right)$

(a) Determine the fixed points of $g(x)$.

(b) For each fixed point x^*, determine whether fixed-point iteration of $g(x)$ converges locally to x^*. If so, determine the exact order and rate of convergence.

(c) This fixed-point iteration was obtained by applying Newton’s method to a function $f(x)$. Determine such a function $f(x)$.

Solution.

(a) Solving $\frac{3}{4}\left(x + \frac{1}{x^3}\right) = x$, we get $3\frac{1}{x^3} = x$ and, thus, $x^4 = 3$.

Hence, the fixed points are $\pm\sqrt[4]{3}$.

(b) Recall that, if $|g'(x^*)| < 1$, then fixed-point iteration of $g(x)$ converges locally to x^*.

$g'(x) = \frac{3}{4} - \frac{9}{4} \cdot \frac{1}{x^4}$

$g'(\pm\sqrt[4]{3}) = \frac{3}{4} - \frac{9}{4} \cdot \frac{1}{3} = 0$

Since $\left|g'(\pm\sqrt[4]{3})\right| < 1$, fixed-point iteration converges locally to each of $\pm\sqrt[4]{3}$. Convergence is at least quadratic. To be more precise, we need to compute $g''(\pm\sqrt[4]{3})$.

$g''(x) = 9\frac{1}{x^5}$

$g''(\pm\sqrt[4]{3}) = \frac{3}{\pm\sqrt[4]{3}} = \pm 3^{3/4}$

Hence, fixed-point iteration converges locally to each of each of $\pm\sqrt[4]{3}$ with order 2 and rate $\frac{1}{2}|g''(\pm\sqrt[4]{3})| = \frac{1}{2}\cdot 3^{3/4}$.

(c) We are looking for a function $f(x)$ such that $x - \frac{f(x)}{f'(x)} = g(x)$. Equivalently,

$$\frac{f'(x)}{f(x)} = \frac{1}{x - g(x)} = \frac{1}{x - \frac{3}{4}\left(x + \frac{1}{x^3}\right)} = \frac{1}{\frac{1}{4}x - \frac{3}{4}\frac{1}{x^3}} = \frac{4x^3}{x^4 - 3}.$$

This is a first-order differential equation which we can solve for $f(x)$ using separation of variables or by realizing that it is a linear DE. (Our approach below is equivalent to separation of variables.)

Note that $\frac{f'(x)}{f(x)} = \frac{\mathrm{d}}{\mathrm{d}x}\ln(f(x))$. Thus, integrating both sides of the DE,

$$\ln(f(x)) \;=\; \int \frac{4x^3}{x^4 - 3}\mathrm{d}x = \ln|x^4 - 3| + C.$$

We conclude that fixed-point iteration of $g(x)$ is equivalent to Newton's method applied to $f(x) = x^4 - 3$.

Problem 7.

(a) We have learned about the bisection method, the regula falsi method, the secant method, and the Newton method. For each method, answer the following:

(1) Does the method always converge?

(2) What does the method require as input? What does it provide as output?

(b) True or false? If the Newton method converges, it must converge quadratically.

(c) Newton's method applied to $f(x) = \sin(x^2 + 1)$ is equivalent to fixed-point iteration of which function $g(x)$?

(d) Give a condition such that fixed-point iteration of a function $f(x)$ converges locally to some value C.

(e) Suppose that x^* is a root of $f(x)$. When does Newton's method fail to locally converge to x^* with order of convergence at least 2?

Solution.

(a) Among these methods, only the bisection method and the regula falsi method always converge. The secant method and the Newton method only converge locally (that is, if the initial approximation is close enough to the desired root).

Apart from a function $f(x)$, the bisection method and the regula falsi method require an interval $[a, b]$ as input such that $f(a)f(b) < 0$ (so that $[a, b]$ is guaranteed to contain a root of $f(x)$). Each of these methods provides as output an interval that is guaranteed to contain a root of $f(x)$. In the case of the regula falsi method, the newly updated endpoint is the final approximation of a root (recall that the interval typically does not shrink down to a point because one endpoint gets stuck).

The secant method requires two initial approximations x_0, x_1 as input, while the Newton method requires a single initial approximation x_0. Each of these two methods provides as output an improved (fingers crossed!) approximation of the root.

(b) That is false. As can be seen for instance in the second part of Problem 2, Newton's method can converge linearly in the case of a repeated root.

(c) Newton's method applied to $f(x) = \sin(x^2 + 1)$ is equivalent to fixed-point iteration of

$$g(x) = x - \frac{f(x)}{f'(x)} = x - \frac{\sin(x^2 + 1)}{\cos(x^2 + 1)\cdot 2x} = x - \frac{1}{2x}\tan(x^2 + 1).$$

(d) C needs to be a fixed point of $f(x)$ (meaning that $f(C) = C$). If further $|f'(C)| < 1$, then fixed-point iteration of $f(x)$ converges locally to C. Combined, the condition is $f(C) = C$ and $|f'(C)| < 1$.

(e) If $f'(x^*) = 0$ (i.e. if x^* is a repeated root), then Newton's method does not converge quadratically.

Problem 8. Suppose we wish to approximate the function $f(x) = x^2 \ln(x)$.

(a) What is the 3rd Taylor polynomial $p_3(x)$ of $f(x)$ at $x = 1$?

(b) Provide an upper bound for the error of approximating $f(x)$ by $p_3(x)$ on the interval $[1, 3]$.

(c) Determine a value b such that the error of approximating $f(x)$ by $p_3(x)$ on the interval $[1, b]$ is less than 0.001.

Solution.

(a) $f'(x) = 2x\ln(x) + x$

$f''(x) = 2\ln(x) + 3$

$f'''(x) = \frac{2}{x}$

Hence, the 3rd Taylor polynomial of $f(x)$ at $x = 1$ is

$$p_3(x) = f(1) + f'(1)(x-1) + \frac{1}{2}f''(1)(x-1)^2 + \frac{1}{6}f'''(1)(x-1)^3 = (x-1) + \frac{3}{2}(x-1)^2 + \frac{1}{3}(x-1)^3.$$

Comment. There is typically no reason to expand this out since this approximation is intended to be used for x of the form $1 + \delta$, where δ is small (i.e. for x close to 1).

(b) Taylor's theorem implies that

$$f(x) - p_3(x) = \frac{f^{(4)}(\xi)}{4!}(x-1)^4$$

for some ξ between 1 and x.

We compute that $f^{(4)}(x) = -\frac{2}{x^2}$. This function is increasing on $(0, \infty)$ and so, in particular, on $[1, 3]$. Therefore, the maximum absolute value on $[1, 3]$ is taken at $x = 1$ or $x = 3$. Since $|f^{(4)}(1)| = 2$ and $|f^{(4)}(3)| = \frac{2}{9}$, we conclude that $|f^{(4)}(\xi)| \leqslant 2$.

On the other hand, $|(x-1)^4| \leqslant 2^4$ for all $x \in [1, 3]$.

We therefore conclude that the error on $[1, 3]$ is bounded by

$$|f(x) - p_3(x)| = \left|\frac{f^{(4)}(\xi)}{4!}(x-1)^4\right| \leqslant \frac{2}{4!} \cdot 2^4 = \frac{4}{3} \approx 1.333.$$

Comment. In this simple case, we can determine the maximal error exactly (without using Taylor's theorem). Since the function $f(x) - p_3(x)$ is decreasing on the interval $[1, 3]$, starting with the value 0, the maximal error must occur at $x = 3$ and is $\left|9\ln(3) - \frac{32}{3}\right| \approx 0.779$.

(c) By the same argument, the error on $[1, 1 + \delta]$ is bounded by

$$|f(x) - p_3(x)| = \left|\frac{f^{(4)}(\xi)}{4!}(x-1)^4\right| \leqslant \frac{2}{4!} \cdot \delta^4 = \frac{\delta^4}{12}.$$

We therefore need $\delta^4 < 0.012$ or $\delta < \sqrt[4]{0.012} \approx 0.331$ (the final approximation of course requires a calculator).

Correspondingly, b can be at most $1 + \sqrt[4]{0.012} \approx 1.331$.

Problem 9.

(a) Explain the following floating-point issue:

```
>>> 0.1 + 0.1 + 0.1 == 0.3
   False
>>> 0.1 + 0.1 + 0.1
   0.30000000000000004
```

(b) Explain the following floating-point issue:

```
>>> 10.**9 + 10.**-9 == 10.**9
   True
```

(c) Explain the following floating-point issue:

```
>>> def f(x):
        return (x-1)**99
>>> f(0.99) < 0
   True
>>> f(1.01) > 0
   True
>>> f(0.99) * f(1.01) < 0
   False
```

Solution.

(a) As we saw in class, 0.1 cannot be stored exactly as a floating-point number (when using base 2). Instead, it gets rounded up slightly. After adding three copies of this number, the error has increased to the point that it becomes visible as in the above output.

(b) We are adding the large number 10^9 to the small number 10^{-9}. In order to store $10^9 + 10^{-9}$ exactly, we need 19 decimal digits. That precision exceeds the precision that double precision floats (which is what Python currently uses) can provide. As a result, the 10^{-9} gets rounded away.

[More precisely, recall that double precision floats use 52 bits for the significand which, together with the initial 1 (which is not stored), means that we are able to store numbers with 53 binary digits of precision. This translates to about $53/\log_2 10 \approx 16$ decimal digits, less than what we need to store $10^9 + 10^{-9}$ exactly.]

(c) Both $f(0.99)$ and $f(1.01)$ are very small in absolute value. Their product is so small that it can no longer be differentiated from 0 in double precision floating-point arithmetic, and so it is treated as 0 in the above code. This issue is called an underflow.