Numerical Methods 35006
Computer Lab 6: Numerical integration and Quadrature

1. Create python code that uses the trapezoidal rule to integrate a function f using the function call

   ```
   result = qtrapn(f,a,b,N)
   ```

   where `f` is the handle to a single variable function, `a` and `b` are the limits of integration and `N` is the number of intervals in the integration region. Test your function by integrating

   $$f(x) = x^3 + 1$$

   on the interval $[0, 2]$.

2. Using your function from Q2, compute the result of integration as you increase the number of intervals `N` from 4 to 1024, doubling each time.

3. Create python code that uses Simpson's rule to integrate a function f using the function call

   ```
   result = qsimpn(f,a,b,N)
   ```

   where the arguments are the same as in Q1. Compare your result with that of Q1, and save both of your functions to a new module `myquad.py`.

4. Using your function `qtrapn` or otherwise, create a refining trapedzoidal integrator, which doubles the number of points in the interval until a specified tolerance has been achieved. It should be called using the function call

   ```
   result = qtrapz(f,a,b,tol)
   ```

   Test this code on the function given above, and think of another function to test it on. Save your integrator to `myquad.py`.

5. In python the zeros and weights for Gaussian quadrature using Legendre polynomials can be found by calling the function `p_roots`, which can be accessed from the module

   ```
   from scipy.special import p_roots
   ```

   The roots and weights for a quadrature involving `n` points can then be obtained by

   ```
   x,w = p_roots(n)
   ```

   Numerically integrate the function

   $$\int_{-1}^{1} e^{x^2} \mathrm{d}x$$

   using Gauss-Legendre quadrature, for values of n points between 1 and 12.

6. Modify your code in Q5 to compute the integral of

   $$\int_{1}^{3} e^{-2x} cos(x) \mathrm{d}x$$

   using Gauss-Legendre quadrature.

7. Create a function `quadz`, that uses Gauss-Legendre quadrature to compute the integral of an arbitrary function `f` over an interval $[a, b]$ to within a given tolerance. The function call should be

   ```
   result = quadz(f,a,b,tol)
   ```

   Test the integral against the functions from earlier in the lab to check that it is working. Save it to your `myquad` module.