Answers below ↓.

Assignment Sheet 3 of MTH2051/3051

Please read the following instructions carefully. If in doubt, please raise issues in the discussion forum.

i) The submission deadline is 23.55pm on Friday of Week 10.

ii) Please complete the template files provided through Moodle. Do not change filenames or headers.

iii) Your code is not required to check whether the input is valid.

iv) Symbolic computation and high-level Matlab commands are prohibited and result in zero marks for the task in which they were used.

v) The marking scheme is full marks for a correct implementation and no marks for an incorrect implementation. (Where possible, tasks are broken into subtasks, so a single error has no large effect.)

vi) Please collect your answers to theoretical questions in a file `notes.pdf` or `notes.jpg`.[1]

vii) Please submit all your Matlab files and your file `notes.xyz` as individual files through Moodle.

Assignment 3.1. (polynomial interpolation, 9 marks)
In this exercise, you will see how polynomial interpolation can behave in computational examples, and why we often use splines instead.

a) Please complete the file `myNewtonCoefficients.m` by implementing an algorithm that computes the scheme of divided differences. Recall that the divided differences can be organised in a lower triangular matrix as explained in Remark 5.7, and that the Newton coefficients can be obtained as in Theorem 5.9. $\boxed{4}$

b) Please complete the file `myEvaluateNewtonPolynomial.m` by implementing the Horner-type algorithm from Remark 4.14. $\boxed{3}$

c) Please run the script `wrapper_3_1.m`, and explain how it is possible that the interpolation polynomials with $N = 8$ approximate

   i) the Heaviside function and $\boxed{1}$

   ii) the rational function f specified in the wrapper $\boxed{1}$

   so poorly. Why is Theorem 5.13 not preventing the error from becoming large?

3-1 c) Please run the script `wrapper_3_1.m`, and explain how it is possible that the interpolation polynomials with $N = 8$ approximate

i) the Heaviside function and $\boxed{1}$

ii) the rational function f specified in the wrapper $\boxed{1}$

so poorly. Why is Theorem 5.13 not preventing the error from becoming large?

i) The Heaviside function despite being continuous, it's derivative is not continous which is a necessity for Th^m 5.13 to prevent error from becoming large.

ii) Despite the rational function being continously differentiable, with increasing n such that $f^n(\xi)$ where ξ lies between our interval is getting much larger, hence our error gets larger.

3-2.

c) Please run the script `wrapper_3_2.m`, and explain the features of the error curve of the forward difference quotient based on Proposition 6.5:

i) Why is the wriggly part wriggly, and why is it diverging as $h \to 0$? $\boxed{1}$

ii) Why is the minimum where it is (position and height)? $\boxed{1}$

iii) Why is the straight part straight, and why is its slope 1? $\boxed{1}$

i) From 6.5 we know that the error is bound by,

$\dfrac{2(\Delta+\Delta^2)}{h} M_0 + \Delta M_1 + \dfrac{h}{2} M_2$, so with smaller h, the error is less bounded, and the wriggly parts are a result of round-off errors for which are inconsistent.

ii) It sits at approximately $h = \sqrt{\Delta}$ $2.52671e^{-9}$ where Δ refers to the machine precision of the machine running the algorithm, and the height is

$\frac{2\Delta}{h} m_0 + \Delta M_1 + \frac{h}{2} M_2$. which is $\approx 2.5671e^{-9}$.

iii) Round off errors no longer become a problem at this point.

3.3

e) Please run the script `wrapper_3_3.m`, determine the slopes of the plotted error curves, and explain your findings by referring to results from the lecture notes and exercises. Hints: $\boxed{1}$

i) What error estimates do we have for the composite quadrature methods we are investigating? $\boxed{1}$

ii) How does the slope of a line in the error plot relate to the error of the quadrature method that generates it? $\boxed{1}$

$$Q_S = \frac{0.125 - 0.015625}{1.6591e^{-5} - 4.0322e^{-9}} \approx 10.2.$$

suggesting error that has the form;

$$e(h) \approx \frac{1}{10} h^{10}$$

$$Q_T = \frac{0.125 - 0.15625}{0.0257 - 0.0004} \approx 4.3$$

$$e(h) \approx 2 \cdot h^4$$

Assignment 3.2. (numerical differentiation, 5 marks)
In this exercise, you will examine the interplay between the theoretical truncation error and the effect of round-off errors in numerical differentiation.

a) Please complete the file `myForwardDQ.m` by implementing the forward difference quotient from Example 6.2. $\boxed{1}$

b) Please complete the file `myCentralDQ.m` by implementing the central difference quotient from Example 6.2. $\boxed{1}$

c) Please run the script `wrapper_3_2.m`, and explain the features of the error curve of the forward difference quotient based on Proposition 6.5:

   i) Why is the wriggly part wriggly, and why is it diverging as $h \to 0$? $\boxed{1}$

   ii) Why is the minimum where it is (position and height)? $\boxed{1}$

   iii) Why is the straight part straight, and why is its slope 1? $\boxed{1}$

Assignment 3.3. (numerical integration, 6 marks)
In this exercise, you will see in a computational example how composite quadrature reduces the quadrature error when the integration interval is divided into more and more subintervals.

a) Complete the file `myTrapezoidal.m` by implementing trapezoidal rule. $\boxed{1}$

b) Please complete the file `mySimpson.m` by implementing Simpson rule. $\boxed{1}$

c) Please complete the file `myCompTrapezoidal.m` by implementing the composite trapezoidal rule. $\boxed{1}$

d) Please complete the file `myCompSimpson.m` by implementing the composite Simpson rule. $\boxed{1}$

e) Please run the script `wrapper_3_3.m`, determine the slopes of the plotted error curves, and explain your findings by referring to results from the lecture notes and exercises. Hints: $\boxed{1}$

   i) What error estimates do we have for the composite quadrature methods we are investigating? $\boxed{1}$

   ii) How does the slope of a line in the error plot relate to the error of the quadrature method that generates it? $\boxed{1}$

Challenge 3.4. (adaptive quadrature)
Adaptive numerical methods avoid unnecessary work. Instead of applying a full composite quadrature to approximate $\int_a^b f(x)dx$, work out a recursive algorithm based on the following instructions.

Fix an error tolerance $\varepsilon > 0$. On any given interval $[\alpha, \beta]$ we have to deal with during the recursion, we compare the plain midpoint rule with the composite midpoint rule with $N = 4$. If we find that

$$|Q_M^1[f;\alpha,\beta] - Q_M^4[f;\alpha,\beta]| \leq \varepsilon \frac{\beta-\alpha}{b-a},$$

then we have reason to believe that both values are reasonably close to the true integral and approximate

$$\int_\alpha^\beta f(x)dx \approx Q_M^4[f;\alpha,\beta].$$

Otherwise we subdivide the interval $[\alpha, \beta]$ into the intervals $[\alpha, (\alpha+\beta)/2]$ and $[(\alpha+\beta)/2, \beta]$ and apply the same procedure to both. If we start the recursion with the interval $[a, b]$, then we would expect that

a) the algorithm terminates in finite time, and

b) we approximate the integral $\int_a^b f(x)dx$ with an error of roughly ε.

Explore under which conditions this is true, and explain your findings with statements from the lecture notes.

let $n = 2$

newton coefficients exist here.

$$\Rightarrow \begin{aligned} y_0 &= \boxed{s[x_0]} \\ y_1 &= s[x_1] \leftarrow \boxed{s[x_0, x_1]} = \left\{ \frac{s[x_1] - s[x_0]}{x_1 - x_0} \right. \\ y_2 &= s[x_2] \leftarrow s[x_1, x_2] \leftarrow \boxed{s[x_0, x_1, x_2]}. \end{aligned}$$

$$s[x_0, x_1, x_2] = \overbrace{\frac{s[x_1, x_2] - s[x_0, x_1]}{x_2 - x_0}}.$$

$$\left\{ \begin{array}{l} \text{for } i = 2 : \text{length}(y) \\ \quad A_{i1} = y_i \\ \quad \text{for } j = 2 : i \\ \quad\quad A_{ij} = \dfrac{A_{i,j-1} - A_{i-1,j-1}}{x_i - x} \end{array} \right\}$$

horners scheme, let $n = 3$.

$\Rightarrow$

$$((c_3(x - x_2) + c_2)(x - x_1) + c_1)(x - x_0) + c_0$$

if nodes empty

$y =$

$y = [c_0 + (x - x_0)] \cdot$ function call

$[c_0 + (x - x_0)] [[c_1 + (x - x_1)) \cdot (_2 + (x - x_2)$