Do all computations in double-precision unless otherwise asked. Answer what is asked in the question; the inclusion of irrelevant facts may result in a small point deduction. Hand in a single pdf file to MarkUS with your code uploaded in separate files but not your solutions file, and also hand in your code separately to MarkUs. If you handwrite your assignment, illegible solutions or parts of solutions will not be graded and may receive zero. If you typeset everything in latex, 5 bonus points will be awarded to your assignment. If you typeset the assignment, you are allowed and encouraged to use this file as a provided template.

1. [30 points] Let $a > 0$. Consider the problem of numerically evaluating the integrals

$$y_n = \int_0^1 \frac{x^n}{x+a} dx \tag{1}$$

for $n = 0, 1, \ldots, 60$. Note that

$$y_n + ay_{n-1} = \int_0^1 \frac{x^n + ax^{n-1}}{x+a} dx = \int_0^1 x^{n-1} dx = 1/n. \tag{2}$$

and that

$$y_0 = \ln(1+a) - \ln(a) \tag{3}$$

(a) [5 points] What can you say about the value of the integral y_n for $n > 1$? What is it bounded above and below by?

(b) [10 points] Consider the algorithm:

1. Evaluate y_0
2. for $n = 1, 2, \ldots, 60$, compute

$$y_n = \frac{1}{n} - ay_{n-1} \tag{4}$$

Implement this algorithm and test for various values of a, e.g. $a = 10$, $a = 1/10$, $a = 2$. For each value of a, what does your algorithm output for y_{60}?

(c) [5 points] For each value of a, does the algorithm agree with the bounds you obtained in part (a)? If not, why not?

(d) [5 points] For which values of a (not just the ones listed) does the algorithm produce the correct result? Why?

(e) [5 points] How would you modify the algorithm in part (b) to compute approximately correct results? Hint: $\lim_{n\to\infty} y_n = 0$.

2. [30 points] We have seen in class the formula

$$\frac{f(x+h) - f(x)}{h} \tag{5}$$

used as an approximation to the derivative $f'(x)$. This is a first-order approximation because the error is $\mathcal{O}(h)$.

(a) [10 points] Using Taylor series expansion for $f(x+h)$ and $f(x-h)$, derive a second order approximation.

(b) [10 points] Let $f(x) = \sin(x)$, and $x = 0.4$. Using values of h in $\{10^{-1}, 10^{-2}, \ldots, 10^{-17}\}$, produce two log-log plots of the value h on the x-axis and the error on the y-axis, one for the first-order method, and one for the second-order method. For both plots, at what value of h is the error approximately minimized, and what is approximately the minimum?

(c) [10 points] What is the advantage of the second order method compared to the first order method? Why does the second order method have this advantage? You may assume that the roundoff error is η/h and the discretization error is h or h^2 depending on the method, and there are no additional sources of error.

3. [40 points] Consider the problem of computing $\sqrt{a}$, assuming you don't have access to library functions and can only compute add, subtract, multiply, divide. This can be converted to a root-finding problem: the function $f(x) = x^2 - a$ has a zero x^* at the value that satisfies $\sqrt{a} = x^*$. We want to compute x^* to an accuracy of 10^{-8}.

(a) [5 points] Show that $[1, a]$ brackets the root x^*.

(b) [10 points] Let $g(x) = x - \frac{1}{2a}(x^2 - a)$. Show that g is a contraction mapping on the interval $[1, a]$.

(c) [25 points] Let $a = 7$. Implement bisection method with the bracket in part (a), fixed point iteration defined in part (b), Newton's method with $x_0 = 1$, and secant method with $x_0 = 1$ and $x_1 = a$ and complete the following table:

	number of iterations	computed root x^*	rate of convergence
Bisection			
Fixed-point iteration			
Newton's method			
Secant method			