Informatique pour l'ingénieur, IFT-1903
Travail Pratique 1

A. Foriero, ing., Ph.D.

1 février 2016

Consignes

- Le travail doit obligatoirement être réalisé sur une base individuelle avec le logiciel MATLAB.
- Toutes les 12 questions valent le même nombre de points soit dix points.
- Chaque étudiant doit remettre un fichier contenant les réponses aux questions.
- Chaque étudiant doit maintenir une copie de son travail (en cas de perte de la copie remise aux correcteurs et pour consultation durant l'examen final de MATLAB).
- Le travail doit être remis le 19 février 2016 dans la boite (sur le site web) associée au TP1.
- Les travaux remis en retard seront assujettis à une pénalité.
- Les étudiants ne peuvent pas poser de questions concernant le travail pratique aux tuteurs et dépanneurs.
- Dans la mesure du possible, tout développement de fonctions utilisateur ou anonyme devra être réalisé sans les fonctions intégrées de MATLAB.
- Toute forme de plagiat sera sanctionnée (voir plan de cours) .

Problème 1

La développement en séries de Taylor de la fonction $f(x) = e^x$ est donné par

$$f(x) = e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + ...$$

Développez, avec le logiciel MATLAB, un fichier fonction utilisateur qui calcul la valeur de e^{-2} en utilisant n termes de cette série. La fonction doit aussi calculer l'erreur relative. (Utilisez la commande MATLAB *format long* pour calculer la valeur exacte de e^{-2})

Ensuite développez un fichier script dans lequel la fonction utilisateur est évoquée pour le calcul approximatif de e^{-2} et de l'erreur relative avec $n = 4, 5, 6, 7$ et 8 termes de la série. Le fichier script doit produire une sortie MATLAB en forme de tableau, soit trois colonnes ayant comme entête la valeur de n, la valeur approximative de e^{-2} et l'erreur relative.

Problème 2

La résolution de l'équation $f(x) = x^2 - s = 0$ est équivalente à la détermination de la racine carrée de s soit $\sqrt{s}$. Développez une fonction utilisateur MATLAB qui détermine la racine carrée d'un nombre positif par solution de l'équation à l'aide de la méthode de Newton-Raphson. Nommez la fonction **[Rs]=Racine(s)**. Les paramètres formels d'entrée et de sortie sont respectivement, s, le nombre pour lequel la racine carrée est déterminée et Rs, la réponse. La fonction MATLAB doit satisfaire les conditions suivantes:

- Vérifiez si l'entrée est positive, si non le programme devrait s'arrêter et afficher un message.
- La valeur initiale pour les itérations devrait être x=s.
- Les itérations devraient s'arrêter quand l'erreur relative estimée est inférieure à 0.00001.
- Le nombre d'itérations devraient se limiter à 25, si non le programme devrait s'arrêter et afficher un message.

  Utilisez la fonction Racine pour déterminer la racine carrée de 625, 1700 et -90.

  Finalement modifiez la fonction pour déterminer la solution de $f(x) = x^3 + 2x^2 - 5$. Utilisez comme valeur initiale $x_o = 3$.

Problème 3

Résolvez le système d'équations non-linéaires

$$e^{x1} - 2x_2 = 0$$

$$x_1^2 + x_2^2 - 25 = 0$$

avec la méthode de Newton-Raphson étendue à deux fonctions. Développez une fonction utilisateur dans l'environnement Matlab. Nommez la fonction **[xsol1,xsol2]= NewRaph2(F1,F2,F11,F22,F12,F21,x1i,x2i,iter,Err)**. Les paramètres formels d'entrée et de sortie sont respectivement des fonctions utilisateurs ou anonymes F1, F2 représentant les deux équations non-linéaires, des fonctions utilisateurs ou anonymes représentant les dérivées premières F11, F22 et croisées F12, F21 de F1, F2, les valeurs initiales x1i,x2i correspondant aux variables x1 et x2, le nombre d'itérations, iter, l'erreur relative, Err, et la solution xsol1, xsol2 correspondant aux valeurs x1 et x2. Expliquez votre choix des valeurs initiales x1i et x2i.

Problème 4

Développez une fonction utilisateur qui détermine les coordonnées polaires, d'un point avec coordonnées Cartésiennes dans un plan bidimensionnel. Pour la fonction, utilisez la nomenclature suivante **[r,d] = CartaPol(x,y)**. Les paramètres formels d'entrée sont les coordonnées Cartésiennes x et y du point. Les paramètres formels de sortie sont la distance radiale du point par rapport à l'origine, r, et l'angle en degrés mesuré par rapport à l'axe positif des x, d. La direction positive de l'angle par rapport à x est dans le sens anti-horaire de sorte que la valeur de d, qui est attribuée à un point, est positive dans les quatre quadrants. Utilisez la fonction pour déterminer les coordonnées polaires des points (12,7),(-15,7),(-10,-6) et (11,-5).

Problème 5

Développez une fonction utilisateur avec nomenclature **[N]=Seleca(M)** qui à comme but de trier une matrice, M, en ordre descendant produisant une matrice, N, de la même dimension. Utilisez l'algorithme du triage par sélection. Triez la matrice suivante

$$M = \begin{bmatrix} 7 & 9 & 2 \\ 8 & 4 & 5 \\ 6 & 1 & 3 \end{bmatrix}$$

pour obtenir

$$N = \begin{bmatrix} 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{bmatrix}.$$

Ensuite triez les matrices suivantes

$$A = \begin{bmatrix} 20 & 45 & 45 \\ 81 & 14 & 5 \\ 6 & 21 & 78 \end{bmatrix} B = \begin{bmatrix} -9.0 & 34.2 & 33.7 \\ 3.4 & 1.4 & 53 \\ 7.0 & 21.0 & 6.54 \end{bmatrix}.$$

Problème 6

Développez un fichier script dans lequel le système d'équations suivantes

$$x + y - z = 4$$

$$x - 2y + 3z = 6$$

$$2x + 3y + z = 7$$

est résolu. Utilisez six syntaxes MATLAB différentes pour le faire. Ensuite développez un fichier fonction avec la nomenclature **[b]=MULT(A,x)** qui a comme but de vérifier la solution. Le fichier fonction MULT.m ne doit pas contenir des fonctions intégrées de MATLAB (en d'autres mots il doit être programmé).

Problème 7

La séquence $\{a_n\}_{n=0}^{\infty}$ générée avec la relation récursive

$$a_n = -\frac{3(n-1)a_{n-1} + (n+1)a_{n-2}}{2(n-1)n}$$

se pose dans l'étude des solutions par séries des équations différentielles. Nous pouvons démontrer que la série

$$y(x) = \sum_{n=0}^{\infty} a_n (x-2)^n$$

est une solution de l'équation différentielle

$$2y'' + (x+1)y' + 3y = 0.$$

Une solution particulière voire fondamentale (toutes les autres solutions peuvent s'écrire en terme de celle-ci) survient pour le choix de $a_0 = 1$ et $a_1 = 0$. Développez une fonction utilisateur MATLAB avec nomenclature **Y1=DifPuis(x,N)** pour l'approximation de la solution en utilisant N+1 termes soit

$$Y_1(x,N) = \sum_{n=0}^{N} a_n (x-2)^n.$$

La fonction utilisateur pour l'approximation doit se baser sur une fonction utilisateur récursive pour le calcul des coefficients a_n. Ensuite avec MATLAB tracez les graphiques de $Y_1(x,5)$,$Y_1(x,10)$,$Y_1(x,15)$ sur l'interval $-3.5 \leq x \leq 7.5$.

Problème 8

Développez une fonction utilisateur qui trace la rotation d'une ellipse. La nomenclature de la fonction utilisateur est **ellipse_trace(xc,yc,a,b,angle)**. Les variables formelles d'entrée sont les coordonnées du centre de l'ellipse (xc,yc), les demi-longueurs des axes horizontaux, a, et verticaux, b, et l'angle de rotation en degrés, angle. La fonction n'a aucun paramètre formel de sortie. Utilisez la fonction pour tracer les ellipses suivantes:
a) xc=3.5, yc=2.0, a=8.5, b=3 avec une rotation de 20 degrés dans le sense anti-horaire et
b) xc=-5.0, yc=1.5, a=4, b=8 avec une rotation de 30 degrés dans le sens horaire.

Problème 9

Développez la fonction utilisateur avec nomenclature **[R]= newreshape(M,r,c)** qui reproduit la fonction intégrée de MATLAB reshape(A,m,n). L'évocation de la fonction doit produire un message d'erreur, *La matrice M n'est pas de taille* $r \times c$, si tel est le cas. Ensuite transformez la matrice

$$M = \begin{bmatrix} 7 & 6 & 2 \\ 2 & 4 & 1 \\ 6 & 3 & 3 \end{bmatrix}$$

avec l'évocation [R]=newreshape(M,1,9) pour produire $R = \begin{bmatrix} 7 & 2 & 6 & 6 & 4 & 3 & 2 & 1 & 3 \end{bmatrix}$ et [R]=newreshape(M,2,9) pour produire le message d'erreur.

Problème 10

Dans la conception des pipelines souterrains, il est nécessaire d'estimer la température ambiante du sol. La température du sol, en fonction de la profondeur, est estimée en modélisant le milieu souterrain comme un milieu solide à température ambiante initialement constante. La température à une profondeur donnée, z, et temps, t, est calculée avec l'expression suivante:

$$\frac{T(z,t) - T_s}{T_i - T_s} = erf(\frac{z}{2\sqrt{\alpha t}}) = \frac{2}{\pi} \int_0^{\frac{z}{2\sqrt{\alpha t}}} e^{-u^2} du$$

où T_s est la température à la surface du sol, T_i est la température ambiante initiale du sol souterrain et $\alpha = 0.128 \times 10^{-6}\ m^2/s$ est la diffusivité thermique. Utilisez la fonction intégrée de MATLAB **Y=erf(X)** et les valeurs $T_s = -20^oC$ et $T_i = 10^oC$ pour répondre aux questions suivantes:

1)Déterminez la température à une profondeur $z = 1.5$ m après 30 jours ($t = 2.592 \times 10^6$ s).

2) Développez un fichier script MATLAB qui génère un graphique de la température en fonction du temps à $z = 1.0$ m de profondeur sur une période de 31 jours. Utilisez des incréments de 1 jour.

3) Développez un fichiez script MATLAB qui génère un graphique tridimensionnel (T fonction de z et t) démontrant la température en fonction de la profondeur et temps sur les intervals $(0 \leq z \leq 3m)$ et $(0 \leq t \leq 2.592 \times 10^6 s)$.

Problème 11

Utilisez la méthode classique de Runge-Kutta (quatrième ordre) pour résoudre l'équation différentielle ordinaire

$$\frac{dy}{dx} = -1.1y + 6e^{-0.3x}$$

sur l'intervalle $(0 \leq x \leq 1.5)$ avec comme condition initiale $y = 3$ à $x = 0$. Comparez vos résultats avec la solution analytique (exacte) $y = 7.5e^{-0.3x} - 4.5e^{-1.1x}$. Quel pas de temps recommandez vous pour la méthode de Runge-Kutta.

Problème 12

Évaluez l'intégrale

$$I = \int_0^1 e^{sin(x)} dx$$

avec l'approche de Monte Carlo. Quelle est la précision de votre réponse? Tracez la courbe de l'approximation en terme des nombres d'essais pour vérifier si elle converge.