Práctica 3 (21, 25 e 26 de novembro de 2024)

4 Integración Numérica. (Tema 3)

Neste tema imos ver algúns métodos numéricos (fórmulas de integración numérica) que nos van permitir aproximar o valor da integral de funcións que non teñen primitiva como combinación de funcións elementais, por exemplo $f(x) = e^{x^2}$. Ou para poder integrar funcións que veñen dadas por un conxunto de datos.

Unha fórmula de integración numérica (ou de cuadratura) é unha suma da forma:

$$\sum_{i=0}^{n} w_i f(x_i) \approx \int_a^b f(x)\,\mathrm{d}x,$$

onde os puntos $x_0, x_1, \ldots, x_n$ son os nodos e $w_0, w_1, \ldots, w_n$ son os pesos asociados a cada nodo.

Empezaremos vendo as fórmulas simples (que usamos para aproximar directamente a integral nun intervalo $[a, b]$) e finalizaremos coas fórmulas compostas (o intervalo de integración $[a, b]$ é dividido en subintervalos). Cargamos xa os módulos de Python que usaremos nesta práctica.

[2]:
```
import numpy as np
import sympy as sp
```

Ademais, tamén imos ver como se pode calcular unha integral en **Python** usando a función `integrate` do módulo **Sympy**. Sexa $f(x) = \arctan(x)$, sabemos que

$$\int f(x)\,\mathrm{d}x = x\arctan(x) - \frac{\ln(x^2+1)}{2} \text{ e que } \int_0^1 f(x)\,\mathrm{d}x = \frac{\pi}{4} - \frac{\ln(2)}{2}.$$

[3]:
```
x = sp.symbols('x')
f = sp.atan(x)
I = sp.integrate(f,x) # Integral indefinida

print('Unha primitiva de ',f, ' é:')
display(I)

Idef = sp.integrate(f,(x,0,1)) # Integral definida

print('A integral de ',f, ' entre 0 e 1 é: ')
display(Idef)
```

```
Unha primitiva de  atan(x)  é:
```

$$x\,\mathrm{atan}\,(x) - \frac{\log\left(x^2+1\right)}{2}$$

```
A integral de  atan(x)  entre 0 e 1 é:
```

$$-\frac{\log(2)}{2}+\frac{\pi}{4}$$

Nota: No caso de que SymPy non sexa capaz de calcular unha primitiva devolve como saída a integral de partida:

[4]:
```
I = sp.integrate(sp.sin(x*sp.cos(x)),x)
display(I)
```

$$\int \sin(x\cos(x))\,dx$$

Tamén poderíamos calcular integrais impropias. Vexamos por exemplo $\int_0^\infty \frac{2}{\sqrt{x}(x+4)}\,\mathrm{d}x$ e $\int_{-1}^{1}\frac{1}{x^3}\,\mathrm{d}x$.

[5]:
```
f = 2/(sp.sqrt(x)*(x+4))
I1 = sp.integrate(f,(x,0,sp.oo))

# Unha forma de combinar texto e variables con display
from IPython.display import Math
display(Math(f'A \ integral \ entre \ 0 \ e \ +{sp.latex(sp.oo)} \ de \ {sp.
 ↪latex(f)} \ é: \ {sp.latex(I1)}'))

g = 1/x**3
I2 = sp.integrate(g,(x,-1,1))
display(Math(f'A \ integral \ entre \ -1 \ e \ 1 \ de \ {sp.latex(g)} \ é: \ {sp.
 ↪latex(I2)}'))

# Vexamos a anterior integral descomposta en dúas de segunda especie:
a,b = sp.symbols('a:b', real=True)
I21 = sp.limit(sp.integrate(g,(x,-1,b)),b,0,'-')
I22 = sp.limit(sp.integrate(g,(x,a,1)),a,0,'+')
display(Math(f'A \ integral \ entre \ -1 \ e \ 0 \ de \ {sp.latex(g)} \ é: \ {sp.
 ↪latex(I21)}'))
display(Math(f'A \ integral \ entre \ -0 \ e \ 1 \ de \ {sp.latex(g)} \ é: \ {sp.
 ↪latex(I22)}'))
display(Math(f'A \ integral \ entre \ -1 \ e \ 1 \ de \ {sp.latex(g)} \ é: \ {sp.
 ↪latex(I21+I22)}'))
```

$$A\ integral\ entre\ 0\ e\ +\infty\ de\ \frac{2}{\sqrt{x}\,(x+4)}\ é:\ \pi$$

$$A\ integral\ entre\ -1\ e\ 1\ de\ \frac{1}{x^3}\ é:\ \mathrm{NaN}$$

$$A\ integral\ entre\ -1\ e\ 0\ de\ \frac{1}{x^3}\ é:\ -\infty$$

$$A\ integral\ entre\ -0\ e\ 1\ de\ \frac{1}{x^3}\ é:\ \infty$$

A integral entre -1 *e* 1 *de* $\dfrac{1}{x^3}$ *é* : NaN

Observación: O valor `NaN` obtido, significa "not a number" que é usado na maioría de linguaxes de programación para indicar que hai unha indeterminación.

4.1 Fórmulas simples

Veremos as fórmulas do rectángulo (dereita, esquerda e punto medio), do trapecio e de Simpson.

4.1.1 Fórmula do rectángulo

Esta é a fórmula máis sinxela de todas, xa que só necesitamos coñecer o valor da función nun punto. A fórmula xeral do rectángulo vén dada por:

$$\int_a^b f(x)\,\mathrm{d}x \approx (b-a)\,f(c), \quad c \in [a,b].$$

Casos particulares:

- **Rectángulo esquerda** se $c = a$.
- **Rectángulo dereita** se $c = b$.
- **Punto medio** (ou Poncelet) se $c = \dfrac{b+a}{2}$

Vexamos como utilizamos este método para aproximar a integral definida dunha función, $\displaystyle\int_0^1 \arctan(x)\,\mathrm{d}x$.

[7]:
```
x = sp.Symbol('x', real = True)
f_exp = sp.atan(x)
f = sp.lambdify(x,f_exp)

# Usamos a fórmula do punto medio para aproximar a integral da arcotanxente␣
↪entre 0 e 1 (xa calculada anteriormente)
a = 0
b = 1
PtoM = (a+b)/2

I01_atan = sp.integrate(f_exp,(x,a,b)) # Integral definida

# Avaliamos a función no punto medio
fPtoM  = f(PtoM)
I_PtoM  = (b-a)*fPtoM
print('O valor exacto é:',float(I01_atan),'\nMentres que o aproximado usando a␣
↪fórmula do punto medio é', I_PtoM)
print('\nO erro cometido é:', np.abs(float(I01_atan)-I_PtoM))
```

```
O valor exacto é: 0.43882457311747564
Mentres que o aproximado usando a fórmula do punto medio é 0.4636476090008061
```

```
O erro cometido é: 0.024823035883330458
```

Exercicio: Calcula a anterior integral usando os métodos do rectángulo esquerda e dereita.

[ ]:
```

```

4.1.2 Fórmula do trapecio

Para construír a fórmula do trapecio precisamos o valor da función en dous puntos (os extremos do intervalo). Obtense a partir do polinomio de Lagrange de grao 1 para os puntos $(a, f(a))$ e $(b, f(b))$. Vexamos como se obtén: * Construímos o polinomio de Lagrange para os nodos $x_0 = a$ e $x_1 = b$:

$$P_L(x) = f(x_0)l_0(x) + f(x_1)l_1(x), \quad l_0(x) = \frac{x - x_1}{x_0 - x_1} \text{ e } l_1(x) = \frac{x - x_0}{x_1 - x_0},$$

que substituíndo x_i polos seus valores obtemos:

$$P_L(x) = f(a)\frac{x-b}{a-b} + f(b)\frac{x-a}{b-a} = \frac{1}{b-a}\left(f(b)(x-a) - f(a)(x-b)\right).$$

* Calculamos a integral do polinomio de Lagrange de orde 1 no intervalo $[a, b]$:

$$\int_a^b P_L(x)\,\mathrm{d}x = \frac{f(b)}{b-a}\int_a^b (x-a)\,\mathrm{d}x - \frac{f(a)}{b-a}\int_a^b (x-b)\,\mathrm{d}x = \frac{b-a}{2}\left(f(a) + f(b)\right).$$

Polo tanto a fórmula do trapecio vén dada por:

$$\int_a^b f(x)\,\mathrm{d}x \approx \frac{b-a}{2}\left(f(a) + f(b)\right).$$

Programemos este método en **Python**:

[6]:
```python
def trapecio(a,b,fa,fb):
    aprox_tr = (b-a) * (fa + fb)/2
    return aprox_tr

# Resolvemos a mesma integral que co punto medio
a = 0
b = 1
fa  = f(a)
fb  = f(b)
I_Trap  =  trapecio(a,b,fa,fb)
print('O valor exacto é:',float(I01_atan),'\nMentres que o aproximado usando a␣
 ↪fórmula do trapecio é', I_Trap)
print('\nO erro cometido é:', np.abs(float(I01_atan)-I_Trap))
```

```
O valor exacto é: 0.43882457311747564
Mentres que o aproximado usando a fórmula do trapecio é 0.39269908169872414

O erro cometido é: 0.046125491418751496
```

4.1.3 Fórmula de Simpson:

Para construír a fórmula de Simpson precisamos o valor da función en tres nodos (os extremos do intervalo e o punto medio). Obteríase polo mesmo procedemento que a fórmula do trapecio só que neste caso utilizaríamos o polinomio de grao dous para interpolar os nodos $x_0 = a$, $x_1 = \dfrac{b+a}{2}$ e $x_2 = b$. A fórmula de Simpson vén dada pola seguinte expresión:

$$\int_a^b f(x)\,\mathrm{d}x \approx \frac{b-a}{6}\left(f(a) + 4f\left(\frac{b+a}{2}\right) + f(b)\right).$$

Queda a comprobación como exercicio (totalmente voluntario), para o cal debemos ter en conta que $(b-a)^3 = b^3 - 3b^2a + 3ba^2 - a^3$.

Exercicio: Programa unha función en **Python** para aproximar a integral dunha función usando o método de Simpson. Úsaa para aproximar a mesma integral que cos anteriores métodos.

[ ]:
```

```

4.2 Fórmulas compostas

As fórmulas compostas constrúense a partir das simples. Para isto debemos dividir o intervalo de integración $[a, b]$ en subintervalos $[x_{i-1}, x_i]$ e aproximar a integral de cada subintervalo usando a fórmula simple que desexemos. Imos supoñer que a lonxitude de cada subintervalo é constante e denotámola por $h = \dfrac{b-a}{n}$. Deste xeito, temos un conxunto de $n+1$ nodos $x_0, x_1, \ldots, x_n$ con $x_0 = a$, $x_n = b$ e $x_i = a + ih$, $i = 0, \ldots, n$. Veremos as fórmulas do punto medio composta, do trapecio composta e de Simpson composta, para as que imos considerar o caso máis restritivo onde os valores da función veñen dados por datos e non podemos obter o valor en máis puntos que os dados inicialmente.

4.2.1 Fórmula do punto medio composta

Para utilizar a fórmula do punto medio simple precisamos coñecer o valor no punto medio do intervalo no que imos aproximar a integral. Polo tanto, temos que utilizar 2 subintervalos de lonxitude h para construír un intervalo de lonxitude $2h$, $[x_{i-1}, x_{i+1}]$, con 3 nodos x_{i-1}, x_i, x_{i+1} no cal poidamos avaliar a función no punto medio $f(x_i)$. Para utilizar esta fórmula precisaremos que o número de subintervalos, n sexa par. Tendo isto en conta, podemos construír a seguinte aproximación da integral:

$$\begin{aligned}\int_a^b f(x)\,\mathrm{d}x = \sum_{i=1}^{\frac{n}{2}} \int_{x_{2i-2}}^{x_{2i}} f(x)\,\mathrm{d}x \quad &\approx \quad \sum_{i=1}^{\frac{n}{2}} (x_{2i} - x_{2i-2})\, f(x_{2i-1}) \\ &= \quad 2h \sum_{i=1}^{\frac{n}{2}} f(x_{2i-1}).\end{aligned}$$

Vexamos como se programa este método en **Python**. Utilizarémolo para aproximar a mesma integral que coas fórmulas simples.

[7]:
```python
# Seguimos utilizando o mesmo exemplo que para as fórmulas simples
x = sp.Symbol('x', real = True)
f_exp = sp.atan(x)
f = sp.lambdify(x,f_exp)
a = 0
b = 1

# Imos considerar 100 subintervalos (un número par)
n = 100
h = (b-a)/n # Lonxitude de cada subintervalo
xi = np.linspace(a,b,n+1) # Construímos os n+1 nodos
yi = f(xi) # Obtemos o valor da función nos n+1 nodos

def PtoMcomposto(y,h):
    n = len(y) - 1
    I_PMC = 2*h*np.sum(y[1:n:2])
    return I_PMC

I_PMC = PtoMcomposto(yi,h)
print('O valor exacto é:',float(I01_atan),'\nMentres que o aproximado usando a␣
 ↪fórmula do punto medio composto é', I_PMC)
print('\nO erro cometido é:', np.abs(float(I01_atan)-I_PMC))
```

```
O valor exacto é: 0.43882457311747564
Mentres que o aproximado usando a fórmula do punto medio composto é
0.43883290693697546

O erro cometido é: 8.333819499828365e-06
```

Observación: No caso de ter datos dunha función en nodos que non están equiespaciados tamén podemos aproximar a integral usando por exemplo a fórmula do rectángulo esquerda ou dereita composta. Consideramos para cada intervalo a súa lonxitude $h_i = x_i - x_{i-1}$ e obtemos:

$$\int_a^b f(x)\,\mathrm{d}x = \sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} f(x)\,\mathrm{d}x \approx \sum_{i=1}^{n} (x_i - x_{i-1})\, f(c_i) = \sum_{i=1}^{k} h_i f(c_i),$$

onde $c_i = x_{i-1}$ ou $c_i = x_i$ se consideramos rectángulo esquerda ou dereita, respectivamente. Vexamos a programación en **Python** do método do rectángulo esquerda composto. Imos utilizalo para aproximar a integral da seguinte función que ven dada por:

x_i	-1	0	0.5	1	3
$f(x_i)$	3	2.4	2.8	3.2	1.7

[22]:
```python
nodos = np.array([-1,0,0.5,1,3])
f_nodos = np.array([3,2.4,2.8,3.2,1.7])

def RE_composto(x,y):
```

```
    n = len(x) - 1
    hi = x[1:n+1]-x[0:n]
    I_REC = np.sum(np.multiply(hi,y[0:n]))
    return I_REC

I_REC = RE_composto(nodos,f_nodos)
print('O valor da integral usando a fórmula do rectángulo esquerda composta é',␣
 ↪I_REC)
```

```
O valor da integral usando a fórmula do rectángulo esquerda composta é 12.0
```

Exercicio: Programa a fórmula do rectángulo dereita composta e obtén a integral da función dada na anterior táboa.

[ ]:

```

```

4.2.2 Fórmula do trapecio composta

Neste caso non precisamos que o número de subintervalos sexa par, posto que se utilizan os valores nos extremos. Polo tanto, considerando que tódolos intervalos teñen a mesma lonxitude h e aplicando a fórmula do trapecio a cada un, obtemos:

$$\begin{aligned}
\int_a^b f(x)\,\mathrm{d}x = \sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} f(x)\,\mathrm{d}x &\approx \sum_{i=1}^{n} \frac{x_i - x_{i-1}}{2} \left(f\left(x_{i-1}\right) + f\left(x_i\right)\right) \\
&= \frac{h}{2}\left(f(x_0) + 2\sum_{i=1}^{n-1} f(x_i) + f(x_n)\right),
\end{aligned}$$

que é a fórmula do trapecio composta. Vexamos como a programamos en **Python**:

[9]:

```
def TrapecioComposto(y,h):
    n = len(y) - 1
    I_TC = h/2*(y[0]+2*np.sum(y[1:n])+y[n])
    return I_TC

# Usámolo para aproximar a mesma integral que nos apartados anteriores
I_TC = TrapecioComposto(yi,h)
print('O valor exacto é:',float(I01_atan),'\nMentres que o aproximado usando a␣
 ↪fórmula do trapecio composta é', I_TC)
print('\nO erro cometido é:', np.abs(float(I01_atan)-I_TC))
```

```
O valor exacto é: 0.43882457311747564
Mentres que o aproximado usando a fórmula do trapecio composta é
0.4388204064160859

O erro cometido é: 4.166701389751726e-06
```

Exercicio: Programa unha función coa que poidamos utilizar o método do trapecio composto para calcular a integral da función dada na seguinte táboa (igual que fixemos co método do rectángulo esquerda composto):

x_i	-1	0	0.5	1	3
$f(x_i)$	3	2.4	2.8	3.2	1.7

[ ]:

4.2.3 Fórmula de Simpson composta

Para poder utilizar este método precisamos, ao igual que coa fórmula do punto medio composta, ter un número de subintervalos, n, par. Isto débese a que precisamos coñecer o valor da función no punto medio do intervalo no que se quere aproximar o valor da integral. Deste xeito igual que para as fórmulas anteriores consideramos subintervalos de lonxitude constante h. Collendo dous subintervalos (igual que co punto medio composto) para aproximar a integral usando a fórmula de Simpson obtemos:

$$\begin{aligned}\int_a^b f(x)\,\mathrm{d}x = \sum_{i=1}^{\frac{n}{2}} \int_{x_{2i-2}}^{x_{2i}} f(x)\,\mathrm{d}x \quad &\approx \quad \sum_{i=1}^{\frac{n}{2}} \frac{x_{2i}-x_{2i-2}}{6}\left(f(x_{2i-2})+4f(x_{2i-1})+f(x_{2i})\right) \\ &= \quad \frac{h}{3}\left(f(x_0)+4\sum_{i=1}^{\frac{n}{2}} f(x_{2i-1})+2\sum_{i=1}^{\frac{n}{2}-1} f(x_{2i})+f(x_n)\right).\end{aligned}$$

Vexamos como programamos a fórmula de Simpson composta en **Python**.

[10]:
```python
def SimpsonComposto(y,h):
    n = len(y) - 1
    I_SC = h/3*(y[0] + 4*np.sum(y[1:n:2]) + 2*np.sum(y[2:n-1:2]) + y[n])
    return I_SC

# Usámolo para aproximar a mesma integral que nos apartados anteriores
I_SC = SimpsonComposto(yi,h)
print('O valor exacto é:',float(I01_atan),'\nMentres que o aproximado usando a␣
↪fórmula de Simpson composta é', I_SC)
print('\nO erro cometido é:', np.abs(float(I01_atan)-I_SC))
```

```
O valor exacto é: 0.43882457311747564
Mentres que o aproximado usando a fórmula de Simpson composta é
0.4388245732563824

O erro cometido é: 1.3890677497130355e-10
```

[ ]:

Exercicio:

- Obtén o valor exacto da seguinte integral $\int_0^1 \cos(2x)\sin(x)\,\mathrm{d}x$.

- Utiliza os métodos do punto medio, trapecio e Simpson compostos con 10, 100 e 1000 subintervalos ($h = 0.1$, $h = 0.01$, e $h = 0.001$) para aproximar a anterior integral.
- Estuda que sucede co erro ao aumentar o número de subintervalos. Cal foi o método co que obtivemos unha mellor aproximación.

```
[ ]:
```

Exercicio: En procedementos de diagnóstico médico por imaxe (como, por exemplo, a resonancia magnética), téñense en conta numerosos datos para obter, mediante cálculos computacionais, unha imaxe tridimensional que permita visualizar a parte do corpo que se estuda. O proceso é semellante ao empregado para calcular o volume dun sólido usando áreas de seccións perpendiculares a un eixe. A seguinte táboa indica o valor da área de algunhas seccións dun tumor, tomadas a unha distancia dun milímetro entre cada dúas imaxes:

$x(cm)$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$A(x)(cm^2)$	0	0.1	0.4	0.3	0.6	0.9	1.2	0.8	0.6	0.2	0.1

Estima o volume do tumor usando as fórmulas do punto medio, trapecio e Simpson compostas.

```
[ ]:
```