



Universidad
Industrial de
Santander

Informe Laboratorio: Análisis Numérico

Práctica No. 8

Daniel Delgado

Código: 2182066

Grupo: B2

Escuela de Ingeniería de Sistemas e Informática
Universidad Industrial de Santander

25 de febrero de 2021

1. Introducción

La aproximación de las integrales definidas es una de las partes fundamentales en la solución de diferentes problemas de la vida real. Esto va en diferentes áreas que van desde la estadística hasta el cálculo de diferentes sólidos. En este sentido, el entendimiento de estos conceptos es realmente importantes para el desarrollo de la identidad matemática.

La comprensión de las diferentes maneras de aproximar integrales, al igual que el desarrollo de la algoritmia relacionada, son los principales temas a tratar durante el desarrollo del presente informe, así como la resolución de los problemas propuestos a manera de pregunta orientadora del componente práctico del mismo.

2. Desarrollo

2.1. Aplicando

a) Integrar $f(x) = x \ln(x)$ entre $[1, 2]$ con trapezoides y $h = 0.25$.

$$\int_a^b f(x) \approx \sum_{k=1}^n \frac{f(x_{k-1}) + f(x_k)}{2} \Delta x$$

$$x_0 = 1$$

$$x_1 = 1.25$$

$$x_2 = 1.5$$

$$x_3 = 1.75$$

$$x_4 = 2$$

$$\int_1^2 x \ln(x) \approx \sum_{k=1}^4 \frac{f(x_{k-1}) + f(x_k)}{2} 0.25$$

$$\int_1^2 x \ln(x) \approx \frac{0.25}{2} [f(x_0) + 2f(x_1) + 2f(x_2) + 2f(x_3) + f(x_4)]$$

$$\int_1^2 x \ln(x) \approx 0.125 [1 \ln(1) + 2(1.25) \ln(1.25) + 2(1.5) \ln(1.5) + 2(1.75) \ln(1.75) + (2) \ln(2)]$$

$$\int_1^2 x \ln(x) \approx 0.125 [2.5 \ln(1.25) + 3 \ln(1.5) + 3.5 \ln(1.75) + 2 \ln(2)]$$

$$\int_1^2 x \ln(x) \approx 0.639900477688$$

- b) Integrar $f(x) = x \ln(x)$ entre $[1, 2]$ con la regla de Simpson y $h = 0.25$.

$$\begin{aligned}\int_a^b x \ln(x) &\approx \frac{b-a}{6} \left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right) \\ \int_1^2 x \ln(x) &\approx \frac{2-1}{6} \left(f(1) + 4f\left(\frac{1+2}{2}\right) + f(2) \right) \\ \int_1^2 x \ln(x) &\approx \frac{1}{6} \left((1) \ln(1) + 4 \left(\frac{3}{2}\right) \ln\left(\frac{3}{2}\right) + 2 \ln(2) \right) \\ \int_1^2 x \ln(x) &\approx 0.636514168295\end{aligned}$$

2.2. Implementando

Con el fin de cumplir los objetivos de la práctica del laboratorio, se pide el desarrollo de la algoritmia relacionada con la aplicación de las reglas del trapecio compuesta y la regla de Simpson compuesta.

- a) En el caso de la regla del trapecio compuesta, se desarrolló `compositeTrapezoid(f,M,a,b)`. Esta función, realizaba la aproximación de la integral para una función f dada en el intervalo $[a,b]$.

```

1 function output = compositeTrapezoid(f, M, a, b)
2     if (isa(f, 'function_handle') && (M > 0))
3         h = (b-a)/M;
4         sumPart = 0;
5         xk = a;
6
7         for index = 1:M
8             sumPart = sumPart + f(xk);
9             xk = xk + h;
10        end
11
12        output = (h/2)*(f(a)+f(b))+h*sumPart;
13    end
14 end

```

De manera inicial, realiza una verificación de los parámetros dados además de realizar el cálculo de $h = \frac{b-a}{M}$ e inicializa las variables `xk` y `sumPart`. Finalmente, ingresa a un ciclo iterativo el cual realiza la sumatoria correspondiente y da salida a la función con la aproximación tras la suma de las partes calculadas.

Con el fin de comprobar el correcto funcionamiento de esta, y aprovechando las aproximaciones realizadas en la sección 2.1, se ejecutó la función en una terminal de MatLab con los mismos parámetros:

```

1 >> compositeTrapezoid(@(x) x*log(x), 5, 1, 2)
2
3 ans =
4
5     0.638603196719876

```

De esto, como es posible apreciar, da como resultado una aproximación cercana a la integral en el intervalo $[1,2]$ para la función $f(x) = x \ln(x)$.

- b) Un proceso similar se realizó para la regla de Simpson compuesta. Con el fin de dar cabida a los requerimientos, se desarrolló la función `compositeSimpson(f, M, a, b)`, la cual, como su nombre lo indica, realiza la aproximación de una integral para una función f dada en el intervalo $[a,b]$ empleando la regla de Simpson compuesta.

```

1 function output = compositeSimpson(f, M, a, b)
2     if (isa(f, 'function_handle') && (M > 0))
3         sumPart = 0;
4         sumPart2 = 0;
5         h = (b-a)/(2*M);
6
7         for k = 1:(M-1)
8             sumPart = sumPart + f(a+2*k*h);
9         end
10
11        for k = 1:M
12            sumPart2 = sumPart2 + f(a+(2*k-1)*h);
13        end
14
15        output = (h/3) * (f(a) + f(b) + 2*sumPart + 4*sumPart2);
16    end
17 end

```

Esta función realiza un proceso similar a la anterior función, inicialmente realiza las comprobaciones de los parámetros datos, seguidamente inicializa, en ese caso, las sentencias for para realizar los cálculos de las sumatorias respectivas, y finalmente, construye la aproximación para la integral.

Con el fin de comprobar el funcionamiento de la función, se repetirá el mismo proceso para ver que se esté realizando la aproximación de manera correcta. En este sentido, se ejecutó la función en una terminal de MatLab.

```

1 >> compositeSimpson(@(x) x*log(x), 5, 1, 2)
2
3 ans =
4
5     0.636294774145023

```

De esto, como es posible apreciar, da como resultado una aproximación cercana a la integral en el intervalo $[1,2]$ para la función $f(x) = x \ln(x)$.

2.3. Interpretando

Con el fin de demostrar las aplicaciones que tienen las reglas del trapecio y Simpson, se nos presenta un problema en el cual debemos calcular la probabilidad de que una máquina falle. En este caso, se nos dice que la probabilidad de que la máquina falle está dada por $\Phi(x) = \frac{1}{2} + (1/\sqrt{2\pi}) \int_0^x e^{-t^2/2}$.

Entonces, se nos pregunta que para dicha máquina, ¿cuál es la probabilidad de que esta falle un total de 5 veces?

Para resolver esto, empleando las funciones creadas en la sección 2.2, podemos ejecutar las funciones de la siguiente manera:

```

1 >> compositeSimpson(@(x) exp(1)^(-(x^2)/2), 5, 0, 5)
2
3 ans =
4
5     1.253313328468611
6
7 >> compositeTrapezoid(@(x) exp(1)^(-(x^2)/2), 5, 0, 5)
8
9 ans =
10
11     2.253312265441977

```

Como es posible de denotar, existe una gran diferencia entre los valores calculados por cada una de las funciones para una misma aproximación. Esto se debe al como se desarrolla el cálculo de la integral en cada una de las funciones en cuanto se refiere a la forma de calcular la aproximación.

Cabe resaltar que, entre las dos funciones, para $M = 5$, la Simpson compuesta es la que tiene una mejor aproximación real a la integral, sin embargo, en el caso de aumentar drásticamente M , digamos $M = 12000000$, la aproximación realizada por la función es mucho mejor en comparación.

Ahora, para calcular la probabilidad que se de un fallo en la máquina, simplemente tenemos que remplazar en la ecuación dada.

$$\Phi(x) = \frac{1}{2} + \frac{1}{\sqrt{2\pi}} \cdot \int_0^x e^{-\frac{t^2}{2}} dt$$

$$\Phi_{Simpson}(x) = \frac{1}{2} + \frac{1.253313328468611}{\sqrt{2\pi}} = 0.999999677317$$

$$\Phi_{Trapezio}(x) = \frac{1}{2} + \frac{2.253312265441977}{\sqrt{2\pi}} = 1.39894153363$$

De esto, partiendo de lo ya nombrado respecto a la función `compositeTrapezoid(f,M,a,b)`, lo que nos dejaría sólo con el valor calculado por la regla de Simpson compuesta, podemos decir que para la máquina, existe una probabilidad del 0.999999677317 de que tenga un total de 5 fallos.

3. Anexos

`compositeTrapezoid.m`

```

1 function output = compositeTrapezoid(f, M, a, b)
2     if (isa(f, 'function_handle') && (M > 0))
3         h = (b-a)/M;
4         sumPart = 0;
5         xk = a;
6
7         for index = 1:M
8             sumPart = sumPart + f(xk);
9             xk = xk + h;
10        end
11
12        output = (h/2)*(f(a)+f(b))+h*sumPart;
13    end
14 end

```

`compositeSimpson.m`

```

1 function output = compositeSimpson(f, M, a, b)
2     if (isa(f, 'function_handle') && (M > 0))
3         sumPart = 0;
4         sumPart2 = 0;
5         h = (b-a)/(2*M);
6
7         for k = 1:(M-1)
8             sumPart = sumPart + f(a+2*k*h);
9         end
10
11        for k = 1:M
12            sumPart2 = sumPart2 + f(a+(2*k-1)*h);
13        end
14
15        output = (h/3) * (f(a) + f(b) + 2*sumPart + 4*sumPart2);
16    end

```

17 `end``trapezoidTester`

```
1 >> compositeTrapezoid(@(x) x*log(x), 5, 1, 2)
2
3 ans =
4
5 0.638603196719876
```

`simponTester`

```
1 >> compositeSimpson(@(x) x*log(x), 5, 1, 2)
2
3 ans =
4
5 0.636294774145023
```

`machineProbabilty`

```
1 >> compositeSimpson(@(x) exp(1)^(-(x^2)/2), 5, 0, 5)
2
3 ans =
4
5 1.253313328468611
6
7 >> compositeTrapezoid(@(x) exp(1)^(-(x^2)/2), 5, 0, 5)
8
9 ans =
10
11 2.253312265441977
```