Eddy Herrera Daza eherrera@javeriana.edu.co

Pontificia Universidad JAVERIANA Bogotá

Taller de Derivación e Integración Numérica

Análisis Numérico

1. Derivación

Teniendo en cuenta el teorema:

Supóngase que se tienen $\{x_0, \dots, x_n\}$ n+1 valores distintos en algún intervalo I y que $f(x) \in C^{n+1}(I)$ entonces, existe $P(x)$ para algún $\xi(x) \in I$ talque:

$$f(x) = P(x) + \frac{f^{(n+1)}(\xi(x))}{(n+1)!}\prod_{i=0}^{i=n}(x - x_i)$$

Luego, para $f(x_0) \approx$ con $x_0 \in I$ y donde $x_1 = x_0 + h$ se tiene que la formula dados dos puntos está por:

$$f(x_0) = \frac{f(x_0 + h) - f(x_0)}{h} - \frac{h}{2}f^{"}(\xi(x))$$

Donde, para valores pequeños de h el error está acotado por $\frac{|hM|}{2}$ con M es una cota de $|f^{"}(x)|$

a. Teniendo en cuenta lo anterior, genere una tabla para $f^{'}(1.8) \approx$ para los siguientes valores de $h = 0.1, 0.01, 0.0011, 0.0001, 0.00001$.
b. Calcule las cotas del error
c. Si $f^{'}(1.8) = 0.5\overline{55}$ considera que las cotas son adecuadas?
d. Cuál es el valor de h que proporciona una aproximación con una precisión de 10^{-4}
e. Supóngase que se tienen tres puntos dados por $x_0; x_1 = x_0 + h; x_2 = x_0 + 2h$ una de las formulas conocida de tres puntos es:

$$f'(x_0) = \frac{1}{2h}[-3f(x_0) + 4f(x_0 + h) - f(x_0 + h) - f(x_0 + 2h)] + \frac{h^2}{3}f^{(3)}(\xi_0)$$

Donde, $\xi_0$se encuentra entre x_0 y $x_0 + 2h$. Utilice está fórmula para encontrar $f^{'}(1.8)$

f. Realice una modificación de la fórmula de los tres puntos, tomando valores entre $(x_0 - h)$ y $(x_0 + h)$ y compare la magnitud del error con la fórmula de la parte e.
g. Determine una fórmula para cinco puntos alrededor de x_0 y aplíquela y compárela con todas las formulas anteriores
h. Aplique la fórmula adecuada para aproximar $f^{"}(1.8)$

i. Teniendo en cuenta que el error total (h)= error de redondeo + error de truncamiento, $e(h) = \frac{\xi}{h} + \frac{h^2}{6}M$ como una función del tamaño del paso. Encuentre el tamaño óptimo del paso.

j. El siguiente código en Python está dado para la aproximar $f'(1); f(x) = xe^x$, córralo y realice una gráfica que muestre como varia la precisión en función de h

```
#Comportamiento del error en diferenciación numérica
from math import*
def f(x):return x*exp(x)
r=5.436563656918091           #Valor exacto con 16 decimales
h=0.1
for i in range(15):
    d=(f(1+h)-f(1))/h
    e=abs(r-d)
    print('%18.15f %18.15f %18.15f %18.15f'%(h,r,d,e))
    h=h/10
```

k. Aplicación: En un circuito con un voltaje $E(t)$ y una inductancia L se tiene que: $E(T) = L\frac{di}{dt} + Ri$ donde R es la resistencia e i es la corriente. La siguiente tabla muestra la medida de la corriente para varios instantes de tiempo (segundos), con R=0.142 ohms y L=0.98 henries. Aproxime el voltaje para los valores de t

t	1.00	1.01	1.02	1.03	1.0
i	3.10	3.12	3.14	3.18	3.24

2. Integración

a. Teniendo en cuenta la regla de los trapecios y que el error de truncamiento está dado por:

$$T = -\frac{h^2}{12}(b-a)\,f''(z), \quad a \le z \le b$$

Estime el número mínimo de trapecios para aproximar $\int_0^2 \sin x dx$ donde la respuesta tenga un error absoluto menor de 0.0001.

b. Implemente el siguiente código en Python para evaluar una aproximación de $\int_0^2 \sin x dx$ para $m = 10{,}100{,}1000$ y compare los resultados anteriores utilizando la función **integrate** de la librería **SymPy** de Python.

```
def trapecios(f,a,b,m):
    h=(b-a)/m
    s=0
    for i in range(1,m):
        s=s+f(a+i*h)
    r=h/2*(f(a)+2*s+f(b))
    return r
```

c. Aplique lo anterior, para aproximar $\int_0^2 \sqrt{x}sinxdx$ y evaluar el error

d. Dados los siguientes puntos:

(0.1, 1.8), (0.2, 2.6), (0.3, 3.0), (0.4, 2.8), (0.5, 1.9)

Utilice la fórmula de Simpson para encontrar una aproximación del área bajo la curva y calculé su error. Qué resultado se obtendría si utilizamos la regla del trapecio

e. Con la fórmula de Simpson integrar iterativamente $\int_0^2 \sqrt{1+cos^2 x}dx$ hasta que el error de truncamiento sea menor de 0.0001

f. Desarrolle un código en Python que permita a través de la regla de Simpson calcular el área entre dos curvas.

g. Pruebe el código anterior, con $m = 10$, para encontrar el área entre las curvas dadas por: $f(x) = 4 + \cos(x+1)$; $g(x) = e^x sinx$

h. Teniendo en cuenta que las fórmulas de Simpson y de Trapecios pertenecen al grupo donde los nodos están igualmente espaciados ósea partición regular; lo cual no siempre arroja las mejores aproximaciones. Por lo tanto, la fórmula de la cuadratura de Gauss con dos puntos es una alternativa.

$$A = \int_a^b f(x)dx = \frac{b-a}{2}\int_{-1}^{1} f\left(\frac{b-a}{2}t + \frac{b+a}{2}\right)dt = \frac{b-a}{2}\left[f\left(-\frac{b-a}{2}\frac{1}{\sqrt{3}} + \frac{b+a}{2}\right) + f\left(\frac{b-a}{2}\frac{1}{\sqrt{3}} + \frac{b+a}{2}\right)\right]$$

Desarrolle un código en Python, que permita utilizar esta fórmula y aplíquela para aproximar $\int_1^2 xe^x dx$. Cuál es la precisión

i. Utilice la misma fórmula de cuadratura de Gauss, pero partcione la integral de la siguiente manera $\int_1^2 xe^x dx = \int_1^{1.5} xe^x dx + \int_{1.5}^2 xe^x dx$ mejoró el resultado?

j. Implemente el código de la parte h, para que se pueda aplicar sucesivamente la fórmula de la cuadratura de Gauss, incrementando el número de subintervalos hasta que se alcance una precisión de k decimales. Pruébelo para $\int_1^2 xe^x dx$ con cuatro decimales.

k. Realice una revisión de la librería SymPy para el caso de las integrales impropias. Es posible para estas integrales utilizar la regla de simpson?

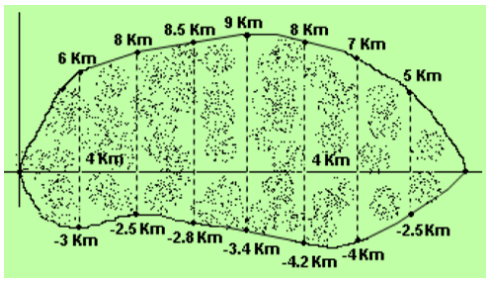
i. Aplique lo encontrado en el numeral anterior para aproximar las siguientes integrales: $\int_1^\infty \frac{dx}{(1+x^2)^3}dx$; $\int_0^1 \frac{sinx}{x}$dx

j. Utilice el siguiente código de la regla de Simpson en dos direcciones (para integrales dobles) para resolver el siguiente problema:
Un lago tiene una forma que aproximadamente es rectangular. Las dimensiones son 200 metros de ancho por 400 metros de largo. Se realiza una partición (grilla) para estimar aproximadamente la profundidad en metros en cada cuadricula de la malla como se muestra en la siguiente tabla de datos. Utilice los datos para estimar el volumen aproximado de agua que contiene el lago

X / Y	0	100	200	300	400
0	0	0	4	6	0
50	0	3	5	7	3
100	1	5	6	9	5
150	0	2	3	5	1
200	0	0	1	2	0

```
from sympy import*
from simpson import*
def simpson2(f,ax,bx,ay,by,mx,my):
    x=Symbol('x')
    dy=(by-ay)/my
    v=ay
    r=[]
    for i in range (0,my+1):
        def g(x): return f(x,v)
        u=simpson(g,ax,bx,mx)
        r=r+[u]
        v=v+dy
    s=0
    for i in range(1,my):
        s=s+2*(2-(i+1)%2)*r[i]
    s=dy/3*(r[0]+s+r[my])
    return s
```

k. En el siguiente gráfico se muestra delineada la zona de derrame de petróleo ocurrido en Caño Limón , donde las mediciones han sido obtenidas a distancias de 4Km. Con la fórmula de Simpson encuentre una aproximación del área total de afectación.



l. Desarrolle un código en Python que permita calcular la longitud de curva y aplíquela para aproximar el recorrido del planeta Mercurio, según la ley de Kepler con excentricidad de la órbita de 0.206 y una longitud del semieje mayor de 0.387 UA

m. Compare los resultados que obtuvo en la tabla que permite evaluar las probabilidades con la normal estándar si aplica el método de cuadratura.

n. Desarrolle un código en Python que permita generar una tabla, donde se pueda aproximar los valores de la distribución binomial a una normal, con una corrección por continuidad de 0.05 y para cuando $\rho = 0.5$ y $n = 1000$. Compare los valores aproximados con los valores exactos de la binomial.