

Tp0

- Ecrire une fonction qui reçoit comme argument une variable x et renvoie la valeur de x^3-6x+2 dans l'intervalle $[-4, 4]$.
- Ecrire un script qui fait appel et trace la fonction précédente avec plot
- Trouver une racine avec fzero après avoir choisi un point de départ judicieux sur le graphe

Gestion des axes :

Les différentes fonctions suivantes permettent de manipuler les graphiques.

- title : title('Texte du titre') Ajoute un titre à la figure.
- xlabel : xlabel('Unité des x')
- ylabel : ylabel('Unité des y')
- legend : legend('Nom de la courbe 1', 'nom courbe 2', ...)
- grid : grid on, grid off, quadrille ou non le graphique.

Tp1

Bissection :

On désire programmer à l'aide de Matlab la méthode de la bissection.

1. Créer une fonction Matlab nommée f qui évalue :

$$f(x) = \cos(x^x) - \sin(e^x)$$

1. tracer f dans l'intervalle $[0.5; 2.5]$.
2. écrire un programme qui calcule la racine approchée de f(x).
3. Exécuter le script avec $\epsilon=10^{-5}$ et en choisissant un intervalle $[a, b]$ approprié.

Point fixe :

On considère $f(x)=\ln(x+1)-x+0.2$ dans $[0.1, 1]$

1. définir la fonction f(x)
2. calculer f(0.1) et f(1), conclure .
3. tracer le graphe de f(x) pour $0.1 \leq x \leq 1$
4. tracer sur le même graphe les fonctions $f_1(x)=x$ et $f_2(x)=\ln(x+1)+0.2$ pour $0.1 \leq x \leq 1$
5. tracer $g'(x)$ pour $0.1 \leq x \leq 1$ et déduire que $\max |g'(x)|=k < 1$
6. écrire un programme qui calcule la racine approchée de f(x) on donne $x_0=0.5$ et $\epsilon=10^{-6}$

Newton :

La méthode de Newton permet d'approcher la racine par la relation suivante :

$$x_{i+1}=g(x_i)=x_i - F(x_i)/F'(x_i) , i=0,1,\dots$$

Avec comme critère d'arrêt $|x_{i+1}-x_i| < \epsilon$, ϵ étant la précision.

Appliquer cette méthode pour la recherche de la racine de la fonction suivante : $F(x)=e^x - 2\cos(x)$

1. dans un premier temps, on se propose de tracer les courbe de $f(x)$, $f'(x)$ et $f''(x)$ dans l'intervalle $[0, 1]$.
2. vérifier que les conditions de newton sont satisfaites en déduire x_0 .
3. Ecrire le script correspondant. Prendre $\epsilon=10^{-6}$.

TP2

1. Ecrire un programme qui permet de résoudre un système d'équations linéaires où A est une matrice triangulaire supérieure.
2. Ecrire un programme qui permet de résoudre un système d'équations linéaires par la méthode de Gauss.
3. Ecrire un programme qui permet de résoudre un système d'équations linéaires par la méthode de Crout (Décomposition LU).
4.
 - a-Ecrire un programme qui permet de résoudre un système d'équations linéaires par la méthode de Jacobie puis par Gauss –Seidel.
 - b. En utilisant la forme matricielle.

Tp3

Soit à calculer l'intégrale suivante : $I = \int_{-1}^2 \frac{e^x \sin x}{x^2 + 1} dx$

Dessiner le graphe de la fonction $f(x) = \frac{e^x \sin x}{x^2 + 1}$ sur l'intervalle $[-1, 2]$.

- a. Calculer la valeur de I avec la fonction trapz et quad de Matlab.
- b. Dans le script principal, écrire un programme permettant d'intégrer une fonction par la méthode des trapèzes.
- c. Dans le script principal, écrire un programme permettant d'intégrer une fonction par la méthode de Simpson.

Tp4

1. Soient les points d'interpolation suivants : (0,1), (1,2), (2,9) et (3,28)
 - a. Tracer y en fonction de x. En déduire (graphiquement) le polynôme passant par ces points (utiliser Tools (Basic Fitting)).
 - b. En utilisant polyfit (coeff=polyfit(x,y,n) où n est le degré du polynôme), déterminer les coefficients a_i du polynôme.
 - On donne $x_0=x(1) : 0.01 : x(\text{end})$. En utilisant polyval ($y_0 = \text{polyval}(\text{coeff}, x_0)$) renvoie la valeur d'un polynôme de degré n évalué en x_0 . coeff est un vecteur de longueur n+1 dont les éléments sont les coefficients du polynôme à évaluer $P_n(x)=a_n x^n + a_{n-1} x^{n-1} + \dots + a_0$.). Tracer y_0 en fonction de x_0 .
 - c. En utilisant la méthode de Lagrange, écrire un script qui permet l'interpolation d'une fonction en x_0 (en passant par les points précédents). Tracer $P_n(x_0)$ en fonction de x_0 .
 - d. Même chose pour la méthode de Newton.

Tp5

1.Dérivation numérique :

On considère :

$$f(x)=\sin(x)\Rightarrow f'(x)=\cos(x)$$

Calculer numériquement la dérivée de $f(x)$ sur l'intervalle $[0,2\pi]$ à l'aide des trois méthodes.

Tracer sur un même graphique :

- la dérivée exacte
- les dérivées numériques

2. Introduction aux EDO

A. Premier ordre :

Une EDO du premier ordre s'écrit :

$$dy/dx=f(x,y), y(x_0)=y_0$$

On considère l'équation différentielle du second ordre suivante :

$$y'(t)+y(t)=0, y(0)=1$$

Résoudre l'équation numériquement sur l'intervalle $[0,10]$ avec un pas $h=0.1$ en utilisant :

- la méthode d'Euler
- la méthode d'Euler modifié
- la méthode de Runge–Kutta d'ordre 4

Tracer les solutions obtenues et les comparer avec la solution exacte.

B. Second ordre :

On considère l'équation différentielle du second ordre suivante :

$$y''(t)+y(t)=0$$

avec les conditions initiales : $y(0)=1, y'(0)=0$

1. Transformer cette équation en un système d'équations différentielles du premier ordre.
2. Résoudre ce système numériquement sur l'intervalle $[0,10]$ avec un pas $h=0.1$ en utilisant :
 - la méthode d'Euler
 - la méthode d'Euler modifié
 - la méthode de Runge–Kutta d'ordre 4
3. Tracer les solutions obtenues et les comparer avec la solution exacte.