Analyse Numérique - SAGEmath/Python

Ines Abdeljaoued Tej [1]

Travaux pratiques numéro 3

Les formules de quadrature ntroduites ont été obtenues en substituant à l'intégrand f son polynôme d'interpolation de Lagrange, à des points équirépartis sur l'intervalle d'intégration. La valeur de l'intégrale de f est donc approchée par l'intégrale de son polynôme d'interpolation.

Pour améliorer la précision de cette approximation, on est donc tenté d'augmenter le degré de l'interpolation utilisée (le nombre de points). Le phénomène de Runge montre cependant que ceci pourrait amener à une approximation catastrophique.

1 Cas particuliers de la formule de Newton-Cotes

1.1 Méthode du trapèze

On obtient cete formule en remplaçant la fonction f par son polynôme d'interpolation de Lagrange de degré un aux points a et b. on a alors

$$I(f) = \int_a^b f(x)dx = \frac{b-a}{2}(f(a)+f(b))$$

Les poids de quadrature valent $\alpha_0 = \alpha_1 = \frac{b-a}{2}$, tandis que les noeuds sont $x_0 = a$ et $x_1 = b$. Si f est de classe C^2 sur $[a, b]$, on peut montrer que l'erreur de quadrature de cette forume vaut :

$$E(f) = -\frac{f'''(\eta)}{12}(b-a)^3, \qquad \eta \in]a, b[$$

1.2 Méthode de Simpson

Cette formule est obtenue en substituant à la fonction f son polynôme de Lagrange de de degré deux aux noeuds $x_0 = a$, $x_1 = \frac{a+b}{2}$ et $x_2 = b$ et s'écrit

$$I(f) = \frac{b-a}{6}(f(a) + 4f(\frac{a+b}{2}) + f(b))$$

Si f est de classe C^4 sur $[a, b]$, on peut montrer que l'erreur de quadrature de cette forume vaut :

$$E(f) = -\frac{f^{(4)}(\eta)}{2880}(b-a)^5, \qquad \eta \in]a, b[$$

1. `inestej@gmail.com`, `http://bit.ly/1JEWZiK`

2 Travail à faire sur Python

On cherche à représenter l'erreur de quadrature des formules de quadrature du trapèze et de Simpson, en fonction du nombre de so-intervalles m pour l'approximation de l'intégrale de $\frac{1}{1+x^2}$ sur $[-5,5]$. On va constater que l'erreur de quadrature $E(f)$, qui est la différence entre la valeur exacte de l'intégrale et sa valeur approchée, tend vers zéro lorsque m augmente.

1. Quelle est la valeur exacte de $\int_{-5}^{5} \frac{dx}{1+x^2}$
2. Trouvez la commande qui donne une valeur approchée de $2 * arctan(5)$
3. On cherche des valeurs approchée de cette intégrale d'abord avec la méthode du trapèze, ensuite avec la méthode de Simpson. Pour cela, écrivez deux fonctions qui dépendent du nombre de sous-intervalles m de $[-5,5]$ et qui implémentent les deux méthodes détaillées ci-dessous.
4. Tracez deux courbes reprenant l'erreur d'intégration pour chacune des méthodes en fonction de m.

Adaptez le code suivant pour répondre aux questions précédentes :

```
var(x)
def trapeze(a,b,f,N):
    X = [a+i*(b-a)/N for i in range(N+1)]
    somme = add([(b-a)/N*(f(X[i])+f(X[i+1]))/2 for i in range(N)])
    p = plot(f, x, a, b)
    q = line([(a,0), (a,f(a))])
    r = line([(b,0), (b,f(b))])
    s = Graphics()
    for i in range(N):
        s += line([(X[i],0), (X[i],f(X[i]))], rgbcolor='red')
        s += line([(X[i],f(X[i])),(X[i+1],f(X[i+1]))], rgbcolor='red')
        s += line([(X[i+1],0),(X[i+1],f(X[i+1]))], rgbcolor='red')
    show(p+q+r+s)
    return(somme.n())
trapeze(-1,1,1/(1+x^2),5)

def simpson(f,a,b,N):
    X = [a+i*(b-a)/(2*N) for i in range(2*N+1)]
    """
    for i in range(N-1):
        print (b-a)/(6*N)*(f(X[2*i]).n()+4*f(X[2*i+1]).n()+f(X[2*i+2]).n()).n(),
    """
    somme = add([(b-a)/(6*N)*(f(X[2*i]).n()+4*f(X[2*i+1]).n()+f(X[2*i+2]).n())
for i in range(N-1)])
    #time somme
    p = plot(f, x, a, b)
    q = line([(a,0), (a,f(a))])
    r = line([(b,0), (b,f(b))])
```

```
    s = Graphics()
    for i in range(N):
        s += line([(X[2*i],0), (X[2*i],f(X[2*i]))], rgbcolor=’red’)
        s += line([(X[2*i],f(X[2*i])),(X[2*i+1],f(X[2*i+1]))], rgbcolor=’red’)
        s += line([(X[2*i+1],f(X[2*i+1])),(X[2*i+2],f(X[2*i+2]))], rgbcolor=’red’)
        s += line([(X[2*i+2],0),(X[2*i+2],f(X[2*i+2]))], rgbcolor=’red’)
        s += point((X[2*i],f(X[2*i])), color = ’black’)
        s += point((X[2*i+1],f(X[2*i+1])), color = ’black’)
        s += point((X[2*i+2],f(X[2*i+2])), color = ’black’)
    show(p+q+r+s)
    return(somme.n())
simpson(1/(1+x^2),-5,5,5)
```