Numerical Mathematics:
Homework 5

Due on T.B.D at 24:00pm

Problem 1

In this problem, we are going to build our own "software" to do numerical integration.

(a) Write a function which applies the Simpson's rule to evaluate an integral between -1 and 1. Call it `[ I ] = simpsonquadrature( f , n )`. The first argument is a function itself, the second argument denotes the number of segments (i.e. no. of piewise quadratric functions) in which the interval is divided. You can test your code by making quick simple inline functions using the @ symbol (please look into the documentation about this).

(b) Do the same, no but now for Gaussian quadrature method. Call it `[ I ] = gaussianquadrature( f , n )`. The second argument `n` implies that the n roots of the n-th Legendre polynomials are used as absiccas. As explained in class, find the roots of the n legendre polynomials by finding the eigenvalues of the appropriate matrix. You can use the `eig` command for that. The weights are found by looking at the first entries of the corresponding eigenvectors.

(c) Can you explain how one could use the written matlab functions to evaluate integrals on a different interval, without making adjustments to the underlying code? Hint: apply a transformation to the function.

Problem 2

Consider the following two integrals:

$$I_1(x) = \int_{-1}^{1} e^{-x^2} dx, \quad I_2(x) = \int_{-1}^{1} 1 - |x| dx$$

Notice that the integrand of the first integral is "smooth", whereas the second integral has a kink at the origin.

(a) Write a matlab function which evaluates the above integrals, for let's say, up to $n = 300$. Consider the absolute difference between the numerical approximation and the true value of the integral. Plot this error as a function of n, by using a log scale for the y axis. Generate this graph for both methods: Simpson and Gaussian (add legends to denote which is which). Adapt the following structure: call the function `[ ] = plotconvergence( func )` , if `func=1` it should plot the first integral, and if `func=2` is should plot the second integral.

(b) To gain better insights on what is happening with the evaluation of the second integral, let us generate a plot of the actual function that is being integrated by the integration scheme. In particular, let us plot the difference between $1 - |x|$ and the numerically integrated function. Call the function `[ ] = ploterrorfunction( method, n )` which generates this plot. If `method='simpson'` consider the simpson and if `method='gaussian'` consider gaussian quadratures. Notice that for the simpson rule, you can simply write the error function by hand. Also, note that the the Lagrange interpolation function which you wrote in homework 3 may come in handy now. You can just copy it as a subroutine into your file.

(c) Explain your observations of the results in question a and question b. Be as detailed as possible. For example, when does Gaussian do well, and when does it not? What is causing the problem?

Problem 3

Consider the trapezoidal rule:

$$I_n = \frac{(b-a)}{n}\sum_{i=1}^{n}\frac{f(x_i)+f(x_{i-1})}{2}$$

Establish the error bound:

$$|I - I_n| \leq K\frac{(b-a)^3}{12n^2}, \quad K := \max_{x\in[a,b]} f''(x)$$

Hint! use integration by parts to show that:

$$I - I_n = \frac{(b-a)}{n}\sum_{i=1}^{n}\frac{f(x_i)+f(x_{i-1})}{2} - \sum_{i=1}^{n}\int_{x_{i-1}}^{x_i} f(x)dx = \sum_{i=1}^{n}\int_{x_{i-1}}^{x_i}(x-\hat{x}_i)f'(x)dx, \qquad \hat{x}_i := \frac{x_{i-1}+x_i}{2}$$

then apply integration by parts again and do some standard shenanigans.

Problem 4

The Laguerre polynomials $a_0(x), a_1(x), a_2(x), \ldots$ form an orthogonal set on $[0,\infty)$ and satisfy:

$$\int_0^\infty e^{-x}a_i(x)a_j(x)dx = 0, \quad i \neq j.$$

The polynomial $a_n(x)$ has n distinct zeros $x_1, x_2, \ldots, x_n$ on $[0,\infty)$. Construct a n-point quadrature formula that evaluates:

$$I = \int_0^\infty e^{-x}f(x)dx$$

and has precision $2n-1$. You must prove the statement that your quadrature formula has precision $2n-1$.

Problem 5

In this problem we are going to learn how quadratures can be used to solve a type of equation which arises very often in practical applications. Let $f(x)$ be a function defined on $[0,1]$, $K(x,y)$ be a function defined on $[0,1]\times[0,1]$, and $\lambda \in \mathbb{R}+$ some positive real parameter. Consider the following equation:

$$u(x) = \lambda\int_0^1 K(x,y)u(y)dy + f(x)$$

where $u(x)$ is the unknown to be determined. Notice in particular that $u(x)$ appears on both the left hand side and inside the integral. Such equations are called integral equations (as opposed to differential equations!). In particular, the equation above is a so-called Fredholm integral equation of second kind. Such problems show up regularly in mathematical physics.

Consider, for example, the following simple boundary value problem:

$$\frac{d^2u(x)}{dx^2} = \lambda u(x) + 1, \quad u(0) = 0, \quad u(1) = 0$$

It can be shown that the solution $u(x)$ to the above equation must satisfy the fredholm equation with:

$$K(x,y) = \begin{cases} x(y-1), & 0 \leq x \leq y \\ y(x-1) & y \leq x \leq 1 \end{cases}, \quad f(x) = \frac{1}{2}x(x-1)$$

(a) The exact solution to the concerning boundary value problem is given by:

$$u(x) = c_1 e^{\sqrt{\lambda}x} + c_2 e^{-\sqrt{\lambda}x} - \frac{1}{\lambda}, \quad \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ e^{\sqrt{\lambda}} & e^{-\sqrt{\lambda}} \end{bmatrix}^{-1} \begin{bmatrix} \frac{1}{\lambda} \\ \frac{1}{\lambda} \end{bmatrix}$$

Verify this by substituting this into the differential equation and checking for the boundary conditions.

We are going to write numerical code to solve the Fredholm equation (and hence, also the boundary value problem). For this, we will be using quadratures. Let $x_i := i/n$ for $i = 0, \ldots, n$ and consider:

$$u(x_i) = \lambda \int_0^1 K(x_i, y)u(y)dy + f(x_i).$$

Apply the trapezoidal rule by selecting the points $y_i = x_i := i/n$ for $i = 0, \ldots, n$ to yield:

$$u(x_i) = \frac{\lambda}{n} \sum_{i=0}^{n-1} \frac{1}{2}(K(x_i, y_i)u(y_i) + K(x_i, y_{i+1})u(y_{i+1})) + f(x_i)$$

(b) Let $u_n = \begin{bmatrix} u(x_0) & u(x_1) & \ldots & u(x_n) \end{bmatrix}^T$ and $y_n = \begin{bmatrix} y(x_0) & y(x_2) & \ldots & y(x_n) \end{bmatrix}^T$. Write down the corresponding linear system which needs to be solved.

(c) Write a matlab function called `[ ] = solveFredholm(lambda, n )` to solve the integral equation. The function should only return a plot showing the error between the true solution and the approximate solution obtained from solving the integral equation. It may be handy to use the matlab routine `interp1` to subtract the two solutions.