CH-250 Numerical methods in chemistry — October 30th, 2014

Part I: Final exam

1. (10 points) Consider a tri-diagonal (diagonally dominant) system of linear equations of arbitrarily (very) large order n. Which method would you use to solve this system? Why? Assuming that $n=3$, show conceptually how you would use the proposed method.

$$\begin{bmatrix} a_1 & -b_1 & 0 & \cdots & 0 \\ -c_2 & a_2 & -b_2 & \ddots & \vdots \\ 0 & -c_3 & a_3 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -b_{n-1} \\ 0 & \cdots & 0 & -c_n & a_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix}$$

2. (20 points) A space shuttle reentering the earth atmosphere decelerates in one hour from its orbital speed (7360.56 m/s) to its touchdown speed (96.11 m/s). Table 1 contains data of its measured velocity at equally spaced time-points.

Time (s)	-3600	-2400	-1200	0
Velocity (m/s)	7360.56	7245.61	6722.22	96.11

Table 1. Space shuttle descent velocity data

a) Based on the provided data, and using forward differences determine the acceleration values of the space shuttle. Detailed calculations and utilized expressions should be provided.

b) If the time vector is given in the variable t, and the accelerations are given in the vector A, write a sequence of Matlab commands that will give you information at which time the astronauts experienced their acceleration dropping below -5 m/s^2.

c) Using the composite trapezoid rule determine the total distance travelled by the shuttle from the first observation until the touchdown. Detailed calculations and utilized expressions should be provided.

3. (20 points) Consider the **velocity** profiles of the space shuttle shown in Figure 1.

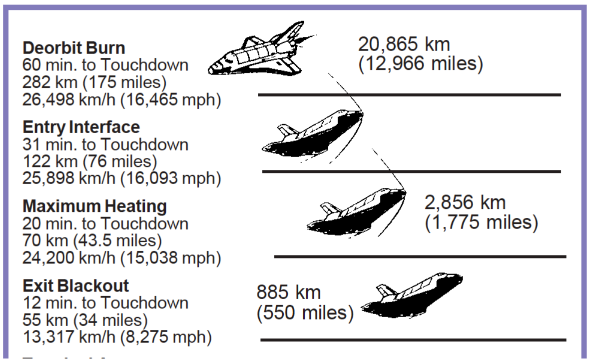

Figure 1: Space shuttle descent procedure (*source nasa.gov*)

a) Find the interpolating polynomial assuming that you have available data points from the Deorbit Burn phase to the Maximum heating phase.

b) What is the velocity of the space shuttle 25 minutes before the touchdown?

c) Find the interpolating polynomial if the data point from the Exit Blackout phase becomes available as well.

Detailed calculus and utilized expressions should be provided as well.

Student name: Sciper no:

Partie I: Examen final

1. (10 points) Observez ce système tridiagonal (à diagonale dominante) d'équations linéaires d'ordre n arbitrairement grand. Quelle méthode utiliseriez-vous pour résoudre ce système? Pourquoi? Pour $n=3$, montrez conceptuellement comment utiliser la méthode choisie.

$$\begin{bmatrix} a_1 & -b_1 & 0 & \cdots & 0 \\ -c_2 & a_2 & -b_2 & \ddots & \vdots \\ 0 & -c_3 & a_3 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -b_{n-1} \\ 0 & \cdots & 0 & -c_n & a_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix}$$

2. (20 points) Une navette spatiale de retour dans l'atmosphère terrestre décélère de sa vitesse orbitale (7360.56 m/s) à sa vitesse d'atterrissage (96.11 m/s) en une heure. Le tableau 1 contient sa vitesse mesurée en des points équidistants dans le temps.

Temps (s)	-3600	-2400	-1200	0
Vitesse (m/s)	7360.56	7245.61	6722.22	96.11

Tableau 1. Données de vitesse de la navette spatiale

a) A l'aide de ces données, utilisez les différences « en avant » (forward differences) pour déterminer les valeurs d'accélération de la navette spatiale. Les calculs détaillés et les expressions utilisées doivent également être fournis.
b) En supposant que les données de temps sont contenues dans un vecteur t et les accélérations dans un vecteur A, écrivez une séquence de commandes Matlab qui donne le temps pour lequel l'accélération subie par les astronautes est au dessous de $-5\ m/s^2$.
c) Utilisez la règle des trapèzes (composite trapezoid rule) pour déterminer la distance totale parcourue par la navette depuis la première observation jusqu'à l'atterrissage. Les calculs détaillés et les expressions utilisées doivent également être fournis.

3. (20 points) La figure 1 montre le profil des **vitesses** de la navette spatiale.

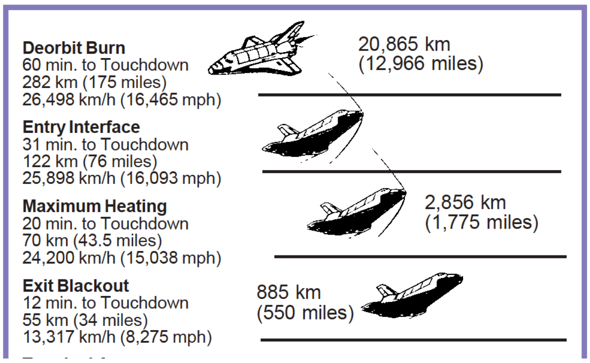

Figure 1: Procédure de descente de la navette spatiale (*source nasa.gov*)

a) Trouvez le polynôme d'interpolation en supposant que les points disponibles sont ceux des phases *Deorbit Burn* à *Maximum heating.*
b) Quelle est la vitesse de la navette spatiale 25 minutes avant l'atterrissage?
c) Trouvez le polynôme d'interpolation si les données du point *Exit Blackout* deviennent également disponible.

Les calculs détaillés et les expressions utilisées doivent également être fournis.

Student name: Sciper no:

4. (20 points) Consider the following system of ODEs

$$\frac{dy_1}{dt} = -80.6y_1 + 119.4y_2$$

$$\frac{dy_2}{dt} = 79.6y_1 - 120.4y_2$$

The general solution of this system can be written in the form:

$$\mathrm{y}(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = C_1 \begin{bmatrix} 3 \\ 2 \end{bmatrix} e^{-t} + C_2 \begin{bmatrix} -1 \\ 1 \end{bmatrix} e^{-200t}$$

a) Discuss which method you would apply to solve numerically the above-mentioned system of ODEs and what would be the appropriate step size.

b) For $C_1 = 1$ and $C_2 = 2$, apply the proposed method with the proposed step size to solve the system in the interval $t \in (0,1)$ (provide the details of the first step).

5. (5 points) The Runge-Kutta method, as learned in the course, does it belong to the group of explicit or implicit methods? Would you use it to solve stiff systems? Why?

6. (5 points) Explain which method you would choose for solving systems of nonlinear equations (continuous and continuously differentiable in the interval around the solution): a) to guarantee convergence to the solution;
b) to have the fastest convergence rate.

7. (20 points) Consider the following nonlinear equation on the interval (pi/2,pi):

$$y = \sin x - \frac{x}{2}$$

a) Using the Newton-Raphson method find numerically the root of this equation with the precision of 10^{-3}. Take as initial point $x_0=2$; the exact solution is $x^*=1.89549$.

b) Estimate the number of Newton-Raphson iterations needed to reach this root with a precision of 10^{-9}.

c) How many iterations would be needed to attain the accuracy of 10^{-9} if we use the bisection method?

Detailed calculus and utilized expressions should be provided as well.

8. (8 points) The *Matlab* function below *poly_function.m* evaluates the following polynomial $f(x) = 5x^3 + 3x^2 + 2x + 6$. We want to compute this polynomial for the given vector of coefficients *a* at the value *x=3*. The output argument of this function should be a scalar *pol*. Try to identify the four errors in this function.

```
>> a = [5 3 2 6];
>> x = 3;

>> polVal = poly_function(a, x);

% function
function pol = poly_function(a,Y)
     Y = 5;
     potency = Y.^(length(a):1);
     pol = sum(a*potency);
end
```

Student name: Sciper no:

4. (20 points) Le système d'ODE suivant

$$\frac{dy_1}{dt} = -80.6 y_1 + 119.4 y_2$$

$$\frac{dy_2}{dt} = 79.6 y_1 - 120.4 y_2$$

A une solution générale de la forme :

$$y(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = C_1 \begin{bmatrix} 3 \\ 2 \end{bmatrix} e^{-t} + C_2 \begin{bmatrix} -1 \\ 1 \end{bmatrix} e^{-200t}$$

a) Expliquez la méthode que vous utiliseriez pour résoudre numériquement le système d'ODE ci-dessus et donnez un choix approprié de pas d'intégration.

b) Pour $C_1 = 1$ et $C_2 = 2$, appliquez la méthode avec le pas d'intégration proposé pour résoudre le système dans l'intervalle $t \in (0,1)$ (donnez les détails du premier pas).

5. (5 points) Est-ce que la méthode de Runge-Kutta, telle qu'apprise dans le cours, est une méthode implicite ou explicite? Est-ce que vous l'utiliseriez pour résoudre un système "stiff"? Pourquoi?

6. (5 points) Expliquez quelle méthode vous utiliseriez pour résoudre un système d'équations non-linéaires (continues et continument différentiables dans l'intervalle proche de la solution): a) pour garantir une convergence vers la solution; b) pour avoir la plus rapide convergence.

7. (20 points) Considérez l'équation non-linéaire suivante dans l'intervalle (pi/2 , pi):

$$y = \sin x - \frac{x}{2}$$

a) En utilisant la méthode de Newton-Raphson, trouvez numériquement le zéro de cette équation avec une précision de 10^{-3}. Prenez comme estimation initiale *$x_0=2$*; la solution exacte est *$x^*=1.89549$*.

b) Estimez le nombre d'itérations nécessaires avec la méthode de Newton-Raphson pour trouver cette solution avec une précision de 10^{-9}.

c) Combien d'itérations seraient nécessaires pour atteindre la précision de 10^{-9} si on utilisait la méthode de la bissection?

Les calculs détaillés et les expressions utilisées doivent également être fournis.

8. (8 points) La fonction *Matlab* suivante, *poly_function.m*, évalue le polynôme $f(x) = 5x^3 + 3x^2 + 2x + 6$. On veut calculer ce polynôme avec le vecteur des coefficients *a* à la valeur *x=3*. Le paramètre de sortie de cette fonction doit être un scalaire, *pol*. Identifiez les quatre erreurs présentes dans cette fonction.

```
>> a = [5 3 2 6];
>> x = 3;

>> polVal = poly_function(a, x);

% function
function pol = poly_function(a,Y)
     Y = 5;
     potency = Y.^(length(a):1);
     pol = sum(a*potency);
end
```

9. Analyze the *Matlab* code given below where *A* is a matrix of size *CxC* and *b* is a vector of size *Cx1*.
(i) (6 points) Try to recognize and explain what is the purpose of this code. In which numerical method that you have seen in the class could it be used?
(ii) (6 points) A couple of lines of code is missing (indicated by the red rectangle) so that the code is not giving the expected outputs according to its intended use. Identify the purpose of the missing part and try to write the missing lines.

```
for col=1:C
    max = abs(A(col,col));
    max_pos = col;
    for i = col+1:C
        if abs(A(i,col)) > max
            max = abs(A(i,col));
            max_pos = i;
        end
    end

    temp = A(col,:);
    A(col,:) = A(max_pos,:);
    A(max_pos,:) = temp;
```

$\boxed{?}$

```
    for row=col+1:C
        mult_factor = A(row,col)/A(col,col);
        A(row,:) = A(row,:) - mult_factor * A(col,:);
        b(row) = b(row) - mult_factor * b(col);
    end
end
```

Student name: Sciper no:

9. Analysez le code *Matlab* donné ci-dessous, où *A* est une matrice de taille *CxC* et *b* est un vecteur de taille *Cx1*.
(i) (6 points) Essayez de reconnaître et expliquez le but de ce code. Pour quelle méthode numérique, vue en cours, pourrait-on utiliser ce code?
(ii) (6 points) Quelques lignes manquent dans ce code (indiquées par le rectangle rouge). A cause de ces lignes manquantes, les paramètres retournés par ce code n'ont pas les valeurs attendues pour l'usage du code. Identifiez le but de la partie manquante et écrivez les lignes de code manquantes.

```
for col=1:C
    max = abs(A(col,col));
    max_pos = col;
    for i = col+1:C
        if abs(A(i,col)) > max
            max = abs(A(i,col));
            max_pos = i;
        end
    end

    temp = A(col,:);
    A(col,:) = A(max_pos,:);
    A(max_pos,:) = temp;

            ?

    for row=col+1:C
        mult_factor = A(row,col)/A(col,col);
        A(row,:) = A(row,:) - mult_factor * A(col,:);
        b(row) = b(row) - mult_factor * b(col);
    end
end
```