Universidad de El Salvador

Facultad de Ciencias Naturales y Matemáticas

Escuela de Matemáticas

Theme: Final Class Project

Cathedra: Numerical analysis

Professor: Msc. Carlos Gámez

Students

Guevara Tobar, Marcela Guadalupe GT13002

Serrano Santos, Kevin Francisco SS13019

University City - June 7, 2016

Problem 1 ***Simulation.*** *Complete the estimation for the probability of a neutron to emerge from the lead wall seen in class an explained in section 13.3 of Cheney.*

Solution. For this problem, we made a program named 'Neutron', which basically makes the following steps:

1. We made a nx7 matrix of random numbers on the interval [0,1]. In our program, n=5000. It simulates that n neutrons try to exit the lead wall.

2. Then, we simulatet the deflection angle that each neutron gets after evety collision.

3. If we have for one neutron that x $\geq$ 5, it means that the neutron exited the five units of tickness of the wall.

4. After all the simulations are made, we compute the average of the probabilities of the neutrons to exit the wall.

5. In order to have a more stable result, we run this program 50 times and take the average of the 50 runs to have the approximation of the probability for a neutron to exit the lead wall.

Our results showed that de probability lies between the 1.8% and the 1.9%.

Problem 2 ***Implement Brent's Method.*** *While secant and false position formally converge faster than bisection, one finds in practice pathological functions for which bisection converges more rapidly. These can be choppy, discontinuous functions, or even smooth functions if the second derivative changes sharply near the root. Bisection always halves the interval, while secant and false position can sometimes spend many cycles slowly pulling distant bounds closer to a root. Ridder's method does a much better job, but it too can sometimes be fooled. Is there a way to combine superlinear convergence with the sureness of bisection? The answer to this is yes: using Bren'ts method.*

Solution. In this problem, We create a function that finds the zero of a function in a given interval:

1. INPUTS:
   (a) f: function from the equation f(x)= 0
   (b) a,b : work interval [a,b]
   (c) tol : Tolerance Error
   (d) MAX : Maximum number of iterations
2. OUTPUT;
   (a) p, wehre p it's the root in the given interval.

Brent's method is a complicated but popular root-finding algorithm combining the bisection method, the secant method and inverse quadratic interpolation. It has the reliability of bisection but it can be as quick as some of the less-reliable methods. The algorithm tries to use the potentially fast-converging secant method or inverse quadratic interpolation if possible, but it falls back to the more robust bisection method if necessary.

The algorithm works as follow:

1. If $f(a)f(b) > 0$ then, there's no root in the given interval.

2. To use inverse quadratic interpolation, the algorithm needs another value, which is obtained as follow, $c = b$ and $f(c) = f(b)$.

3. In each iteration, first rename a, b, c and adjust bounding interval d if $f(b)$ and $f(c)$ have the same sign. then, the interval $[a, b]$ is exchanged.

4. Then, the convergence is verified, and the function creates another value as follow, $xm = 0.5(c - b)$

5. If $|xm|$ is smaller than the new tolerance, then the algorithm has found the root. Else, inverse quadratic interpolation is used to find the root, if interpolation failed, bisection is used.

When the number of iterations exceed MAX, then the function gives an aproximation with a greater tolerance than 'tol'. Else, the algorithm gives the root accurate to within 'tol'.

Problem 3 ***Moler, Numerical Computing with MATLAB, exercise 6.1, modified.*** *Implement Adaptive Quadrature using Simpson's Rule as worked in class. Complete the following table, where the blank spaces correspond to the number of function evaluations needed to obtain the desired precision.*

Solution. In order to give a solution to this problem, we made a function named 'Cuadratura', who has the following characteristics:

1. INPUTS:
   - (a) f: the fuction which we are going to approximate its integral.
   - (b) A,B: the work interval [A,B].
   - (c) tol: the tolerance required.
2. OUTPUTS:
   - (a) n: number of function evaluations.
   - (b) app: the approximation of the integral.
   - (c) In case the number of function evaluations exceeds 2000, then we assume there is a point of discontinuity and it's not possible for the function to compute the integral in the given interval. The function returns a message of error and the approximation at that moment.

$f(x)$	a	b	tol	$\#function\ eval$
$x^2e^{-x} + cos(x)$	-1	1	1e-6	61
$x^2e^{-x} + cos(x)$	-1	1	1e-8	205
$sin(x)$	0	π	1e-8	129
$cos(x)$	0	$(9/2)\pi$	1e-6	321
$\sqrt{x}$	0	1	1e-8	345
$\sqrt{x}log(x)$	eps	1	1e-8	565
$tan(sin(x)) - sin(tan(x))$	0	π	1e-8	Not possible to compute the integral.
$1/(3x-1)$	0	1	1e-4	Not possible to compute the integral.
$x^{8/3}(1-x)^{10/3}$	0	1	1e-8	85
$x^{25}(1-x)^2$	0	1	1e-8	57
$\frac{1}{(x-3)^2+0.01} + \frac{1}{(x-9)^2+0.04}$	0	2	1e-6	49

Problem 4 ***Monte Carlo.*** *What is the expected value of the volume of a tetrahedron formed by four points chosen randomly inside the tetrahedron whose vertices are (0,0,0), (0,1,0),(0,0,1), and (1,0,0)? (The precise answer is unknown!)*

Solution. In this problem, we find the equation of the plane that gives us the conditions for randomly generated points, the equation was found by the formula 3 points for a plane. The plane equation is as follows $z + x + y = 1$. Thus, the conditions for the volume of the tetrahedron are:

$$V = \begin{cases} z + x + y \leq 1 \\ 0 \leq x \leq 1 \\ 0 \leq y \leq 1 \\ 0 \leq x \leq 1 \end{cases}$$

For this, we have two m-files, one of these is a function that gives the expected value of the tetrahedron by Monte Carlo's method. The other give us the average of 100 runs function. In this way, we have the expected value of the tetrahedron.