Problem 1. Prove that the closed Newton-Cotes rule $Q_{NC(m)}$ will compute the integral

$$\int_0^1 x^{k-1}dx = \frac{1}{k}$$

exactly for $k = 1, 2, \cdots, m$. Here the quadrature nodes are the fixed points $x_j = (j-1)/(m-1)$ for $j = 1, 2, \cdots, m$. Therefore, the weights $w_1, w_2, \cdots, w_m$ satisfy

$$w_1 x_1^{k-1} + w_2 x_2^{k-1} + \cdots + w_m x_m^{k-1} = \frac{1}{k}, \quad k = 1, 2, \cdots, m.$$

These conditions define a *linear* system. Write a function `ComputeClosedNewtonCotesWeights` to solve this system for the weights (returning vector of coefficients), and compare your computed weights with those given by M. Abramowitz and I. Stegun (in `ClosedNewtonCotesWeights` these weights are hardcoded). For comparison you can use a norm of the following defference vector:

```
norm(ComputeNewtonCotesClosedWeights(m)-NewtonCotesClosedWeights(m))
```

Problem 2. Exercise 5.2.4(c), Sauer's textbook, page 263.

Problem 3. The three-point Gauss-Legendre formula for approximating integrals has the form

$$\int_{x_k}^{x_{k+1}} f(x)dx = h\left[w_1 f(x_k + c_1 h) + w_2 f(x_k + c_2 h) + w_3 f(x_k + c_3 h)\right], \qquad h = x_{k+1} - x_k,$$

Consider nonlinear system:

$$\begin{aligned} w_1 + w_2 + w_3 &= 1, \\ w_1 c_1 + w_2 c_2 + w_3 c_3 &= 1/2, \\ w_1 c_1^2 + w_2 c_2^2 + w_3 c_3^2 &= 1/3, \\ w_1 c_1^3 + w_2 c_2^3 + w_3 c_3^3 &= 1/4, \\ w_1 c_1^4 + w_2 c_2^4 + w_3 c_3^4 &= 1/5, \\ w_1 c_1^5 + w_2 c_2^5 + w_3 c_3^5 &= 1/6. \end{aligned}$$

These conditions ensure that over the interval $[0; 1]$ the rule integrates exactly all polynomials of degree at most 5. **(a)** Numerically solve the system using Newton's method.
(b) Using your results from **(a)**, write a function to implement a composite Gauss-Legendre rule to approximate the integral

$$\int_{-\pi/2}^{\pi/2} e^x \cos(5x)dx$$

using 2, 4, 8, 16, and 32 subintervals. Compare with the exact answer, and plot the partition number versus the error.

Problem 4. In class we discussed both the error estimate

$$\left|\int_a^b f(x)dx - Q_{NC(3)}\right| = \frac{1}{2880} M_4 (b-a)^5,$$

for the three-point, closed, Newton-Cotes (Simpson) rule, as well as the error estimate

$$\left| \int_a^b f(x)dx - Q_{NC(4)} \right| = \frac{1}{6480} M_4 (b-a)^5,$$

for the four-point, closed, Newton-Cotes (Simpson 3/8) rule, where in each case M_4 is a bound on $|f^{(4)}(x)|$ for $x \in [a,b]$. As remarked, the Simpson 3/8 rule has a slightly better error bound, but requires one extra function evaluation. The similar scaling in these estimates suggests that the Simpson rule is superior to the Simpson 3/8 rule if our measure is *accuracy per function evaluation.*

That the Simpson rule is indeed better in this measure may be confirmed in the composite-rule setting. For example, both $Q^{(3)}_{NC(3)}$ and $Q^{(2)}_{NC(4)}$ are based on $3(3-1)+1 = 2(4-1)+1 = 7$ points, whence each requires seven function evaluations.

(a) Approximate $\int_0^2 \arctan(x)dx$ with each of the seven-point composite rules above, computing errors against the exact answer obtained by the Fundamental Theorem of Calculus.

(b) Use the general error formula [Eq. (12) of the notes `quad3`] to prove that, in the composite-rule setting, the error estimate for the Simpson rule is better (in terms of accuracy per function evaluation) than the one for the Simpson 3/8 rule.

Problem 1. Code functions for Euler, Trapezoid, and RK4 methods. For the problem from Example 6.25 (Sauer, p. 334) use these function to demonstrate dependence of local and global truncation error for every of these methods on a grid step (in one plot described below)

$$h = 0.5,\ 0.3,\ 0.1,\ 0.03,\ 0.05,\ 0.01,\ 0.003,\ 0.005,\ 0.001,\ 0.0005.$$

Compare with derived formulas and discuss. For this use one (!) plot for each type of error (local and global) in coordinates $\log(error)$ vs $\log(h)$, explain why you have straight lines in such coordinates for any power-like function h^n and what has to be the slope in this case.

Problem 2. Code two functions for a Backward Euler method using for the first one fixed point iterations and Newton's method for the second one.

Use this function to reproduce Figure 6.22 for Example 6.25 (see above), compare errors (local and global) for Euler, Backward Euler (use any subroutine of two coded before), Trapezoid, and RK4 methods for steps $h = 0.3$, $h = 0.1$, $h = 0.01$.