Université de Jijel
Département d'électrotechnique
Master I Electromécanique
Module : Méthodes numériques appliquées

TP_01

I. Résolution de système d'équations linéaires

I.1. Méthode de Jacobi

La méthode itérative générale de Jacobi consiste à écrire :

$$x_i = \frac{y_i - \sum_{\substack{j=1 \\ j\neq i}}^{n} a_{i,j}\, x_j}{a_{i,i}} \quad , \quad a_{i,i} \neq 0$$

et ceci pour chaque itération.

1- Appliquer la méthode de Jacobi pour trouver les racines du système d'équation linéaire suivant :

$$\begin{cases} x_1 + x_2 + x_3 = 1 \\ 2x_1 - x_2 + 3x_3 = 4 \\ 3x_1 + 2x_2 - 2x_3 = -2 \end{cases}$$

2- Ecrire un programme Matlab permettant de résoudre ce système (Prendre comme conditions initiales les valeurs $x_1 = x_2 = x_3 = 1$).

- Changer les valeurs initiales, exécuter le programme puis conclure.

3- Ecrire un programme Matlab permettant de résoudre ce système par l'utilisation d'un facteur de relaxation λ=0.2.

- Déterminer le nombre d'itérations nécessaires pour obtenir une erreur égale à 10^{-6}.

II. Résolution de système d'équations non Linéaires

II.1. Méthode de Newton-Raphson

Nous allons appliquer cette méthode à un cas simple : résolution d'un système de deux équations non linéaires à deux inconnues, par exemple :

$$\begin{cases} f(x,y) \\ g(x,y) \end{cases} \tag{1}$$

l'algorithme de Newton-Raphson devient :

$$\begin{pmatrix} x_k \\ y_k \end{pmatrix} = \begin{pmatrix} x_{k-1} \\ y_{k-1} \end{pmatrix} - \begin{pmatrix} \dfrac{\partial f(x,y)}{\partial x} & \dfrac{\partial f(x,y)}{\partial y} \\ \dfrac{\partial g(x,y)}{\partial x} & \dfrac{\partial g(x,y)}{\partial y} \end{pmatrix}^{-1}_{(x_{k-1},y_{k-1})} \begin{pmatrix} f(x_{k-1},y_{k-1}) \\ g(x_{k-1},y_{k-1}) \end{pmatrix} \tag{2}$$

1- Appliquer cette méthode pour trouver les racines du système d'équation non linéaire suivante:

$$\begin{cases} f(x,y) = x^2 - x + y^2 = 0 \\ g(x,y) = x^2 - y^2 - y = 0 \end{cases} \qquad (3)$$

Afin de déterminer si le système admet des solutions sur [-1, 1]x[-1,1] , on tracera les surfaces représentatives des deux fonctions dans le but de montrer l'existence éventuelle d'un couple (x,y) pour lequel elles s'annulent.

2- Ecrire un programme Matlab qui recherche les solutions du système par l'algorithme de Newton-Raphson. On prend comme conditions initiales les valeurs :

- x(1)=0.85; y(1)=0.35;
- x(1)=0.2; y(1)=0.1;

3- Dans Matlab, on peut obtenir la même solution avec la fonction 'fsolve'.

Chargé du TP
T. HACIB

Université de Jijel
Département d'électrotechnique
Master I Electromécanique
Module : Méthodes numériques appliquées

TP_02

I. Interpolation

I.1. Interpolation de Lagrange

Les masses volumiques d'un matériau pour différentes températures sont données par le tableau ci-dessous :

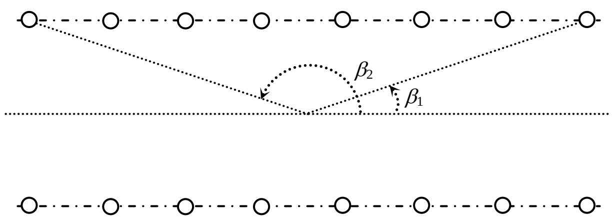
i	1	2	3
Température T (en C°)	94	205	371
Masse volumique R (T) (en kg/m³)	929	902	860

1- Ecrire la formule d'interpolation de Lagrange qui permet d'interpoler les différents points de données précédentes.
2- Trouver les masses volumiques pour T=251 $C°$, T=305 $C°$ et T=800 $C°$ en utilisant l'interpolation de Lagrange.
3- Utiliser la fonction Matlab "interp1" pour interpoler ces points de données.

II. Intégration numérique

II.1. Méthode du Trapèze

L'amplitude de l'induction magnétique dans le centre de l'axe d'une bobine (voir figure ci-dessous), est donnée par l'intégrale suivant:



$$B = \int_{\beta_1}^{\beta_2} \frac{\mu_0 \, n I}{2} sin\beta \;\, d\beta$$

Où μ_0 est la perméabilité de l'air ($4\pi.10^{-7}$ H/m), n est le nombre de tours (200) et I est le courant (1 A).
Si la bobine est suffisamment longue de tel sorte que $\beta_1 \approx 0$ et $\beta_2 \approx \pi$.

1- Donner une approximation de B en utilisant la méthode des trapèzes pour six sous-intervalles, chaqu'un égal à $\pi/6$.
2- Étudier l'influence du nombre de sous-intervalles (n) sur l'erreur d'intégration.
3- Au moyen de la commandes Matlab "trapz" évaluer cet intégrale.

Chargé du TP
T. HACIB

Université Mohamed Seddik Benyahia - Jijel
Département d'électrotechnique
Master I Electromécanique
Module : Méthodes numériques appliquées

TP_03

I. Résolution numérique des équations différentielles

I.1. Résolution d'une ED par la méthode d'Euler à l'ordre 1

Soit à résoudre l'équation différentielle suivante :

$$M.\frac{dV}{dt} = -C.V^2 + M.g \qquad \text{où :} \qquad \begin{cases} M = 70kg \\ g = 9{,}81N/kg \\ C = 0{,}27kg/m \end{cases}$$

1- Déterminer numériquement $V(t)$. On choisit un pas de temps h = 0.1 s, et on donne comme condition initiale $V(t = 0) = 0$.

2- Tracer la solution de cette équation différentielle pour $0 \leq t \leq 20$ s.

I.2. Résolution d'une ED par la méthode de Runge-Kutta

Une plaque métallique épaisse à la température de 200°C (ou 473°K) est soudainement placée dans une chambre de 25°K, où la plaque est refroidie à la fois par la convection naturelle et le transfert radiatif de chaleur. On donne les constantes physiques suivantes :

$$\begin{cases} \rho = 300kg/m^3 \rightarrow masse-volumique \\ V = 0{,}001m^3 \rightarrow volume \\ A = 0{,}25m^2 \rightarrow surface-d'échange \\ C = 900J/(kg.°K) \rightarrow chaleur-spécifique \\ h_c = 30J/(m^2 °K) \rightarrow coefficient-de-transfert-de-chaleur \\ \varepsilon = 0{,}8 \rightarrow émissivité \\ \sigma = 5{,}67.10^{-8}W/(m^2 °K^4) \rightarrow constante-de-Boltzmann \end{cases}$$

En supposant que la distribution de température dans le métal est uniforme, l'équation donnant la température en fonction du temps est :

$$\begin{cases} \dfrac{dT}{dt} = \dfrac{A}{\rho C.V}.\left(\varepsilon.\sigma\left(297^4 - T^4\right) + h_c.(297 - T)\right) \\ T(0) = 473 \end{cases}$$

1- Ecrire un programme Matlab permettant de résoudre cette équation différentielle par la méthode de Runge-Kutta à l'ordre 2 pour $0<t<180$ et h=1s.

2- En utilisant les fonctions interne de Matlab, écrire un programme permettant de résoudre cette équation par la méthode de Runge-Kutta à l'ordre 2 (ode23) puis à l'ordre 4 (ode45).

3- Tracer la solution de cette équation différentielle.

Chargé du TP
T. HACIB

Université Mohamed Seddik Benyahia - Jijel
Département d'électrotechnique
Master I électromécanique
Module : Méthodes numériques appliquées

TP_04

RESOLUTION DES EQUATIONS AUX DERIVEES PARTIELLES

1. Méthode des différences finies

Soit un fil métallique de section droite très petite par rapport à sa longueur L de façon à ce que le flux de chaleur existe seulement suivant la longueur du fil (Problème 1D). Si en plus la source de chaleur est absente, l'équation de Fourrier traduisant le transfert de chaleur par conduction prend la forme suivante:

$$\frac{\partial T}{\partial t} = a\frac{\partial^2 T}{\partial x^2}$$

Les conditions aux limites seront du type Dirichlet

$T(0) = 400$ et $T(L) = 100$.

Où $a=\lambda /c\rho$, représente la diffusivité thermique ($a = 1$).

c : Chaleur spécifique.

ρ : Masse volumique.

Q : Source de chaleur par unité de temps et de volume.

λ : Conductivité thermique.

1- Ecrire un programme dans Matlab permettant de résoudre cette équation aux dérivées partielles par la méthode des différences finies dans le cas stationnaire.

2. Ecrire un programme dans Matlab permettant de résoudre cette équation aux dérivées partielles par la méthode des différences finies dans le cas non-stationnaire (utiliser un schéma explicit) pour deux valeurs de Δt ($\Delta t = 0.1$ et $\Delta t = 0.01$).

Les conditions initiales sont $T(0, x) = 100$, pour $0 < x < L$.

Chargé du module
Pr. T. HACIB

Université Mohamed Seddik Benyahia - Jijel
Département d'électrotechnique
Master I électromécanique
Module : Méthodes numériques appliquées

TP_05

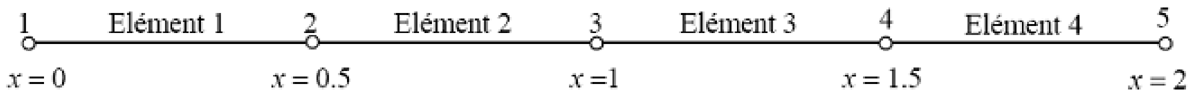
RESOLUTION DES EQUATIONS AUX DERIVEES PARTIELLES

2. Méthode des éléments finis

Soit l'équation aux dérivés partielles à résoudre sur un domaine Ω = [0, 2]:

$$\frac{du}{dx} + 2x(u-1) = 0 \quad \text{avec} \quad u(0) = 0$$

Ω est un domaine de dimension 1. Il est divisé en 4 éléments de taille 0.5. Chaque élément contient deux nœuds sur lesquelles la fonction u est interpolée.



- Ecrire un programme dans Matlab permettant de résoudre cette équation aux dérivées partielles par la méthode des éléments finis.

Chargé du module
Pr. T. HACIB

Université Mohamed Seddik Benyahia - Jijel
Département d'électrotechnique
Master I électromécanique
Module : Méthodes numériques appliquées

TP_06

1. Méthode d’optimisation sans contraintes

On s'intéresse à minimiser la fonction Sinus cardinal souvent utilisée pour illustrer le comportement des différents algorithmes d’optimisations sans contraintes. Cette fonction est caractérisée par l’existence d’un seul optimum global et plusieurs optima locaux. L’optimum global est $x^*=(0,0)$ avec $f_1(x^*)=-1$. Elle est définie de la façon suivante:

$$f_1(x_1,x_2) = -\left(\frac{\sin(\pi \cdot x_1)}{\pi \cdot x_1} + \frac{\sin(\pi \cdot x_2)}{\pi \cdot x_2}\right)$$
$$-3 \le x_1 \le 3$$
$$-3 \le x_2 \le 3$$

1- Implémenter la méthode du gradient conjugué et l’appliquer à la fonction $f_1(x)$ en partant de différents points initiaux.
2- Implémenter la méthode du Quasi-newton avec mise à jour BFGS et l’appliquer à la fonction $f_1(x)$ en partant de différents points initiaux.

2. Méthode d’optimisation avec contraintes

Dans ce cas, il faut minimiser la fonction objectif en respectant les contraintes d’égalités et de non égalités. Supposons que l’on veuille minimiser la fonction :

$$e^x(4x^2+2y^2+4xy+2y+1)$$

sous les contraintes :

$$\begin{cases} c_1 : xy - x - y \le -\dfrac{3}{2} \\ c_2 : y - x \le 10 \end{cases}$$

1- Implémenter la méthode de pénalité et l’appliquer à la fonction en partant de différents points initiaux.
2- Implémenter l’algorithme génétique et l’appliquer à la fonction en partant de différents points initiaux.

Chargé du module
Dr. T. HACIB