Projet : Estimation numérique d'intégrale

Intégration numérique par méthodes des trapèzes et Simpson

Introduction

Certaines intégrales rencontrées en sciences ne possèdent pas de solution analytique simple, ou bien la fonction n'est connue que de manière discrète. L'intégration numérique permet d'obtenir une approximation contrôlable en fonction du nombre de subdivisions et de la méthode choisie.

Dans ce projet, nous étudions l'intégration de la fonction $f(x) = e^{-2x}$ sur l'intervalle $[0, 1]$ en utilisant les méthodes des trapèzes et de Simpson, et comparons les résultats obtenus.

Objectifs

- Implémenter les méthodes des trapèzes et Simpson.
- Comparer visuellement et numériquement les approximations par rapport à la fonction exacte.
- Étudier la convergence et le compromis précision / coût computationnel.

Méthodes numériques

Méthode des trapèzes

$$I \approx \frac{h}{2}\left(f(x_0) + 2f(x_1) + \cdots + 2f(x_{n-1}) + f(x_n)\right), \quad h = \frac{b-a}{n}.$$

Méthode de Simpson

$$I \approx \frac{h}{3}\left(f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \cdots + f(x_n)\right), \quad n \text{ pair}.$$

Code MATLAB — Fonctions utilisées

Fonction trapèzes

```
function I = trapeze_integration(f, a, b, n)
    h = (b - a) / n;
    x = a:h:b;
    y = f(x);
    I = h * (sum(y(2:end-1)) + (y(1) + y(end))/2);
end
```

Fonction Simpson

```
function I = simpson_integration(f, a, b, n)
    if mod(n, 2) ~= 0
        n = n + 1;
    end
    h = (b - a) / n;
    x = a:h:b;
    y = f(x);
    I = h/3 * (y(1) + 4*sum(y(2:2:end-1)) + 2*sum(y(3:2:end-2)) + y(end));
end
```

Tests et visualisations

Comparaison directe Trapèzes vs Simpson vs fonction exacte

```
f = @(x) exp(-2*x);
a = 0; b = 1;
n_demo = 10;

h = (b-a)/n_demo;
x_nodes = a:h:b;
y_nodes = f(x_nodes);

I_trap = trapeze_integration(f,a,b,n_demo);
I_simp = simpson_integration(f,a,b,n_demo);

x_fine = linspace(a,b,200);
y_fine = f(x_fine);

figure;
plot(x_fine, y_fine, 'k-', 'LineWidth',2); hold on; grid on;
plot(x_nodes, y_nodes, 'bo-', 'MarkerFaceColor','b');

% Trapezes
for i=1:n_demo
    plot([x_nodes(i) x_nodes(i+1)], [y_nodes(i) y_nodes(i+1)], 'b-', 'LineWidth',1.5);
    fill([x_nodes(i) x_nodes(i) x_nodes(i+1) x_nodes(i+1)], ...
         [0 y_nodes(i) y_nodes(i+1) 0], 'b', 'FaceAlpha',0.2);
end

% Simpson
for i=1:2:n_demo-1
    xx = linspace(x_nodes(i), x_nodes(i+2),50);
    coef = polyfit(x_nodes(i:i+2), y_nodes(i:i+2), 2);
    plot(xx, polyval(coef,xx), 'r-', 'LineWidth',1.5);
end

xlabel('x'); ylabel('f(x)');
title('Comparaison de f(x)=exp(-2x) : Trap zes vs Simpson');
legend('f(x)','Points','Trap zes','Simpson');
saveas(gcf,'comparaison_trap_simp.png');
```

Visualisation des figures

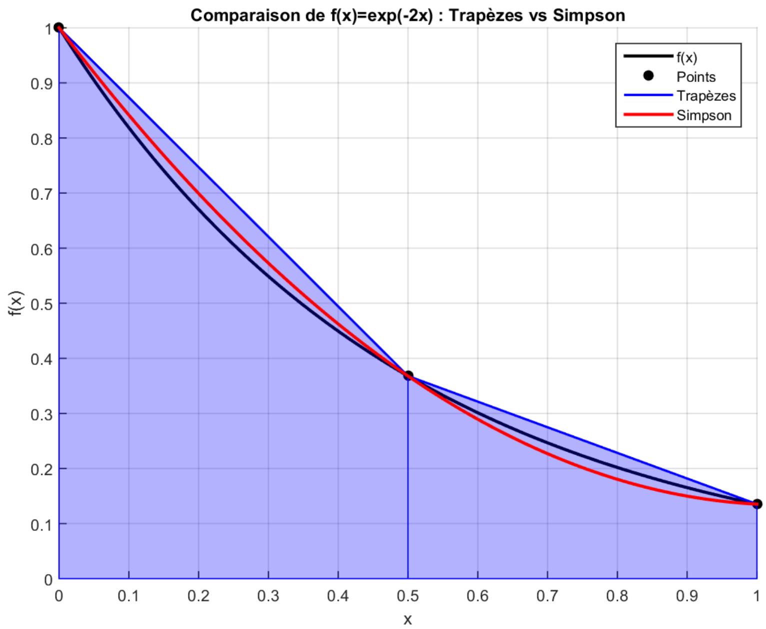

Figure : Comparaison directe entre Trapèzes, Simpson et la fonction exacte.

Analyse des résultats

- **Précision** : Simpson suit très bien la fonction exacte même pour un nombre de subdivisions faible.
- **Trapèzes** : approximation plus grossière, surtout visible pour peu de subdivisions.
- **Convergence** : Simpson converge beaucoup plus rapidement (ordre 4) que trapèzes (ordre 2).
- **Compromis** : Trapèzes est plus simple et rapide pour peu de précision, Simpson est préférable pour une meilleure précision.

Conclusion

La comparaison directe montre clairement que la méthode de Simpson est plus performante pour suivre fidèlement la fonction $f(x) = e^{-2x}$. Néanmoins, la méthode des trapèzes reste robuste et simple malgré sa précision.

Le choix de la méthode dépend du niveau de précision requis et du coût computationnel acceptable. L'intégration numérique permet ainsi de traiter efficacement des fonctions sans solution analytique simple.