Práctica 5: Integración

Jose Antonio Lorencio Abril

1 Ejercicios

1.1 Queremos obtener una tabla de valores para la función

$$\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-\frac{t^2}{2}}dt$$

(distribución normal de media cero y desviación típica uno) para los valores de x comprendidos entre 0 y 4 con incrementos de 0.1 utilizando la regla compuesta de Simpson

1.1.1 Calcula en primer lugar, mediante un programa Matlab, la integral

$$\frac{1}{2}+\frac{1}{\sqrt{2\pi}}\int_{0}^{1} e^{-\frac{t^2}{2}}dt$$

con un error menor que 10^{-6}.

El fichero simpson_fi.m queda adjunto terminado, tal como hicimos en clase. Cambiando que la función sea tratada como una cadena de caracteres, a que sea tratada como una función de Octave. También tenemos el fichero original, simpson.m, terminado, aunque este no lo voy a utilizar para resolver la práctica.

Ahora bien, debemos estimar el error, que, como vimos en clase de teoría

$$I - S_N = \frac{1}{N^4}\frac{(b-a)^5}{180}f^{5)}(\xi)$$

Sin embargo, también podemos pensar en el siguiente error:

$$I - S_{2N} = \frac{1}{16N^4}\frac{(b-a)^5}{180}f^{5)}(\xi)$$

Suponiendo que ambos ξ son el mismo, podemos pensar que

$$I - S_{2N} = E_{2N} \qquad I - S_N = 16E_{2N}$$

Restando ambas expresiones:

$$S_{2N} - S_N = 16E_{2N}$$

Por otro lado

$$\int_M^{\infty} e^{-\frac{x^2}{2}}dx < \int_M^{\infty} e^{-\frac{Mx}{2}}dx = \frac{2}{M}e^{-\frac{M^2}{2}}, \qquad si \quad m>1$$

Así, vamos a hacer un programa que encuentre M para obtener un error menor que 10^{-7}, para que, sumado al error de calculo de

$$\int_0^M e^{-\frac{x^2}{2}}dx$$

el error total sea menor que 10^{-6}, tal como buscamos.

```
M=2;
while (2/M*f(M)>1e-7)
  M=2*M;
end
```

Una vez conocemos M, vamos a calcular el número de puntos de interpolación que necesitaremos para que el error cometido sea menor que 10^{-6}.

```
n=2; %numero de particiones para simpson
S1=simpson_fi(f,0,M,n);
S2=simpson_fi(f,0,M,2*n);

while (abs(S2-S1)>1e-6)
  n=2*n;
  S1=S2;
  S2=simpson_fi(f,0,M,2*n);
end
```

Como vemos, lo que hacemos es ir doblando la cantidad de puntos usados, hasta que encontramos la cantidad que nos da un error menor que el requerido.

Por último, queda calcular la integral

$$\int_0^M e^{-\frac{x^2}{2}}dx$$

```
for (i=0:40)
  for(j=0:9)
    T(i+1,j+1)=0.5+simpson_fi(f,0,i*0.1+j*0.01,n);
  end
end
```

Y ya tenemos todos los datos necesarios para mostrar la tabla pedida en el enunciado.

También nos pide calcular

$$\frac{1}{2}+\frac{1}{\sqrt{2\pi}}\int_0^1 e^{-\frac{x^2}{2}}dx$$

Para ello, basta ver el valor almacenado en

$$T(11,1)=\frac{1}{2}+simpson\_fi(e^{-\frac{x^2}{2}},1,n)$$

Que es precisamente lo que queremos.

```
printf('El valor de la integral buscada es %d\n', T(11,1));
```

Que muestra:

```
El valor de la integral buscada es 0.841345
```

1.1.2 Utiliza las ideas del apartado anterior para construir la tabla deseada.

Ahora, la matriz T tiene todos los datos que necesitamos para contruir la tabla. Solo falta darle un formato 'amigable'.

Para ello, he creado un código que genera un archivo HTML en el que la tabla queda mucho más ordenada.

```
A="<html>\n\t<head>\n\t</head>\n\t<body>\n\t\t<table style=
        \"width:100%\">\n\t\t<tr  style=
            \"background-color: #DEB887\">\n\t\t\t<td></td>\n";
for  (j=0:9)
  A = [A "\t\t\t<td>" num2str(j*0.01)  "</td>\n"];
end
for  (i=0:40)
  A=[A "\t\t<tr>\n\t\t\t<td style=\"background-color:
          #DEB887\">" num2str(i*0.1)  "</td>\n"];
   for(j=0:9)
     A=[A "\t\t\t<td>" num2str(T(i+1,j+1))  "</td>\n"];
   end
  A = [A "\t\t</tr>\n"];
end
A = [A "\t\t</tr>\n"];
A = [A "\t</body>\n</html>"];
save  tabla_normal.html A
```

Como se puede apreciar, recorro T y voy generando la tabla HTML como una cadena de texto. Al final vuelco esta cadena en el archivo 'tabla_normal.html', adjunto a esta memoria. Clickando en él podemos ver la tabla generada.

1.2 Escribe programas en Matlab para las fórmulas de cuadratura de Gauss-Legendre, Gauss-Chebychev de tres y cuatro puntos (n=2, n=3) y utilizalos para aproximar las integrales

$$\int_{-1}^{1} \frac{sinx}{x} dx \qquad \int_{-1}^{1} \frac{e^x}{\sqrt{1-x^2}} dx, \quad \int_{0}^{2} \frac{1}{1+x^2} dx, \quad \int_{0}^{\frac{\pi}{4}} x^2 sinx dx, \quad \int_{1}^{1.5} x^2 logx dx$$

Compara los resultados obtenidos con los que saldrían si aplicas el método de los trapecios o el de Simpson. Utiliza los comandos de Matlab/Octave quad, quadl y trapz y compara los resultados. Mira en la ayuda de Matlab/Octave la información sobre estos comandos.

Para poder usar estos métodos lo primero que debemos hacer es llevar el intervalo $[a, b]$ en el que queremos calcular la integral al intervalo $[-1, 1]$.

Para ello usamos el cambio de variable

$$g(t) = a + \frac{b-a}{2}(1+t)$$

Una vez tenemos el cambio de variable, basta saber los nodos de interpolación y los pesos, y calcular

$$\sum \frac{1}{W(x_i)} w_i f(g(x_i))$$

Siendo w_i los pesos y x_i los correspondientes nodos. $W(x_i)$ es la función peso. En Gauss-Legendre es $W(x) = 1, \forall x$ y en Gauss-Chebyshev es $W(x) = \frac{1}{\sqrt{1-x^2}}$. Dividimos por la función peso puesto que

$$\int_{-1}^{1} f dx = \int_{-1}^{1} \frac{W}{W} f dx \approx \sum \frac{1}{W(x_i)} w_i f(x_i)$$

Estos métodos y notas quedan definidos en gauss_legendre.m y gauss_chebyshev.m.

- **Quad:** quad(f,a,b) evalúa numéricamente la integral de f entre a y b usando rutinas Fortran del paquete QUADPACK, basadas en cuadratura de Gauss.
- **QuadL:** quadl(f, a, b) evalúa numéricamente la integral de f entre a y b usando una adaptación de la regla de Lobatto. [Ver Anexo 1 (8)]
- **TrapZ:** trapz(x,y) evalúa numéricamente la integral de los puntos (x_i, y_i) usando el método del trapecio.

Para realizar este ejercicio, he programado el módulo comparar.m, que recibe una función y el inicio y final de un intervalo. Calcula la integral aproximada con Gauss-Legendre, con Gauss-Chebyshev para n=2 y n=3 y las compara con trapecios, simpson, quad, quadl y trapz.

Vamos a ver la salida obtenida para cada función.

a) $\frac{senx}{x}$:

```
Función: f =

@(x) (sin (x) + (x == 0)) ./ (x + (x == 0))

n       Gauss-Legendre  Gauss-Chebyshev
2       1.8921750e+00   1.9683174e+00

Con errores absolutos:
|GL-Trapecios|  |GL-Simpson|    |GC-Trapecios|  |GC-Simpson|
8.9170017e-05   5.6980670e-06   7.6231570e-02   7.6148098e-02

Comparamos con quad, quadl, trapz:
|GL-quad|       |GL-quadl|      |GL-trapz|      |GC-quad|       |GC-quadl|      |GC-trapz|
8.8571094e-06   8.8571094e-06   9.5556290e-01   7.6151257e-02   7.6151257e-02   1.0317053e+00


Función: f =

@(x) (sin (x) + (x == 0)) ./ (x + (x == 0))

n       Gauss-Legendre  Gauss-Chebyshev
3       1.8921661e+00   1.9352435e+00

Con errores absolutos:
|GL-Trapecios|  |GL-Simpson|    |GC-Trapecios|  |GC-Simpson|
8.0281553e-05   3.1903966e-06   4.3157650e-02   4.3074178e-02

Comparamos con quad, quadl, trapz:
|GL-quad|       |GL-quadl|      |GL-trapz|      |GC-quad|       |GC-quadl|      |GC-trapz|
3.1354152e-08   3.1354153e-08   9.5555401e-01   4.3077337e-02   4.3077337e-02   9.9863138e-01
```

Como podemos observar, con Gauss-Legendre obtenemos menores errores que con Gauss-Chebyshev, además, los errores son mejores con n=3.

b) $\frac{e^x}{\sqrt{1-x^2}}$: en esta función apreciamos un claro problema en los extremos del intervalo. Si alguno de los métodos de aproximación de integral usa estos puntos para evaluar la función, obtendremos una división por 0. Trabajando al respecto parece ser que trapz, quad y quadl hacen uso de estos puntos para la aproximación. Sabemos también, de antemano, que nuestros métodos de los trapecios y de Simpson los usan también. Por tanto, no obtendremos buenas aproximaciones con estas funciones. Esto, efectivamente lo comprobamos en la salida del programa:

```
Función: f =

@(x) e .^ x ./ (1 - x .^ 2)

n       Gauss-Legendre  Gauss-Chebyshev
2       4.5425090e+00   6.9074464e+00

Con errores absolutos:
|GL-Trapecios|  |GL-Simpson|    |GC-Trapecios|  |GC-Simpson|
Inf      Inf      Inf      Inf

Comparamos con quad, quadl, trapz:
|GL-quad|       |GL-quadl|      |GL-trapz|      |GC-quad|       |GC-quadl|      |GC-trapz|
Inf      Inf      Inf      Inf      Inf      Inf


Función: f =

@(x) e .^ x ./ (1 - x .^ 2)

n       Gauss-Legendre  Gauss-Chebyshev
3       5.3140481e+00   7.8109173e+00

Con errores absolutos:
|GL-Trapecios|  |GL-Simpson|    |GC-Trapecios|  |GC-Simpson|
Inf      Inf      Inf      Inf

Comparamos con quad, quadl, trapz:
|GL-quad|       |GL-quadl|      |GL-trapz|      |GC-quad|       |GC-quadl|      |GC-trapz|
Inf      Inf      Inf      Inf      Inf      Inf
```

Vemos que dan infinito todos los errores relativos, dado que las integrales calculadas con los métodos ya mencionados dan Inf. Voy a intentar arreglar un poco esto tomando el intervalo $[a + 0.00001, b - 0.00001]$.

```
Función: f =

@(x) e .^ x ./ (1 - x .^ 2)

n       Gauss-Legendre  Gauss-Chebyshev
2       4.5423356e+00   6.9069902e+00

Con errores absolutos:
|GL-Trapecios|  |GL-Simpson|    |GC-Trapecios|  |GC-Simpson|
3.0873992e+03   1.0286244e+04   3.0850345e+03   1.0283879e+04

Comparamos con quad, quadl, trapz:
|GL-quad|       |GL-quadl|      |GL-trapz|      |GC-quad|       |GC-quadl|      |GC-trapz|
1.3177168e+01   1.3177168e+01   7.7039600e+02   1.0812514e+01   1.0812514e+01   7.6803135e+02


Función: f =

@(x) e .^ x ./ (1 - x .^ 2)

n       Gauss-Legendre  Gauss-Chebyshev
3       5.3137513e+00   7.8100927e+00

Con errores absolutos:
|GL-Trapecios|  |GL-Simpson|    |GC-Trapecios|  |GC-Simpson|
3.0866278e+03   1.0285472e+04   3.0841314e+03   1.0282976e+04

Comparamos con quad, quadl, trapz:
|GL-quad|       |GL-quadl|      |GL-trapz|      |GC-quad|       |GC-quadl|      |GC-trapz|
1.2405753e+01   1.2405753e+01   7.6962459e+02   9.9094112e+00   9.9094112e+00   7.6712825e+02
```

Esto está bastante mejor. Podemos observar como las integrales calculadas por Gauss-Legendre y por Gauss-Chebyshev apenas varían, pero ahora podemos compararlas con las obtenidas por los otros métodos.

Vemos que las diferencias son bastante significativas, lo que se debe al carácter de la función.

Poco podemos afirmar, pero lo que está claro es que difícilmente habremos obtenido una buena aproximación. Un indicador de esto es que los propios valores arrojados por nuestros dos métodos distan bastante.

c) $\frac{1}{1+x^2}$:

```
Función: f =

@(x) 1 ./ (1 + x .^ 2)

n       Gauss-Legendre  Gauss-Chebyshev
2       1.1070336e+00   1.1547863e+00

Con errores absolutos:
|GL-Trapecios|  |GL-Simpson|    |GC-Trapecios|  |GC-Simpson|
9.3746136e-05   1.1288783e-04   4.7658955e-02   4.7639814e-02

Comparamos con quad, quadl, trapz:
|GL-quad|       |GL-quadl|      |GL-trapz|      |GC-quad|       |GC-quadl|      |GC-trapz|
1.1507865e-04   1.1507865e-04   5.5899772e-01   4.7637623e-02   4.7637623e-02   6.0675042e-01


Función: f =

@(x) 1 ./ (1 + x .^ 2)

n       Gauss-Legendre  Gauss-Chebyshev
3       1.1067400e+00   1.1373502e+00

Con errores absolutos:
|GL-Trapecios|  |GL-Simpson|    |GC-Trapecios|  |GC-Simpson|
3.8740286e-04   4.0654456e-04   3.0222817e-02   3.0203675e-02

Comparamos con quad, quadl, trapz:
|GL-quad|       |GL-quadl|      |GL-trapz|      |GC-quad|       |GC-quadl|      |GC-trapz|
4.0873538e-04   4.0873538e-04   5.5870406e-01   3.0201484e-02   3.0201484e-02   5.8931428e-01
```

En esta ocasión, al igual que en la primera, vemos que Gauss-Legendre parece aproximar la integral mejor que Gauss-Chebyshev. Sin embargo, ahora producimos un mejor resultado con n=2 que con n=3.

d) $x^2 senx$:

```
Función: f =

@(x) x .^ 2 .* sin (x)

n       Gauss-Legendre  Gauss-Chebyshev
2       8.8753854e-02   9.8157601e-02

Con errores absolutos:
|GL-Trapecios|  |GL-Simpson|    |GC-Trapecios|  |GC-Simpson|
3.3238136e-05   2.6089140e-07   9.3705091e-03   9.4034863e-03

Comparamos con quad, quadl, trapz:
|GL-quad|       |GL-quadl|      |GL-trapz|      |GC-quad|       |GC-quadl|      |GC-trapz|
1.4308174e-06   1.4308174e-06   2.3133044e-02   9.4023164e-03   9.4023164e-03   1.3729296e-02


Función: f =

@(x) x .^ 2 .* sin (x)

n       Gauss-Legendre  Gauss-Chebyshev
3       8.8755286e-02   9.3622440e-02

Con errores absolutos:
|GL-Trapecios|  |GL-Simpson|    |GC-Trapecios|  |GC-Simpson|
3.1805591e-05   1.1716538e-06   4.8353483e-03   4.8683256e-03

Comparamos con quad, quadl, trapz:
|GL-quad|       |GL-quadl|      |GL-trapz|      |GC-quad|       |GC-quadl|      |GC-trapz|
1.7278448e-09   1.7278448e-09   2.3131611e-02   4.8671556e-03   4.8671556e-03   1.8264457e-02
```

De nuevo, Gauss-Legendre aproxima mejor que Gauss-Chebyshev.

En esta ocasión sí apreciamos una mejora muy significativa en la aproximación con el aumento de n.

e)$x^2 log x$:

```
Función: f =

@(x) x .^ 2 .* log (x)

n       Gauss-Legendre  Gauss-Chebyshev
2       1.9225938e-01   2.0367418e-01

Con errores absolutos:
|GL-Trapecios|  |GL-Simpson|    |GC-Trapecios|  |GC-Simpson|
1.4283780e-05   4.2643255e-08   1.1400517e-02   1.1414844e-02

Comparamos con quad, quadl, trapz:
|GL-quad|       |GL-quadl|      |GL-trapz|      |GC-quad|       |GC-quadl|      |GC-trapz|
1.9524083e-08   1.9524083e-08   1.8842137e-01   1.1414820e-02   1.1414820e-02   1.7700657e-01


Función: f =

@(x) x .^ 2 .* log (x)

n       Gauss-Legendre  Gauss-Chebyshev
3       1.9225936e-01   1.9840542e-01

Con errores absolutos:
|GL-Trapecios|  |GL-Simpson|    |GC-Trapecios|  |GC-Simpson|
1.4303232e-05   2.3191239e-08   6.1317608e-03   6.1460873e-03

Comparamos con quad, quadl, trapz:
|GL-quad|       |GL-quadl|      |GL-trapz|      |GC-quad|       |GC-quadl|      |GC-trapz|
7.2067158e-11   7.2067130e-11   1.8842139e-01   6.1460642e-03   6.1460642e-03   1.8227533e-01
```

Como era de esperar, Gauss-Legendre aproxima mucho mejor que Gauss-Chebyshev. También observamos una mejoría con n=3.

Así, comprobamos como diferentes métodos de integración numérica producen distintos resultados y tienen distintas debilidades. Por ejemplo, en **b)**, cuando hemos reducido un poco el intervalo para poder usar algunos métodos, es evidente que estamos perdiendo información sobre la integral. A cambio, ganamos en variedad de métodos utilizables. A la hora de hacer decisiones de este tipo, debemos tener en cuenta cuánta información perdemos, y a cambio de qué lo estamos haciendo. Este cambio puede que merezca la pena si nos permite usar métodos mucho más precisos de integración numérica o si el valor de la integral apenas varía eliminando esa porción del intervalo.

También es importante fijarnos a qué función nos enfrentamos y qué método funcionará mejor para integrarla. Para ello disponemos de las estimaciones de las diferencias por medio de las derivadas sucesivas.

1.3 Anexo 1: Fórmulas de Lobatto

Para cada $N \geq 3$, es la que se obtiene al integrar el polinomio interpolador de grado N basado en los nodos $x_0 = a < x_1 < ... < x_N = b$, donde los nodos $x_1, ..., x_N$ se eligen para que el polinomio

$$W(x) = (x-a)(x-x_1)...(x-x_{N-1})(x-b)$$

verifique que

$$\int_a^b W(x)(x - \frac{a+b}{2})^k dx = 0, \quad k = 0, 1, ..., N-2$$