L’intégration numérique consiste à intégrer (de façon approchée) une fonction sur un intervalle borné [*a*,*b* ], c’est-à dire-calculer l’aire sous la courbe représentant la fonction, à partir d’un calcul ou d’une mesure en un nombre fini de point.

La répartition des points en abscisse est généralement uniforme (le pas d’échantillonnage *h* est constant) mais il existe des méthodes à pas variable, ou encore à pas adaptatif qui présente l’intérêt d’être plus précis mais plus longue à mettre en œuvre. Ces méthodes ne seront pas présentées dans ce TP.

Pour un pas constant, la précision de l’intégration numérique peut s’améliorer en augmentant le nombre de points *n* (en diminuant le pas d’échantillonnage *h*).

On pourra donc calculer l’intégrale recherchée en sommant :

* Des rectangles (degré 0): interpolation constante
* Des trapèzes (degré 1): interpolation linéaire
* Des figures se terminant par des paraboles (degré 2): interpolation quadratique

Intégration de degré 0

## Méthode des rectangles (point à gauche)

### Principes

La méthode des rectangles consiste à considérer la fonction d’interpolation constante sur chaque intervalle de largeur h. On peut la prendre égale à la valeur prise par le point à gauche du rectangle :

A l’aide du pseudo code vu en cours, programmer une fonction calculant l’intégrale en utilisant cette méthode. Cette fonction prendra en arguments :

* La fonction à intégrer
* Les bornes d’intégration a et b.
* Le nombre de points n

On a : . Elle retournera le résultat approché de l’intégrale.

### Précision

Nous allons tester le programme sur une fonction dont le résultat est connu.

On sait par exemple que : I1=

1. Tracer la fonction à intégrer entre 0 et pour la visualiser.
2. Tester votre programme sur cette fonction en prenant n=10.
3. En ajoutant une boucle « for » pour afficher les différents résultats, calculer l’intégrale pour n=100 et n=1000.
4. Dans la même boucle « for » et pour chaque résultat, calculer l’erreur commise lorsque l’on utilise l’intégration numérique : erreur=abs(I-I1).
5. Remplir le tableau en annexe et vérifier que l’erreur diminue en 1/n. . On dit que l’erreur est majorée par une constante multiplié par 1/n notée O(1/n)

## Méthode des rectangles (point à droite)

Sur la même feuille, copier-coller votre fonction d’intégration précédente et modifier la pour prendre en compte le fait que l’on prendra le point à droite du rectangle. Dans ce cas, on montre qu’une expression approchée de l’intégrale est :

A l’aide du pseudo code vu en cours, programmer une fonction calculant l’intégrale en utilisant cette méthode.

* Tester votre programme sur cette fonction en prenant n=10, n=100 et n=1000. On pourra faire une boucle « for » pour afficher les différents résultats.
* Dans la même boucle « for » et pour chaque résultat, calculer l’erreur commise lorsque l’on utilise l’intégration numérique.
* Remplir le tableau en annexe et vérifier que l’erreur diminue en 1/n.

## Méthode des rectangles (point au milieu)

Sur la même feuille, copier-coller votre fonction d’intégration précédente et modifier la pour prendre en compte le fait que l’on prendra le point à droite du rectangle. Dans ce cas, on montre qu’une expression approchée de l’intégrale est :

A l’aide du pseudo code vu en cours, programmer une fonction calculant l’intégrale en utilisant cette méthode.

* Tester votre programme sur cette fonction en prenant n=10, n=100 et n=1000. On pourra faire une boucle « for » pour afficher les différents résultats.
* Dans la même boucle « for » et pour chaque résultat, calculer l’erreur commise lorsque l’on utilise l’intégration numérique.
* Remplir le tableau en annexe et vérifier que l’erreur diminue en 1/n².

**Intégration de degré 1- Méthode des trapèzes

La méthode des trapèzes s’appuie sur une interpolation linéaire entre deux points. La valeur approchée de l’intégrale s’écrit alors :

A l’aide du pseudo code vu en cours, programmer une fonction calculant l’intégrale en utilisant cette méthode.

* Tester votre programme sur cette fonction en prenant n=10, n=100 et n=1000. On pourra faire une boucle « for » pour afficher les différents résultats.
* Dans la même boucle « for » et pour chaque résultat, calculer l’erreur commise lorsque l’on utilise l’intégration numérique.
* Remplir le tableau en annexe et vérifier que l’erreur diminue en 1/n².
* Vérifier que le résultat obtenu est bien la moyenne des résultats obtenus par la méthode des rectangles à gauche et celle des rectangles à droite.
* Constater également que la méthode des trapèzes est moins performante que celle du point milieu.

Intégration de degré 2- Méthode de Simpson

La méthode de Simpson consiste à interpoler la courbe par une parabole. La valeur approchée de l’intégrale s’écrit alors :

* Tester votre programme sur cette fonction en prenant n=10, n=100 et n=1000. On pourra faire une boucle « for » pour afficher les différents résultats.
* Dans la même boucle « for » et pour chaque résultat, calculer l’erreur commise lorsque l’on utilise l’intégration numérique.
* Remplir le tableau en annexe et vérifier que l’erreur diminue en 1/n4.

Utilisation de la bibliothèque Scipy

La bibliothèque scipy.integrate contient deux fonctions d’intégration numérique :

* trapz(f(x),x) qui intègre la fonction f sur un intervalle défini par le tableau donné par x par la méthode des trapèzes.
* simps(f(x),x) qui intègre la fonction f sur un intervalle défini par le tableau donné par x par la méthode de Simpson.

x contient donc les bornes d’intégration. Le nombre de points contenus dans x influe aussi sur la précision du résultat.

Faire un programme python utilisant ces deux fonctions. Votre programme devra permettre l’intégration de la même fonction que dans les exemples précédents. Vous intégrerez votre fonction en utilisant 10, 100 et 1000 points dans le tableau de x.

Pour chaque cas, calculer l’erreur et la comparer à celle obtenue dans vos programme « trapèze » et « simpson ». Conclusion.

**ANNEXE**:Erreur d’intégration en fonction de la méthode d’intégration numérique

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Méthode |  | n=10 | n=100 | n=1000 |
| Rectangle à gauche | Résultat | **0,261525178938** | **0,0147370999888** | **-0,00664456181012** |
| Erreur commise | **0,270508469959** | **0,0237203910099** | **0,00233872921101** |
| Variation erreur |  | Divisée par environ 10 | Divisée par environ 10 |
| Rectangle à droite | Résultat | **-0,205480442939** | **-0,0319634621989** | **-0,0113146180289** |
| Erreur commise | **0,196497151918** | **0,0229801711778** | **0,00233132700776** |
| Variation erreur |  | Divisée par environ 10 | Divisée par environ 10 |
| Rectangle au milieu | Résultat | **-0,0274838399554** | **-0,0091683458855** | **-0,00898514157193** |
| Erreur commise | **0,0185005489343** | **0,000185054864369** | **1,85055080425e-06** |
| Variation erreur |  | Divisée par environ 100 | Divisée par environ 100 |
| Trapèze | Résultat | **0,0280223679993** | **-0,00861318110509** | **-0,0089795899195** |
| Erreur commise | **0,0370056590204** | **0,000370109916035** | **3,70110162565e-06** |
| Moyenne(rec\_g+rec\_d) | **0,0280223679993** | **-0,00861318110505** | **-0,00897958991951** |
| Variation erreur |  | Divisée par environ 100 | Divisée par environ 100 |
| Méthode de Simpson | Résultat | **-0,00898177063717** | **-0,0089832909587** | **-0,00898329102112** |
| Erreur commise | **1,52038395969e-06** | **6,24324723864e-11** | **6,14959472234e-15** |
| Variation erreur |  | Divisée par environ 10000 | Divisée par environ 10000 |
| Trapèze avec Scipy | Résultat | **0.0367001212646** | **-0.00860566637985** | **-0.00897958250618** |
| Erreur commise | **0.0456834122858** | **0.00037762464128** | **3.70851494715e-06** |
| Variation erreur |  |  |  |
| Simpson avec Scipy | Résultat | **0.002378014178** | **-0.00897439048799** | **-0.00898328235328** |
| Erreur commise | **0.0113613051991** | **8.90053313478e-06** | **8.66785041659e-09** |
| Variation erreur |  |  |  |