Math 340 • Fall 2025 • Midterm Exam • October 23, 2025 • Prof. Matveev

Read each problem assignment carefully. Please explain all steps in your solution, briefly.

1) (14pts) Find the first **two non-zero terms** in the Taylor expansion of $\boldsymbol{f(x) = \frac{2x}{\sin(2x)}}$ near $x = 0$, using composition of Taylor expansions of two common elementary functions. Then, use your answer to estimate $\boldsymbol{f(0.6)}$.

Step 1: Use expansion $\sin(u) \approx u - \frac{u^3}{3}$ where $u = 2x$:

$$f(x) = \frac{2x}{\sin(2x)} \approx \frac{2x}{2x - \frac{(2x)^3}{3!}} = \frac{\cancel{2x}}{\cancel{2x}\left(1 - \frac{(2x)^2}{3!}\right)} = \frac{1}{1 - \frac{4x^2}{6}} = \frac{1}{1 - \frac{2}{3}x^2}$$

Step 2: Use geomertric series, $\frac{1}{1-u} \approx 1 + u$, where $u = \frac{2}{3}x^2$: $f(x) \approx \frac{1}{1 - \frac{2}{3}x^2} \approx \boxed{1 + \frac{2}{3}x^2}$

Step 3: Plug in $x = 0.6$: $f(x) \approx 1 + \frac{2}{3}0.6^2 = 1 + 2 \cdot 0.2 \cdot 0.6 = 1 + 2 \cdot 0.12 = \boxed{1.24}$

Note 1: there is no other way to find the Taylor polynomial of this function (L'Hopital rule is messy, opaque, but equivalent)

Note 2: actual value is close: $f(0.6) = 1.2875$

Note 3: partial credit given if you stopped after first step (i.e. didn't use geometric series)

---

2) (14pts) Which of the functions below suffer catastrophic **loss of all significant digits** when computed in double precision for small $|x|$? Write down the precise **condition on** $|x|$ when this happens. Finally, **modify** each expression suffering from this problem so that it provides a more accurate result when evaluated for very small $|x|$:

$$(a)\ f(x) = \frac{1}{16} - \frac{1}{16 + x^8}, \quad (b)\ g(x) = 1 - \cos(2x^4), \quad (c)\ h(x) = \frac{1}{1 + \sin(x)}.$$

Only **(a)** and **(b)** involve subtraction of quantities which become equal in the limit $x \to 0$:

(a) $f(x) = \frac{1}{16} - \frac{1}{16 + x^8}$: will be rounded off to **zero** when $\frac{x^8}{16} \le 10^{-16} \Rightarrow \boxed{|x| \le \sqrt{2} \cdot 10^{-2} \approx 0.014}$

Modification to avoid catastrophic loss of significance: $f(x) = \frac{1}{16} - \frac{1}{16 + x^8} = \frac{(16 + x^8) - 16}{16(16 + x^8)} = \boxed{\frac{x^8}{16(16 + x^8)}}$

(b) $g(x) = 1 - \cos(2x^4)$: will also return **zero** when the cosine function is rounded off to 1. This happens when the 2nd term in the Taylor expansion of cosine becomes too small to add to one:

$$\cos(2x^4) \approx 1 - \frac{(2x^4)^2}{2!} = 1 - 2x^8 \to 1 \quad \text{when} \quad 2x^8 \le 10^{-16} \Rightarrow \boxed{|x| \le \frac{10^{-2}}{\sqrt[8]{2}}} \approx 0.009$$

Modification: below this level, simply replace $g(x) = 1 - \cos(2x^4) \approx 1 - (1 - 2x^8) = \boxed{2x^8}$

Note: these results are very accurate, as you can verify by evaluating these in MATLAB below and above these thresholds

3) (12pts) Convert $x = 37.1$ from decimal to **binary** number system, keeping 8 digits after the fractional point. Then, write this binary number is standard scientific notation (significant part times an integer power of 2).

First, convert the integer part (use division by 2 algorithm, if you prefer): $37 = \underset{2^5}{32} + \underset{2^2}{4} + \underset{2^0}{1} = 100101_2$

Now, the fractional part, using our simple algorithm:
$$\begin{pmatrix} 0.1 * 2 = 0 + 0.2 \\ 0.2 * 2 = 0 + 0.4 \\ 0.4 * 2 = 0 + 0.8 \\ 0.8 * 2 = 1 + 0.6 \\ 0.6 * 2 = 1 + 0.2 \end{pmatrix}$$

The sequence **0011** has to repeat (since we reached 0.2 again), so: Binary(37.1) = $100101.0\overline{0011}\ldots = 1.0010100\overline{0011}\cdot 2^5$

Note: full credit given as along as at least 8 bits in the final fraction were shown

---

4) (14pts) Consider the iteration $x_{n+1} = g(x_n)$ with $g(x) = x + \frac{7}{16}(x^2 - 1)$

a) Find all fixed points of this iteration, and determine their stability (**hint:** fixed points are integers)

$$(a)\ r = r + \frac{7}{16}(r^2 - 1) \Rightarrow r^2 - 1 = 0 \Rightarrow r_{1,2} = \pm 1 \Rightarrow \begin{cases} r_1 = -1: \ |g'(r_1)| = 1 + \frac{7}{8} r_1 = 1 - \frac{7}{8} = \frac{1}{8} < 1: \ \text{Stable} \\ r_2 = +1: \ |g'(r_2)| = 1 + \frac{7}{8} r_2 = \frac{15}{8} > 1: \ \text{Unstable} \end{cases}$$

b) For each **stable** fixed point, find the **number of iteration** needed to reach the fixed point to within an absolute distance of 2^{-52} (machine precision), assuming initial distance from the fixed point of $\epsilon_0 \approx 1/2$.

$$C \equiv |g'(r_1)| = \frac{1}{8} \Rightarrow \varepsilon_{n+1} \approx C\,\varepsilon_n \Rightarrow \varepsilon_n \approx C^n \varepsilon_0 = \frac{1/2}{8^n} = \frac{1}{2}\frac{1}{(2^3)^n} = \frac{1}{2^{3n+1}} \le \frac{1}{2^{52}} \Rightarrow 3n+1 \ge 52 \Rightarrow \boxed{n \ge 17}$$

---

5) (14pts) Write down a function $f(x)$ whose root (zero) equals $r = 1/\sqrt[3]{4}$. Your function should only involve multiplication, integer powers, and/or division (no fractional powers). Write down the Newton's iteration for finding the root of this function, and perform two iterations to approximate $r = 1/\sqrt[3]{4}$, starting with $x_0 = 1$. Compare your result with the actual value of the root: $1/\sqrt[3]{4} \approx 0.63$.

$f(x) = 4x^3 - 1$ has a root at $r = \frac{1}{\sqrt[3]{4}}$

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

Two iterations: $x_0 = 1 \Rightarrow x_1 = \frac{2}{3} + \frac{1}{12} = \frac{9}{12} = \frac{3}{4} \Rightarrow x_2 = \frac{2}{3}\frac{3}{4} + \frac{1}{12\left(\frac{3}{4}\right)^2} = \boxed{\frac{1}{2} + \frac{4}{27}} = \frac{35}{54}$

Since $\frac{4}{27} \approx 0.15$, our second iterate is $x_2 \approx 0.5 + 0.15 = 0.65$, quite close to the exact value of 0.63 after only 2 iterations, starting from a poor guess!

6) (12pts) Consider 3 different fixed-point iterations methods, each characterized by distinct error behavior, as follows:

$$(a)\ \epsilon_{n+1} = 0.01\epsilon_n, \quad (b)\ \epsilon_{n+1} = 10\epsilon_n^3, \quad (c)\ \epsilon_{n+1} = \epsilon_n^2$$

Compute $\boldsymbol{\epsilon_2}$ for each case, assuming that $\boldsymbol{\epsilon_0 = 10^{-2}}$. How many iterations are needed to reduce the error to the level of $\boldsymbol{\epsilon_n \le 10^{-16}}$, in each of the three cases? Which of the three iterations converges the fastest?

(a) $\varepsilon_1 = 0.01\varepsilon_0 = 10^{-2}\cdot 10^{-2} = 10^{-4}$; $\quad \varepsilon_2 = 0.01\varepsilon_1 = 10^{-2}\cdot 10^{-4} = 10^{-6}$

Number of iterations to reach 10^{-16} :

$\varepsilon_n = (0.01)^n \varepsilon_0 = 10^{-2n}\cdot 10^{-2} = 10^{-(2n+2)} = 10^{-16}$ when $2n+2=16 \Rightarrow \boxed{n=7}$

(b) $\varepsilon_1 = 10\varepsilon_0^3 = 10\cdot\left(10^{-2}\right)^3 = 10^{-5}$; $\quad \varepsilon_2 = 10\varepsilon_1^3 = 10\cdot\left(10^{-5}\right)^3 = 10^{-14}$

Almost there, but could do **one more iteration**: $\varepsilon_3 = 10\cdot\left(10^{-14}\right)^2 = 10^{-27} \Rightarrow \boxed{n=3}$ **Best method!**

(c) $\varepsilon_1 = \varepsilon_0^2 = \left(10^{-2}\right)^2 = 10^{-4}$; $\quad \varepsilon_2 = \varepsilon_1^2 = \left(10^{-4}\right)^2 = 10^{-8}$

One more iteration needed: $\varepsilon_3 = \left(10^{-8}\right)^2 = 10^{-16} \Rightarrow \boxed{n=3}$

[**Side note**: general result in this case is $\varepsilon_n = \varepsilon_0^{2^n} = \left(10^{-2}\right)^{2^n} = 10^{-2^{n+1}}$]

Conclusion: the higher the method order, the faster is the convergence, even if the error coefficient is unfavorably large.

---

7) (14pts) Perform least-squares fit of the model $\boldsymbol{y_{model}(t;p,q) = p + q\cdot t^2}$, where p and q are unknown parameters, to the data set $\{(t_k, y_k)\} = \{(-1,1),(0,5),(1,3)\}$. Compute all **residuals**, and **plot** the model and the data points. Make sure to explain your solution method, very briefly.

The model basis functions are $\phi_1(t) = 1$, $\phi_2(t) = t^2$, therefore our system is (see course notes for details):

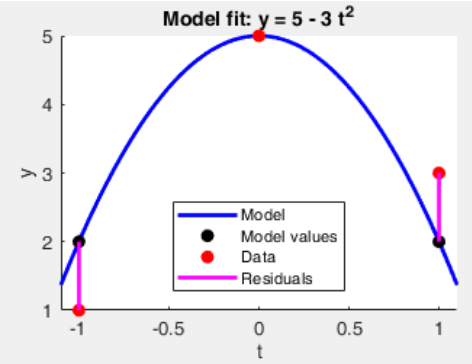
$$\begin{pmatrix} \sum_{k=1}^N \phi_1(t_k)^2 & \sum_{k=1}^N \phi_1(t_k)\phi_2(t_k) \\ \sum_{k=1}^N \phi_1(t_k)\phi_2(t_k) & \sum_{k=1}^N \phi_2(t_k)^2 \end{pmatrix}\begin{pmatrix} p \\ q \end{pmatrix} = \begin{pmatrix} \sum_{k=1}^N 1 & \sum_{k=1}^N t_k^2 \\ \sum_{k=1}^N t_k^2 & \sum_{k=1}^N t_k^4 \end{pmatrix}\begin{pmatrix} p \\ q \end{pmatrix} = \begin{pmatrix} \sum_{k=1}^N y_k \\ \sum_{k=1}^N y_k t_k^2 \end{pmatrix}$$

Then, plug in data points $\{(-1,1),(0,5),(1,3)\}$:

$$\begin{cases} \sum_{k=1}^N t_k^4 = \sum_{k=1}^N t_k^2 = 1+0+1 = 2, \ \sum_{k=1}^N y_k = 1+5+3 = 9 \\ \sum_{k=1}^N t_k^2 y_k = 1\cdot 1 + 0\cdot 5 + 1\cdot 3 = 4 \end{cases} \Rightarrow \begin{pmatrix} 3 & 2 \\ 2 & 2 \end{pmatrix}\begin{pmatrix} p \\ q \end{pmatrix} = \begin{pmatrix} 9 \\ 4 \end{pmatrix} \quad \begin{matrix} R_1 - R_2 \to R_1 \\ \frac{1}{2}R_2 - R_1 \to R_2 \end{matrix} \quad \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} p \\ q \end{pmatrix} = \begin{pmatrix} 5 \\ -3 \end{pmatrix}$$

Thus, $\boldsymbol{p = 5}$, $\boldsymbol{q = -3}$, so the model is $\boldsymbol{y_{model} = 5 - 3t^2}$

Residuals, $y_{data}(t_k) - y_{data}(t_k)$, equal $r_1 = 1 - (5-3) = \mathbf{-1}$; $r_2 = 5 - 5 = \mathbf{0}$; $r_3 = 3 - (5-3) = \mathbf{+1}$



8) (6pts) Consider the following MATLAB program. Write down the output value of variable **F** after the first **two iterations** of the for-loop, in terms of the values of **x** and **q**. What elementary function is this code computing?

```
function F = myFunction(x, q)
   F = 1;   term = 1;
   for k = 1 : 4
         term = term * q * x / k;
         F = F + term;
         q = q - 1;
   end
end
```

End of loop iteration k = 1:

- term -> term * q * x / k = q * x
- F -> F + term = 1 + q * x
- q -> q – 1;

End of loop iteration k = 2:

- term -> term * q * x / k = q (q – 1) * x^2 / 2
- F -> F + term = 1 + q * x + q (q – 1) * x^2 / 2
- q -> q – 2;

Conclusion:

This MATLAB function is computing the 4th degree Taylor polynomial approximation of $f(x) = (1 + x)^q$