

Part I exam

1. (15 points) Apply the Jacobi method to the system

$$x_1 - 5x_2 = -4$$

$$7x_1 - x_2 = 6$$

using the initial solution $(x_1, x_2) = (0, 0)$ and perform 3 iterations. Solve the system by hand and compare. Does the solution converge? Discuss. Provide all calculations and steps.

2. (20 points) Consider the following nonlinear equation on the interval $(2.5, \pi)$:

$$2e^x \sin x = 2x - 1$$

- Apply the Regula-Falsi method to find numerically the root of this equation. Perform 3 iterations. What is the attained accuracy if you know that the exact solution is $x^* = 3.01816421$.
- Repeat a. with the Newton-Raphson method and 2 iterations. Take as initial point $x_0 = 2.5$.

Detailed calculus and utilized expressions should be provided as well.

3. (25 points) Consider the following ODE

$$\frac{dx}{dt} + 2x(t) = 0, \quad x(0) = 3$$

- Perform 2 iterations of Adams-Bashforth-Moulton method (where we employ Adams-Bashforth predictor and Adams-Moulton corrector together)
 - Can this method be used for solving stiff systems?
 - Is the Euler backward method implicit or explicit? Explain.
- Detailed calculus and utilized expressions should be provided as well.

4. (15 points) The velocity of a car measured by its GPS device at 2sec intervals during a speed test is given in the table:

Time (s)	0	2	4	6	8
Velocity (m/s)	0	24.8	36.1	47.6	58.7

- Use central differences to approximate the acceleration of the car at $t=1$ s and $t=4$ s. At which of the two time point the approximation has a better accuracy?
 - Explain which Simpson method would be the most appropriate to estimate how many meters the car passed during the test? Perform the calculus, and provide the result and utilized expressions.
5. a) (3 points) The following function performs the successive over-relaxation method. Unfortunately, it does not give any output though there are no errors in syntax. Find what is missing in order to make it work (at the position of the asterisk *).

```
function [x count] = SOR(A, b, e, w, x0)
%Successive over-relaxation algorithm
%initialization
[m, n] = size(A);
x = x0;
count = 0;
errF = norm(A*x - b);
%main loop
while errF > e
    for i = 1:m
        x(i) = (1-w)*x(i) + w/A(i, i)*(b(i) - A(i, :) * x + A(i, i)*x(i));
    end
    *
    count = count + 1;
```

Student name:

Sciper no:

end

b) (5 points) The *Matlab* function below *rmvNegatives.m* is supposed to receive any row vector *Vin* as input, remove its negative elements, and return the resulting vector to the output *Vout*. Find and explain the conceptual (not syntax) mistake in the function code, and suggest a way to correct it so that the function works properly (the coding is not necessary, explain the concept).

```
>> a = [2 -1 5 -3 0 1 6];
>> b = rmvNegatives(a);

function Vout = rmvNegatives(Vin)
for i=1:size(Vin,2)
    if Vin(i)<0
        Vin(i) = [];
    end
end
Vout = Vin;
```

c) (8 points) The *Matlab* function *myNR.m* is supposed to be finding the solution of any function $F(x)=0$ based on the Newton-Raphson method, for a defined convergence threshold, and for a maximum number of iterations. The inputs to this function are: (i) the function handle *fun* for $F(x)$, (ii) the function handle *dfun* of its first derivative, and (iii) the initial guess x_0 of the solution. Fill in the lines marked with * (asterisk) so that the function is able to perform its purpose.

```
function NRsol = myNR(fun,dfun,x_old)
    max_it = 10^5;
    convThresh = 10^-5;
    iterationN=0;
    diff_x = 10^3;
*           % condition of the iteration loop
*     x_new =           % update of solution
    diff_x = abs(x_new-x_old);
    x_old = x_new;
*           % update of the current iteration number

    end
    NRsol = x_new;
end
```

d) (3 points) Write a script that forms the vector $V = [1, r^1, r^2, \dots, r^N]$ for $N=10$ and $r=1/2$. Do not use *for* or *while* loop.

e) (3 points) Consider the following function:

```
function [F,dF] = myFunction(fH,x0,h)
    F = fH(x0);
    dF = (fH(x0+0.5*h)-fH(x0-0.5*h))/h;
end
```

Explain what is its purpose.

f) (3 points) Write a script that defines an anonymous function for the expression $\sin(x)*e^{(2*x)}$ and that calls the function defined in e) with input arguments $x_0 = 2$ and $h = 1 \times 10^{-6}$.

Part I exam

1. (15 points) Appliquez la méthode de Jacobi sur le système suivant

$$x_1 - 5x_2 = -4$$

$$7x_1 - x_2 = 6$$

en utilisant comme solution initiale $(x_1, x_2) = (0, 0)$, et effectuez 3 itérations. Résolvez le système à la main et comparez. La solution converge-t-elle? Expliquez. Donnez tous les calculs et étapes.

2. (20 points) Considérez l'équation non linéaire suivante sur l'intervalle $(2.5, \pi)$:

$$2e^x \sin x = 2x - 1$$

- Appliquez la méthode regula falsi afin de trouver numériquement le zéro de cette équation. Effectuez 3 itérations. Quelle est la précision obtenue en sachant que la solution exacte est $x^* = 3.01816421$.
- Répétez le point a. avec la méthode de Newton-Raphson sur 2 itérations. Prenez comme point initial $x_0 = 2.5$;

Les calculs détaillés ainsi que les expressions utilisées doivent être rendus.

3. (25 points) Considérez l'équation différentielle ordinaire (ODE) suivante :

$$\frac{dx}{dt} + 2x(t) = 0, \quad x(0) = 3$$

- Effectuez 2 itérations de la méthode d'Adams-Bashforth-Moulton (où l'on emploie ensemble le prédicteur d'Adams-Bashforth et le correcteur d'Adams-Moulton)
- Est-ce que cette méthode peut être utilisée pour résoudre des systèmes rigides (stiff systems) ?
- La méthode « Euler backward » est-elle implicite ou explicite ? Expliquez.

Les calculs détaillés ainsi que les expressions utilisées doivent être rendus.

4. (15 points) La vitesse d'une voiture mesurée par son appareil GPS par intervalles de 2s lors d'un test de vitesse est donnée dans ce tableau :

Temps (s)	0	2	4	6	8
Vitesse (m/s)	0	24.8	36.1	47.6	58.7

- Utilisez les différences centrées (central differences) pour approximer l'accélération de la voiture en $t=1s$ et $t=4s$. Pour lequel de ces deux temps l'approximation a-t-elle une meilleure précision ?
 - Expliquez quelle méthode de Simpson serait la plus appropriée pour estimer combien de mètres la voiture a parcourus durant le test ? Effectuez les calculs, et donnez le résultat et les expressions utilisées.
5. a) (3 points) La fonction suivante effectue la méthode de surrelaxation successive (successive over-relaxation method). Malheureusement, aucun résultat n'en sort alors qu'il n'y a pas d'erreur de syntaxe. Trouvez ce qu'il manque à l'emplacement (*) afin de la faire fonctionner.

```
function [x count] = SOR(A,b,e,w,x0)
%Successive over-relaxation algorithm
%initialization
[m,n]=size(A);
x=x0;
count=0;
errF=norm(A*x-b);
%main loop
while errF > e
    for i = 1:m
        x(i)=(1-w)*x(i)+w/A(i,i)*(b(i)-A(i,:)*x+A(i,i)*x(i));
    end
    *
```

```

    count = count+1;
end

```

b) (5 points) La fonction *Matlab* ci-dessous *rmvNegatives.m* est censée recevoir un vecteur ligne *Vin* en entrée, supprimer ses éléments négatifs, et donner le vecteur résultant *Vout* en sortie. Trouvez l'erreur conceptuelle (non syntaxique) dans le code de la fonction, expliquez cette erreur et suggérez un moyen de la corriger pour que la fonction marche convenablement (le code n'est pas nécessaire, expliquez l'idée).

```

>> a = [2 -1 5 -3 0 1 6];
>> b = rmvNegatives(a);

function Vout = rmvNegatives(Vin)
for i=1:size(Vin,2)
    if Vin(i)<0
        Vin(i) = [];
    end
end
Vout = Vin;

```

c) (8 points) La fonction *Matlab* ci-dessous *myNR.m* est censée trouver la solution d'une quelconque fonction $F(x)=0$ en se basant sur la méthode de Newton-Raphson, pour un seuil de convergence donné, et pour un nombre maximal d'itérations. Les arguments d'entrée de cette fonction sont : (i) la poignée de fonction (function handle) de $F(x)$, (ii) la poignée de fonction *dfun* de sa première dérivée, et (iii) la valeur initiale x_0 de la solution. Remplissez les lignes marquées d'un astérisque * afin que la fonction accomplisse sa tâche.

```

function NRsol = myNR(fun,dfun,x_old)
    max_it = 10^5;
    convThresh = 10^-5;
    iterationN=0;
    diff_x = 10^3;
*           % condition of the iteration loop
*   x_new =           % update of solution
    diff_x = abs(x_new-x_old);
    x_old = x_new;
*           % update of the current iteration number

    end
    NRsol = x_new;
End

```

d) (3 points) Ecrivez un script qui forme le vecteur $[1, r^1, r^2, \dots, r^N]$ pour $N=10$ et $r=1/2$. N'utilisez pas de boucles *for* ou *while*.

e) (3 points) Considérez la fonction suivante :

```

function [F,dF] = myFunction(fH,x0,h)
    F = fH(x0);
    dF = (fH(x0+0.5*h)-fH(x0-0.5*h))/h;
end

```

Expliquez son but.

f) (3 points) Ecrivez un script qui définit une fonction anonyme pour l'expression $\sin(x)*e^{(2*x)}$ et qui lance la fonction définie en e) avec les arguments d'entrée $x_0 = 2$ and $h = 1 \times 10^{-6}$.