Universidad de El Salvador

Facultad de Ciencias Naturales y Matemática

Escuela de Matemática

Proyecto Final de Análisis Númerico I

Alumnos:
Mario Salvador Rivera Calero RC13012
Jonathan Vladimir Ramírez Juárez RJ13001

Profesor:
Msr Carlos Gámez.

Junio 7 de 2016

The Problems.

$\boxed{\textbf{Problem 1.}}$ **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.

$\boxed{\textbf{Problem 2.}}$ **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. Ridders' 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. Read and implement Brent's method from [Press, et al., p. 354]. Show how your program works.

$\boxed{\textbf{Problem 3.}}$ **[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.

$f(x)$	a	b	`tol`	# function eval
$x^2e^{-x} + \cos x$	-1	1	1e-6	
$x^2e^{-x} + \cos x$	-1	1	1e-8	
$\sin x$	0	π	1e-8	
$\cos x$	0	$(9/2)\,\pi$	1e-6	
$\sqrt{x}$	0	1	1e-8	
$\sqrt{x}\log x$	eps	1	1e-8	
$\tan(\sin x) - \sin(\tan x)$	0	π	1e-8	
$1/(3x-1)$	0	1	1e-4	
$x^{8/3}(1-x)^{10/3}$	0	1	1e-8	
$x^{25}(1-x)^2$	0	1	1e-8	
$\dfrac{1}{(x-3)^2+0.01} + \dfrac{1}{(x-9)^2+0.04} - 6$	0	2	1e-6	

Suggestion: Learn about the use of global variables.

$\boxed{\textbf{Problem 4.}}$ **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!)

References

[Press, et al.] Press, W. H., Numerical Recipes in Fortran 77 (Second Edition), Cambridge, Cambridge University Press.

1. Ejecución de Algoritmos

Problema 1.

Paso 1) Abrir Octave/MATLAB y entrar a la carpeta donde están guardados los archivos .m.

```
octave:1> cd Escritorio/
octave:2> cd Proyecto % Donde "Proyecto" es el nombre de la carpeta que contiene
los archivos necesarios.
```

Paso 2) Ahora ejecutamos **probabilidad = Problem1(1000)** y tenemos:

```
octave:3> probabilidad = Problem1(1000)
probabilidad =  0.017000
```

Donde **probabilidad = Problem1(1000)** devuelve que la probabilidad estimada es 0.017000.

Problema 2.

Paso 1) Abrir Octave/MATLAB y entrar a la carpeta donde están guardados los archivos .m.

```
octave:1> cd Escritorio/
octave:2> cd Proyecto % Donde "Proyecto" es el nombre de la carpeta que contiene
los archivos necesarios.
```

Paso 2) Ahora ejecutamos [**p itmax**] = **brent(@(x)**$(log(x)-1), 2, 3, 10^{-8})$ y tenemos:

```
octave:3> [p itmax] = brent(@(x)(log(x)-1),2,3,10^(-8))
p =  2.71828170077059
itmax =  63
```

Donde [**p itmax**] = **brent(@(x)**$(log(x)-1), 2, 3, 10^{-8})$ devuelve los valores de p = 2.71828170077059 que es la aproximación y de itmax = 63 que es la cantidad de iteraciones necesarias para obtener la aproximación.

Problema 3.

Paso 1) Abrir Octave/MATLAB y entrar a la carpeta donde están guardados los archivos .m.

```
octave:1> cd Escritorio/
octave:2> cd Proyecto % Donde "Proyecto" es el nombre de la carpeta que contiene
los archivos necesarios.
```

Paso 2) Ahora ejecutamos **tabla** y tenemos:

```
octave:3> tabla
p =  2.56182684717907
N =  61
p =  2.56182659432894
```

```
N =  205
p =  2.00000000403226
N =  129
p =  1.00000004907711
N =  321
p =  0.666666664204711
N =  345
p = -0.444444442670871
N =  565
p =  2.76960736151804
N =  206105
p =  0.00737204543976672
N =  85
p =  1.01750521205773e-004
N =  57
p = -11.3047977948160
N =  49
```

Esto es sin tomar en cuenta la función $1/(3x-1)$ ya que el método no converge en el intervalo dado porque la función no es acotada en ese intervalo.

Donde p nos devuelve las aproximaciones para cada una de las funciones que están en la tabla y N es el número de iteraciones necesarias para alcanzar la tolerancia.

Además la tabla completa es la siguiente:

$f(x)$	a	b	tol	# function eval	aprox
$x^2e^{-x}+cosx$	-1	1	$1e-6$	61	2.56182684717907
$x^2e^{-x}+cosx$	-1	1	$1e-8$	205	2.56182659432894
$sinx$	0	π	$1e-8$	129	2.00000000403226
$cosx$	0	$(9/2)\pi$	$1e-6$	321	1.00000004907711
$\sqrt{x}$	0	1	$1e-8$	345	0.666666664204711
$\sqrt{x}logx$	eps	1	$1e-8$	565	-0.444444442670871
$tan(sinx)-sin(tanx)$	0	π	$1e-8$	206105	2.76960736151804
$1/(3x-1)$	0	1	$1e-4$	∞	No Converge
$x^{8/3}(1-x)^{10/3}$	0	1	$1e-8$	85	0.00737204543976672
$x^{25}(1-x)^2$	0	1	$1e-8$	57	1.01750521205773e-004
$\frac{1}{(x-3)^2+0{,}01}+\frac{1}{(x-9)^2+0{,}04}-6$	0	2	$1e-6$	49	-11.3047977948160

Problema 4.

Paso 1) Abrir Octave/MATLAB y entrar a la carpeta donde están guardados los archivos .m.

```
octave:1> cd Escritorio/
octave:2> cd Proyecto % Donde "Proyecto" es el nombre de la carpeta que contiene
los archivos necesarios.
```

Paso 2) Ahora ejecutamos **volumen = Problem4(100)** y tenemos:

```
octave:3> volumen = Problem4(100)
volumen = 0.0033039
```

Donde **volumen = Problem4(100)** devuelve el valor de volumen = 0.0033039 que es el volumen estimado del tetraedro.