

Tarea 11



Nombre:	Jairo Saul Diaz Soto
Maestría:	Ciencias Computacionales
Modulo:	Métodos Numéricos
Instructor:	Dr. Luis Daniel Blanco Cocom
Fecha de entrega:	2023 - 11 - 05

1. Mínimos Cuadrados

En muchas ocasiones, se tienen conjuntos de datos los cuales o bien no son ideales, contienen ruido o algún tipo de error, sin embargo, se quiere saber si este conjunto de datos, pueden representar a algún modelo matemático.

Entonces, aparece el método de mínimos cuadrados, el cual aproxima un polinomio a un conjunto de datos, minimizando el error entre la curva y cada par ordenado del conjunto de datos.

De esta forma es posible obtener una curva que, aunque no necesariamente pase exactamente por el conjunto de datos, genera una curva que aproxima la información, pudiendo contrastar esta con algún modelo.

Para lograr esto, se propone la siguiente desigualdad que debe ser satisfecha

$$\sum_{i=0}^n [y_i - \hat{f}(x_i)]^2 \leq \sum_{i=0}^n [y_i - P_m(x_i)]^2$$

para todo $\hat{f} \in \mathcal{P}_m$ con $P_m \in \mathcal{P}_m$ donde x_i, y_i los pares ordenados del conjunto de datos.

De manera general se propone la regresión por minimos cuadrados \hat{f} como

$$\hat{f}(x) = \sum_i a_i \phi_i(x)$$

A continuación un algoritmo que implementa lo anterior.

Algorithm 1: Algoritmo de Mínimos Cuadrados con Regularización

Data: Arreglos de datos x y y , valor de regularización λ , tamaño del arreglo sz , índice k , grado del polinomio n , función $function(x, j)$

Result: Arreglo de coeficientes $coef$

- (1) $B \leftarrow \text{GenerarMatriz}(sz, n);$
 - (2) **for** $i \leftarrow 1$ **to** sz **do**
 - (3) **for** $j \leftarrow 1$ **to** n **do**
 - (4) **if** $k < 0$ **then**
 - (5) $B[i][j] \leftarrow function(x[i], j);$
 - (6) **else**
 - (7) $B[i][j] \leftarrow function(x[i] - x[k], j);$
 - (8) $BT \leftarrow \text{TransponerMatriz}(B, sz, n);$
 - (9) $A \leftarrow \text{MultiplicarMatrices}(BT, B, n, sz, n);$
 - (10) **for** $i \leftarrow 1$ **to** n **do**
 - (11) $A[i][i] \leftarrow A[i][i] + \lambda;$
 - (12) $b \leftarrow \text{MultiplicarMatrizPorVector}(BT, y, n, sz);$
 - (13) $qr \leftarrow \text{FactorizacionQR}(A, n);$
 - (14) $qt \leftarrow \text{TransponerMatriz}(qr.Left, n, n);$
 - (15) $b1 \leftarrow \text{MultiplicarMatrizPorVector}(qt, b, n, n);$
 - (16) $coef \leftarrow \text{ResolverSistema}(qr.Right, b1, n);$
 - (17) **return** $coef;$
-

A continuación se muestran algunos resultados obtenidos donde se prueba el método

Evaluacion del punto t=1.200000 es suth(t)=0.025954			
funcion	lambda	prediccion	error abs
polinomial	0.0000000	0.024409	0.001545
polinomial	0.0000100	0.024922	0.001032
polinomial	0.0000001	0.024415	0.001539
trigonometrica	0.0000000	0.030671	0.004716
trigonometrica	0.0000100	0.029826	0.003872
trigonometrica	0.0000001	0.029770	0.003816
exponencial	0.0000000	0.030752	0.004798
exponencial	0.0000100	0.029047	0.003092
exponencial	0.0000001	0.030913	0.004959

Figura 1: Evaluación de la función de Sutherland para la viscosidad del Oxígeno (Primer colección de puntos, grado del polinomio 3) y distintos valores de regularización ($\lambda = 0, 1e-5, 1e-7$) para regresión polinómica, trigonométrica y exponencial.

Evaluacion del punto t=1.200000 es suth(t)=0.025954			
funcion	lambda	prediccion	error abs
polinomial	0.0000000	0.025865	0.000089
polinomial	0.0000100	0.025870	0.000085
polinomial	0.0000001	0.025865	0.000089
trigonometrica	0.0000000	0.025801	0.000154
trigonometrica	0.0000100	0.026568	0.000613
trigonometrica	0.0000001	0.026144	0.000189
exponencial	0.0000000	0.026118	0.000164
exponencial	0.0000100	0.026954	0.001000
exponencial	0.0000001	0.026180	0.000226

Figura 2: Evaluación de la función de Sutherland para la viscosidad del Oxígeno (Segunda colección de puntos, grado del polinomio 3) y distintos valores de regularización ($\lambda = 0, 1e-5, 1e-7$) para regresión polinómica, trigonométrica y exponencial.

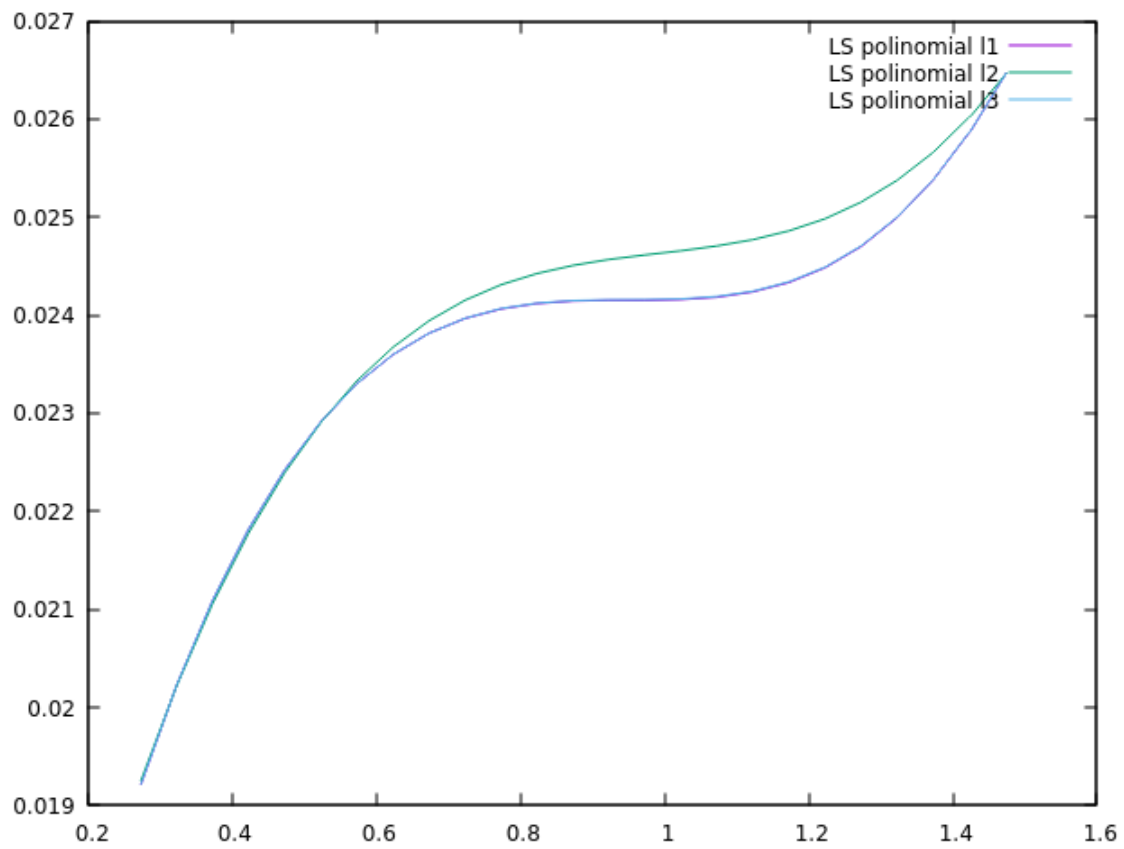


Figura 3: Evaluación de la función de Sutherland con una regresión exponencial para la viscosidad del Oxígeno en un intervalo dado (Primer colección de puntos, grado del polinomio 3) y distintos valores de regularización ($\lambda = 0, 1e-5, 1e-7$)

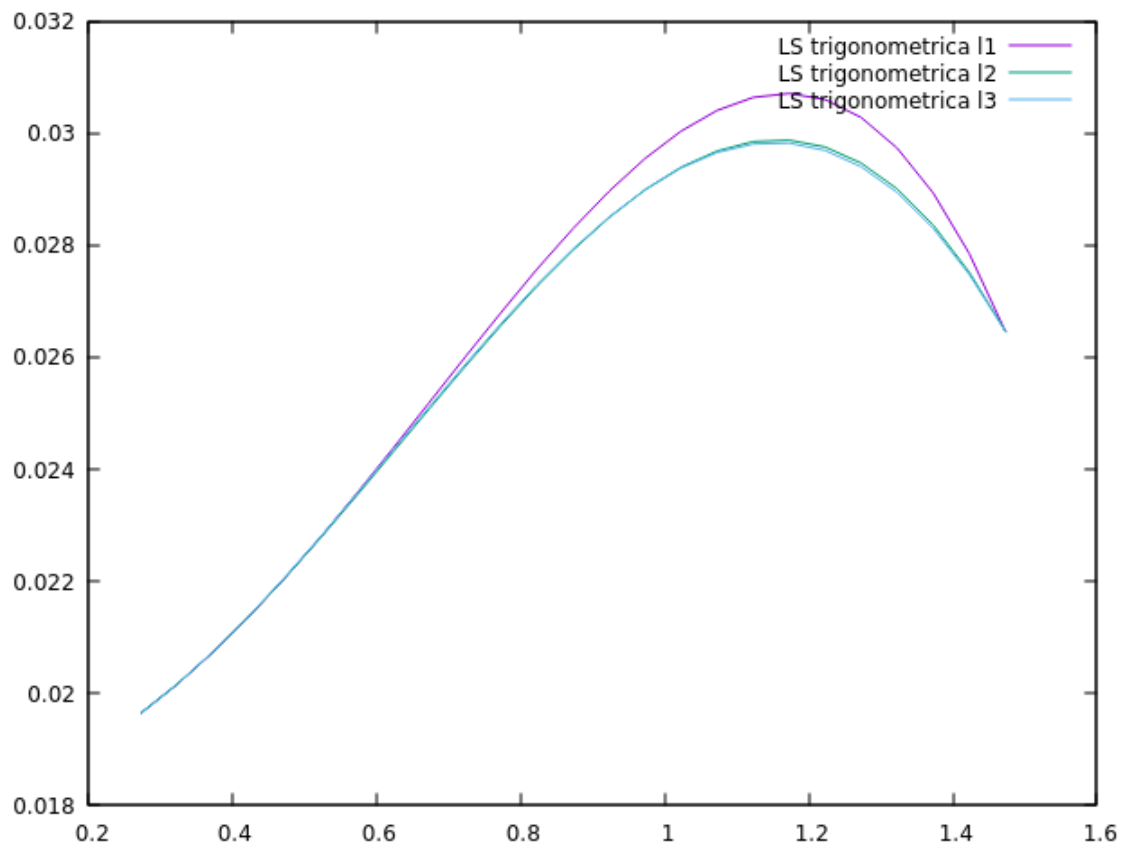


Figura 4: Evaluación de la función de Sutherland con una regresión exponencial para la viscosidad del Oxígeno en un intervalo dado (Primer colección de puntos, grado del polinomio 3) y distintos valores de regularización ($\lambda = 0, 1e-5, 1e-7$)

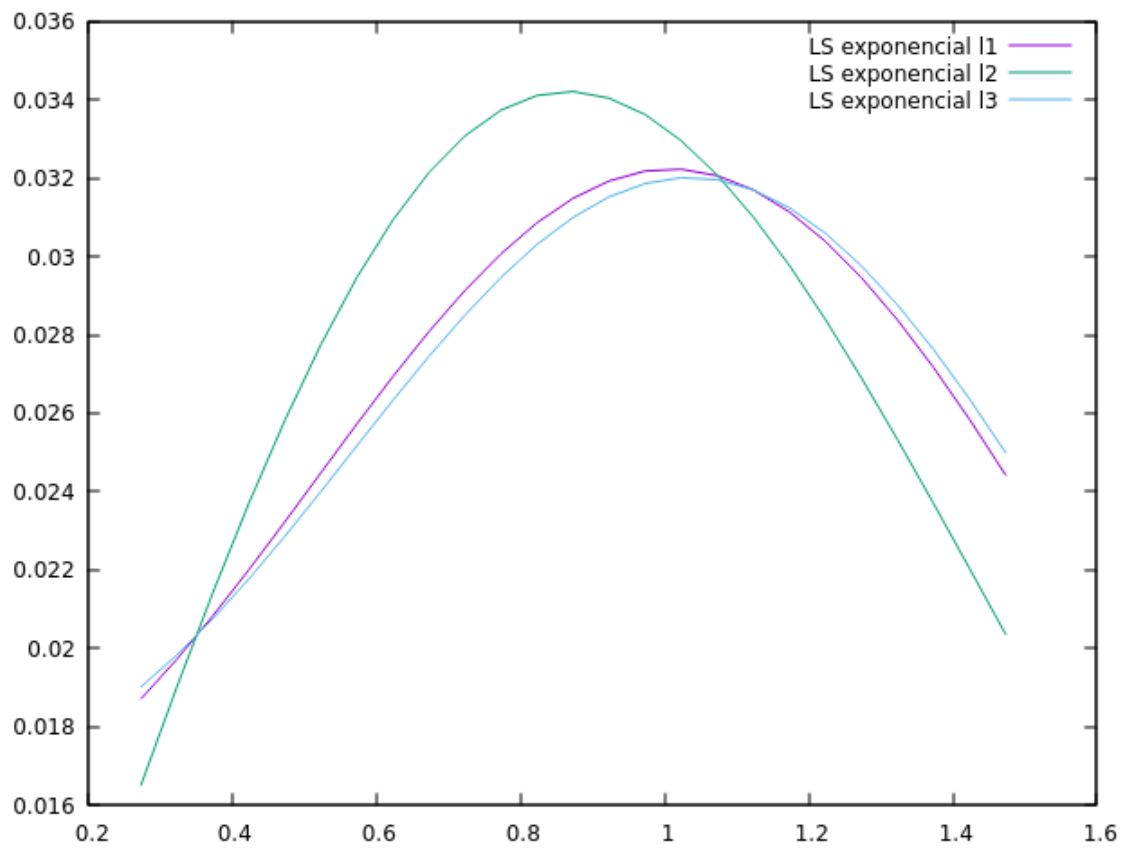


Figura 5: Evaluación de la función de Sutherland con una regresión exponencial para la viscosidad del Oxígeno en un intervalo dado (Primer colección de puntos, grado del polinomio 3) y distintos valores de regularización ($\lambda = 0, 1e-5, 1e-7$)

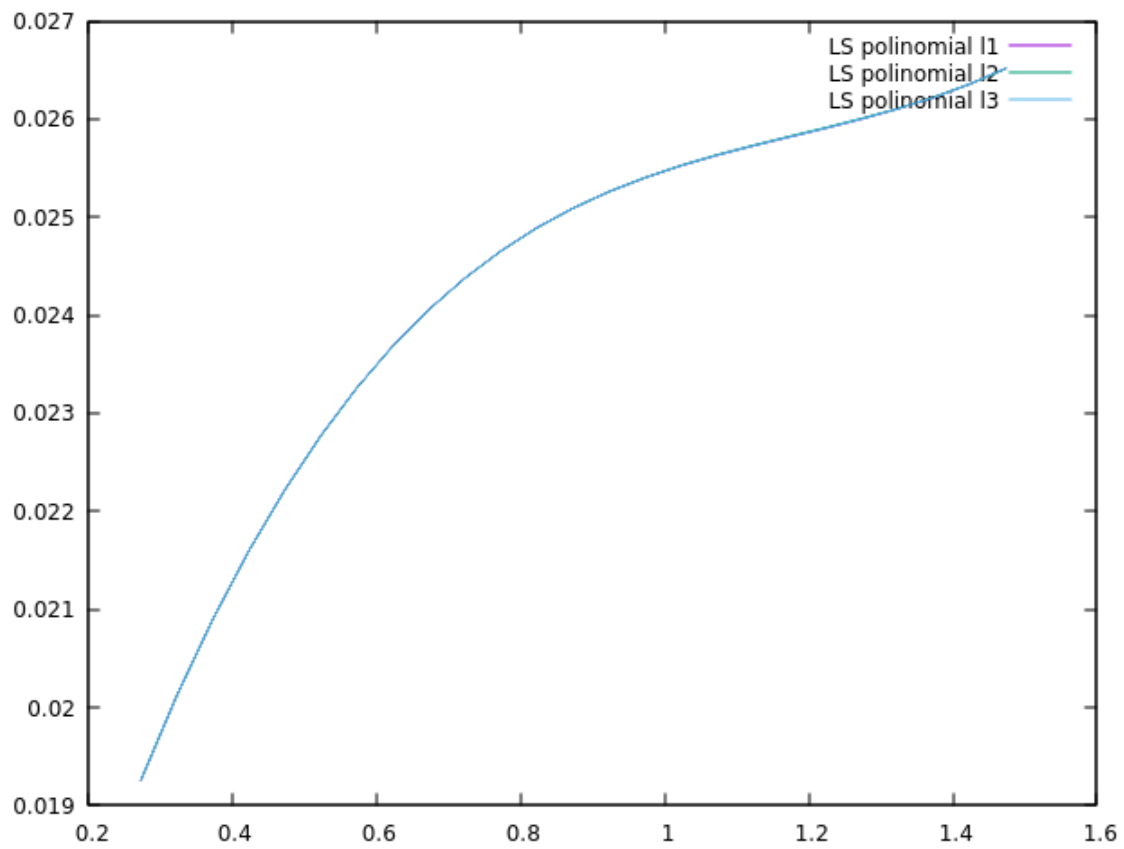


Figura 6: Evaluación de la función de Sutherland con una regresión exponencial para la viscosidad del Oxígeno en un intervalo dado (Segunda colección de puntos, grado del polinomio 3) y distintos valores de regularización ($\lambda = 0, 1e - 5, 1e - 7$)

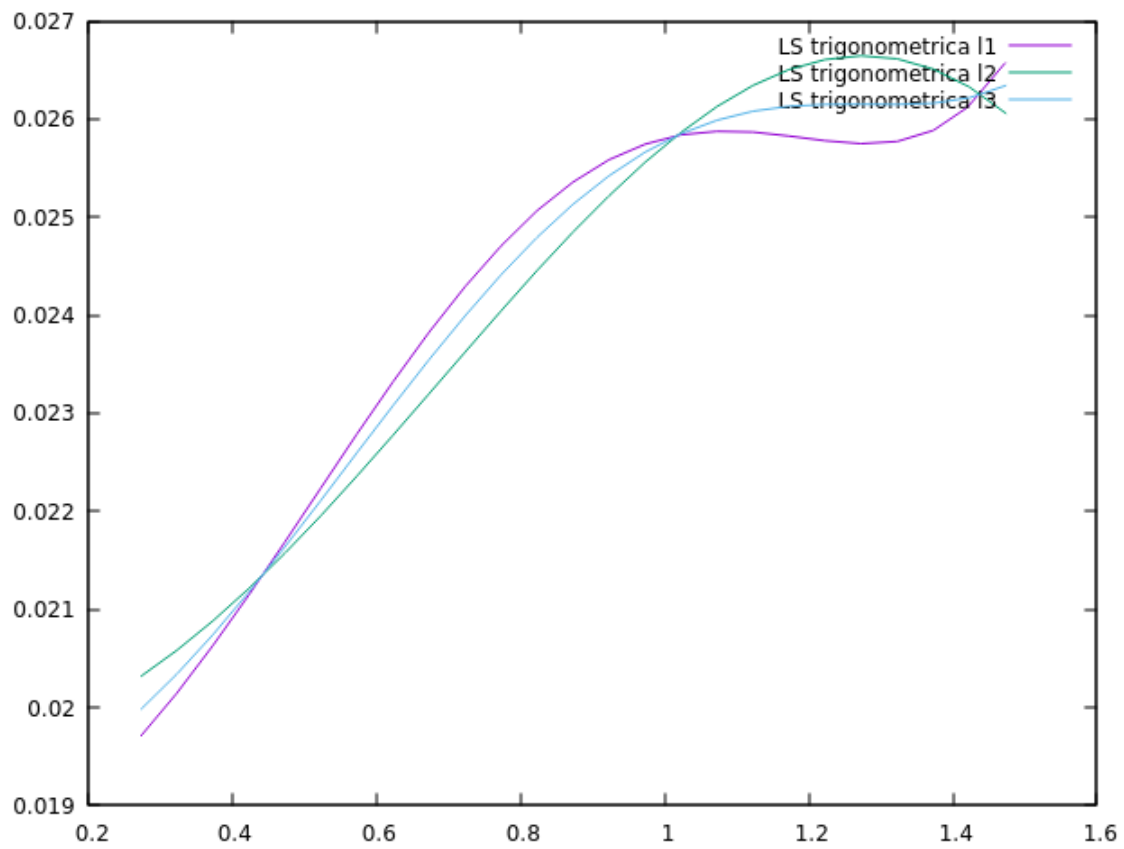


Figura 7: Evaluación de la función de Sutherland con una regresión exponencial para la viscosidad del Oxígeno en un intervalo dado (Segunda colección de puntos, grado del polinomio 3) y distintos valores de regularización ($\lambda = 0, 1e-5, 1e-7$)

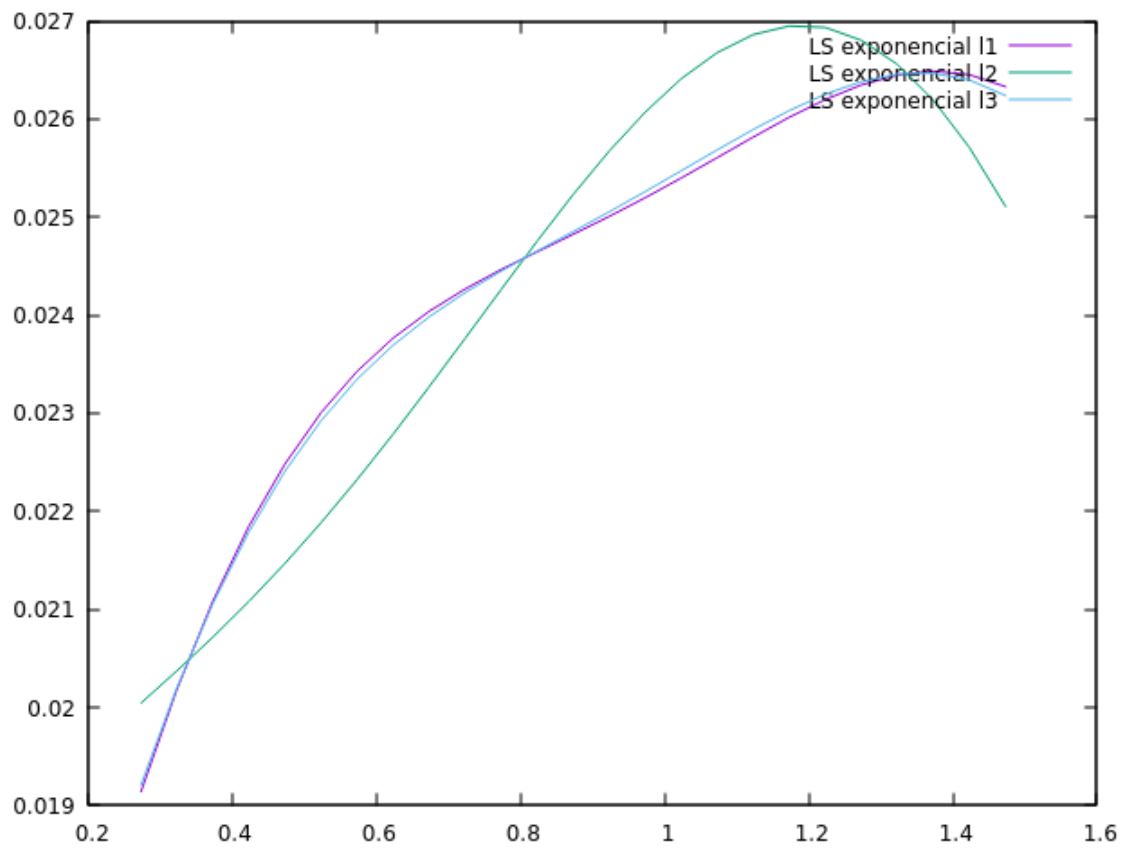


Figura 8: Evaluación de la función de Sutherland con una regresión exponencial para la viscosidad del Oxígeno en un intervalo dado (Segunda colección de puntos, grado del polinomio 3) y distintos valores de regularización ($\lambda = 0, 1e-5, 1e-7$)

De lo anterior, es posible observar, que para los casos cuando el valor de regularización es 0 en contraste con el valor de $1e - 7$ en todas las gráficas prácticamente son sumamente similares, e inclusive en la evaluación de un solo punto podemos ver que el cambio es minúsculo, sin embrago, en contraste contra el tercer valor prevalece en la mayoría de los casos la forma de la curva, sin embargo, en algunos lugares está es más pronunciada que con valores absolutos más pequeños de regularización.

En algunos casos, el valor de regularización logra un cambio de rotación de la curva, para algunos segmentos inclusive cambia de cóncavo a convexo y viceversa.

2. Integración numérica

En la práctica, se suelen necesitar la realización de integrales definidas para funciones sumamente complejas. Es por ello, que es necesaria la implementación de métodos numéricos los cuales ayuden a obtener aproximaciones de estas integrales sin la necesidad de resolver estas de forma directa.

Existen distintos métodos por los cuales estas se obtienen, algunos de los métodos más populares por su simplicidad y precisión son los métodos denominados Newton - Cotes, de los cuales, existen los abiertos y cerrados.

De los métodos abiertos, según el número de puntos que utilizan para realizar la aproximación (Lo cual afecta directamente en la precisión del método), se presentan los algoritmos para los métodos de grado 0, 1, 2, 3 a continuación

Algorithm 2: Algoritmo para NCO0

Data: Función $function(x)$, límites a y b

Result: Resultado de la integral utilizando NCO0

```

1  $h \leftarrow (b - a)/2;$ 
2  $x0 \leftarrow a + h;$ 
3  $integral \leftarrow 2 \cdot h \cdot function(x0);$ 
4 return  $integral;$ 
```

Algorithm 3: Algoritmo para NCO1

Data: Función $function(x)$, límites a y b

Result: Resultado de la integral utilizando NCO1

```

1  $h \leftarrow (b - a)/3;$ 
2  $x0 \leftarrow a + h;$ 
3  $x1 \leftarrow x0 + h;$ 
4  $integral \leftarrow (3h/2) \cdot (function(x0) + function(x1));$ 
5 return  $integral;$ 
```

Algorithm 4: Algoritmo para NCO2

Data: Función $function(x)$, límites a y b

Result: Resultado de la integral utilizando NCO2

```
1  $h \leftarrow (b - a)/4;$ 
2  $x0 \leftarrow a + h;$ 
3  $x1 \leftarrow x0 + h;$ 
4  $x2 \leftarrow x1 + h;$ 
5  $integral \leftarrow (4h/3) \cdot ((2 \cdot function(x0)) - function(x1) + (2 \cdot function(x2)));$ 
6 return  $integral;$ 
```

Algorithm 5: Algoritmo para NCO3

Data: Función $function(x)$, límites a y b

Result: Resultado de la integral utilizando NCO3

```
1  $h \leftarrow (b - a)/5;$ 
2  $x0 \leftarrow a + h;$ 
3  $x1 \leftarrow x0 + h;$ 
4  $x2 \leftarrow x1 + h;$ 
5  $x3 \leftarrow x2 + h;$ 
6  $integral \leftarrow (5h/24) \cdot ((11 \cdot function(x0)) + function(x1) + function(x2) + (11 \cdot function(x3)));$ 
7 return  $integral;$ 
```

Los métodos cerrados para el grado 1, 2, 3, 4 tienen los siguientes algoritmos

Algorithm 6: Algoritmo para NCC1

Data: Función $function(x)$, límites a y b

Result: Resultado de la integral utilizando NCC1

```
1  $h \leftarrow (b - a)/1;$ 
2  $x0 \leftarrow a;$ 
3  $x1 \leftarrow x0 + h;$ 
4  $integral \leftarrow (h/2) \cdot (function(x0) + function(x1));$ 
5 return  $integral;$ 
```

Algorithm 7: Algoritmo para NCC2

Data: Función $function(x)$, límites a y b

Result: Resultado de la integral utilizando NCC2

```
1  $h \leftarrow (b - a)/2$ ;  
2  $x_0 \leftarrow a$ ;  
3  $x_1 \leftarrow x_0 + h$ ;  
4  $x_2 \leftarrow x_1 + h$ ;  
5  $integral \leftarrow (h/3) \cdot (function(x_0) + 4 \cdot function(x_1) + function(x_2))$ ;  
6 return  $integral$ ;
```

Algorithm 8: Algoritmo para NCC3

Data: Función $function(x)$, límites a y b

Result: Resultado de la integral utilizando NCC3

```
1  $h \leftarrow (b - a)/3$ ;  
2  $x_0 \leftarrow a$ ;  
3  $x_1 \leftarrow x_0 + h$ ;  
4  $x_2 \leftarrow x_1 + h$ ;  
5  $x_3 \leftarrow x_2 + h$ ;  
6  $integral \leftarrow (3h/8) \cdot (function(x_0) + 3 \cdot function(x_1) + 3 \cdot function(x_2) + function(x_3))$ ;  
7 return  $integral$ ;
```

Algorithm 9: Algoritmo para NCC4

Data: Función $function(x)$, límites a y b

Result: Resultado de la integral utilizando NCC4

```
1  $h \leftarrow (b - a)/4$ ;  
2  $x_0 \leftarrow a$ ;  
3  $x_1 \leftarrow x_0 + h$ ;  
4  $x_2 \leftarrow x_1 + h$ ;  
5  $x_3 \leftarrow x_2 + h$ ;  
6  $x_4 \leftarrow x_3 + h$ ;  
7  $integral \leftarrow (2h/45) \cdot (7 \cdot function(x_0) + 32 \cdot function(x_1) + 12 \cdot function(x_2) + 32 \cdot$   
    $function(x_3) + 7 \cdot function(x_4))$ ;  
8 return  $integral$ ;
```

Otros métodos, que mejoran bastante la precisión, son los métodos denominados de cuadratura Gaussiana, para los cuales se tienen los siguientes algoritmos para los grados 1, 2, 3, 4, 5

Algorithm 10: Algoritmo para GC1

Data: Función $function(x)$, límites a y b

Result: Resultado de la integral utilizando GC1

```

1  $x \leftarrow 1/\sqrt{3};$ 
2  $t1 \leftarrow \frac{(b-a) \cdot x + a + b}{2};$ 
3  $t2 \leftarrow \frac{(b-a) \cdot (-x) + a + b}{2};$ 
4  $integral \leftarrow \frac{b-a}{2} \cdot (function(t1) + function(t2));$ 
5 return  $integral;$ 

```

Algorithm 11: Algoritmo para GC2

Data: Función $function(x)$, límites a y b

Result: Resultado de la integral utilizando GC2

```

1  $x \leftarrow \frac{\sqrt{15}}{5};$ 
2  $t1 \leftarrow \frac{(b-a) \cdot x + a + b}{2};$ 
3  $t2 \leftarrow \frac{(b-a) \cdot (-x) + a + b}{2};$ 
4  $t3 \leftarrow \frac{(b-a) \cdot 0 + a + b}{2};$ 
5  $p1 \leftarrow \frac{5}{9} \cdot (function(t1) + function(t2));$ 
6  $p2 \leftarrow \frac{8}{9} \cdot function(t3);$ 
7  $integral \leftarrow \frac{b-a}{2} \cdot (p1 + p2);$ 
8 return  $integral;$ 

```

Algorithm 12: Algoritmo para GC3

Data: Función $function(x)$, límites a y b

Result: Resultado de la integral utilizando GC3

```
1  $x1 \leftarrow \frac{1}{35} \cdot \sqrt{525 - 70 \cdot \sqrt{30}};$ 
2  $x2 \leftarrow \frac{1}{35} \cdot \sqrt{525 + 70 \cdot \sqrt{30}};$ 
3  $t1 \leftarrow \frac{(b-a) \cdot x1 + a + b}{2};$ 
4  $t2 \leftarrow \frac{(b-a) \cdot (-x1) + a + b}{2};$ 
5  $t3 \leftarrow \frac{(b-a) \cdot x2 + a + b}{2};$ 
6  $t4 \leftarrow \frac{(b-a) \cdot (-x2) + a + b}{2};$ 
7  $p1 \leftarrow \frac{1}{36} \cdot (18 + \sqrt{30}) \cdot (function(t1) + function(t2));$ 
8  $p2 \leftarrow \frac{1}{36} \cdot (18 - \sqrt{30}) \cdot (function(t3) + function(t4));$ 
9  $integral \leftarrow \frac{b-a}{2} \cdot (p1 + p2);$ 
10 return  $integral;$ 
```

Algorithm 13: Algoritmo para GC4

Data: Función $function(x)$, límites a y b

Result: Resultado de la integral utilizando GC4

```
1  $x1 \leftarrow \frac{1}{21} \cdot \sqrt{245 - 14 \cdot \sqrt{70}};$ 
2  $x2 \leftarrow \frac{1}{21} \cdot \sqrt{245 + 14 \cdot \sqrt{70}};$ 
3  $t1 \leftarrow \frac{(b-a) \cdot 0 + a + b}{2};$ 
4  $t2 \leftarrow \frac{(b-a) \cdot x1 + a + b}{2};$ 
5  $t3 \leftarrow \frac{(b-a) \cdot (-x1) + a + b}{2};$ 
6  $t4 \leftarrow \frac{(b-a) \cdot x2 + a + b}{2};$ 
7  $t5 \leftarrow \frac{(b-a) \cdot (-x2) + a + b}{2};$ 
8  $p1 \leftarrow \frac{128}{225} \cdot function(t1);$ 
9  $p2 \leftarrow \frac{1}{900} \cdot (322 + 13 \cdot \sqrt{70}) \cdot (function(t2) + function(t3));$ 
10  $p3 \leftarrow \frac{1}{900} \cdot (322 - 13 \cdot \sqrt{70}) \cdot (function(t4) + function(t5));$ 
11  $integral \leftarrow \frac{b-a}{2} \cdot (p1 + p2 + p3);$ 
12 return  $integral;$ 
```

Algorithm 14: Algoritmo para GC5

Data: Función $function(x)$, límites a y b

Result: Resultado de la integral utilizando GC5

```
1  $x1 \leftarrow 0,932469514203152;$ 
2  $x2 \leftarrow 0,661209386466265;$ 
3  $x3 \leftarrow 0,238619186083197;$ 
4  $t1 \leftarrow \frac{(b-a) \cdot x1 + a + b}{2};$ 
5  $t2 \leftarrow \frac{(b-a) \cdot (-x1) + a + b}{2};$ 
6  $t3 \leftarrow \frac{(b-a) \cdot x2 + a + b}{2};$ 
7  $t4 \leftarrow \frac{(b-a) \cdot (-x2) + a + b}{2};$ 
8  $t5 \leftarrow \frac{(b-a) \cdot x3 + a + b}{2};$ 
9  $t6 \leftarrow \frac{(b-a) \cdot (-x3) + a + b}{2};$ 
10  $c1 \leftarrow 0,171324492379170;$ 
11  $c2 \leftarrow 0,360761573048193;$ 
12  $c3 \leftarrow 0,467913934572691;$ 
13  $p1 \leftarrow c1 \cdot (function(t1) + function(t2));$ 
14  $p2 \leftarrow c2 \cdot (function(t3) + function(t4));$ 
15  $p3 \leftarrow c3 \cdot (function(t5) + function(t6));$ 
16  $integral \leftarrow \frac{b-a}{2} \cdot (p1 + p2 + p3);$ 
17 return  $integral;$ 
```

Algunos resultados obtenidos de esta aplicación se muestran a continuación

f(x) = sen(x)			
Metodo	a	b	int
NC00	0.000000	0.785398	0.300559
NC01	0.000000	0.785398	0.297988
NC02	0.000000	0.785398	0.292859
NC03	0.000000	0.785398	0.292869
NCC1	0.000000	0.785398	0.277680
NCC2	0.000000	0.785398	0.292933
NCC3	0.000000	0.785398	0.292911
NCC4	0.000000	0.785398	0.292893
GC1	0.000000	0.785398	0.292867
GC2	0.000000	0.785398	0.292893
GC3	0.000000	0.785398	0.292893
GC4	0.000000	0.785398	0.292893
GC5	0.000000	0.785398	0.292893

Figura 9: Aplicación de técnicas de integración numérica.

f(x) = x**2 ln(x)			
Metodo	a	b	int
NC00	1.000000	1.500000	0.174331
NC01	1.000000	1.500000	0.180313
NC02	1.000000	1.500000	0.192272
NC03	1.000000	1.500000	0.192268
NCC1	1.000000	1.500000	0.228074
NCC2	1.000000	1.500000	0.192245
NCC3	1.000000	1.500000	0.192253
NCC4	1.000000	1.500000	0.192259
GC1	1.000000	1.500000	0.192269
GC2	1.000000	1.500000	0.192259
GC3	1.000000	1.500000	0.192259
GC4	1.000000	1.500000	0.192259
GC5	1.000000	1.500000	0.192259

Figura 10: Aplicación de técnicas de integración numérica.

$f(x) = x^2 \exp(-x)$			
Metodo	a	b	int
NC00	0.000000	1.000000	0.151633
NC01	0.000000	1.000000	0.153900
NC02	0.000000	1.000000	0.159043
NC03	0.000000	1.000000	0.159514
NCC1	0.000000	1.000000	0.183940
NCC2	0.000000	1.000000	0.162402
NCC3	0.000000	1.000000	0.161410
NCC4	0.000000	1.000000	0.160611
GC1	0.000000	1.000000	0.159410
GC2	0.000000	1.000000	0.160595
GC3	0.000000	1.000000	0.160603
GC4	0.000000	1.000000	0.160603
GC5	0.000000	1.000000	0.160603

Figura 11: Aplicación de técnicas de integración numérica.

3. Método de extrapolación Richardson

El método de Richardson, es un método el cual, mediante una cantidad de aproximaciones de algún valor o parámetro, obtiene una aproximación mas precisa de este mismo. [2]

Entonces, este método trabaja con una serie de aproximaciones, las cuales mediante una combinación de estas genera la aproximación mediante una extrapolación más precisa.

A continuación un ejemplo del método

Suponga que se quiere aproximar la derivada de una función $f(x)$ en un punto x_0 .

Se calculan dos aproximaciones iniciales de la derivada utilizando tamaños de paso h y $h/2$. Estas aproximaciones se denotan como D_h y $D_{h/2}$, respectivamente.

$$D_h = \frac{f(x_0 + h) - f(x_0)}{h}$$

$$D_{h/2} = \frac{f(x_0 + h/2) - f(x_0)}{h/2}$$

Se aplica el método de extrapolación de Richardson para obtener una aproximación más precisa de la derivada. La fórmula general de extrapolación de Richardson es:

$$D_{\text{Rich}} = D_h + \frac{D_h - D_{h/2}}{2^k - 1}$$

Donde D_{Rich} es la aproximación mejorada de la derivada, D_h y $D_{h/2}$ son las aproximaciones iniciales con tamaños de paso h y $h/2$ y k es el orden de la extrapolación.

Se sustituyen las aproximaciones iniciales D_h y $D_{h/2}$ en la fórmula y calculamos D_{Rich} .

La aproximación D_{Rich} es una estimación más precisa de la derivada de $f(x)$ en x_0 .

Si se requiere una mayor precisión, el proceso se puede repetir con tamaños de paso más pequeños y aplicar nuevamente la extrapolación de Richardson.

4. Método de Romberg

Este es un método para integración numérica, el cual mediante la conjunción de otras técnicas, como lo es el método trapezoidal, junto con el método de Richardson. [2]

Una ejemplificación de como funciona esto, es lo siguiente

Sí se quiere calcular la siguiente integral definida:

$$\int_0^1 e^{-x^2} dx$$

Se elije un valor de n para la partición. Comenzando con $n = 1$ para una aproximación inicial. Se calcula la aproximación inicial de la integral usando la regla compuesta trapezoidal:

$$R(0,0) = \frac{b-a}{2n} \left[f(a) + 2 \sum_{i=1}^{n-1} f\left(a + \frac{i(b-a)}{n}\right) + f(b) \right]$$

Para $n = 1$ y $a = 0, b = 1$, esto resulta en:

$$R(0,0) = \frac{1}{2} [e^0 + 2 \cdot e^{-1/4} + e^{-1}] \approx 0,74682$$

Se crea una tabla de Romberg e ingresa la aproximación inicial $R(0,0)$ en la posición correspondiente:

$$\begin{array}{ccc} R(0,0) & & \\ R(1,0) & R(1,1) & \\ R(2,0) & R(2,1) & R(2,2) \end{array}$$

Se incrementa el valor de n (por ejemplo, $n = 2$) y calcula una nueva aproximación $R(1,0)$ usando la regla compuesta trapezoidal:

$$R(1,0) = \frac{b-a}{2n} \left[f(a) + 2 \sum_{i=1}^{n-1} f\left(a + \frac{i(b-a)}{n}\right) + f(b) \right]$$

Para $n = 2$ y $a = 0, b = 1$, esto resulta en:

$$R(1,0) = \frac{1}{4} [e^0 + 2 \cdot e^{-1/8} + 2 \cdot e^{-1/4} + 2 \cdot e^{-3/8} + e^{-1}] \approx 0,74682$$

Se calcula $R(1,1)$ utilizando la fórmula de extrapolación de Richardson:

$$R(1,1) = R(1,0) + \frac{R(1,0) - R(0,0)}{3}$$

Se sustituyen los valores para $R(1,0)$ y $R(0,0)$ en la fórmula:

$$R(1,1) = 0,74682 + \frac{0,74682 - 0,74682}{3} = 0,74682$$

Se ingresan $R(1,0)$ y $R(1,1)$ en la tabla de Romberg:

$$\begin{array}{ccc} R(0,0) & & \\ R(1,0) & R(1,1) & \\ R(2,0) & R(2,1) & R(2,2) \end{array}$$

Se puede repetir el procedimiento con valores de n cada vez más grandes hasta que la aproximación sea lo suficientemente precisa o hasta que alcances un valor máximo de n .

La aproximación final de la integral será $R(2,2)$ en la tabla de Romberg.

5. Referencias

- [1] Burden, R.L., Faires, J.D. and Burden, A.M. 2015. *Numerical analysis*. Cengage learning.
- [2] Quarteroni, A., Sacco, R. and Saleri, F. 1999. *Numerical Mathematics*. Springer.