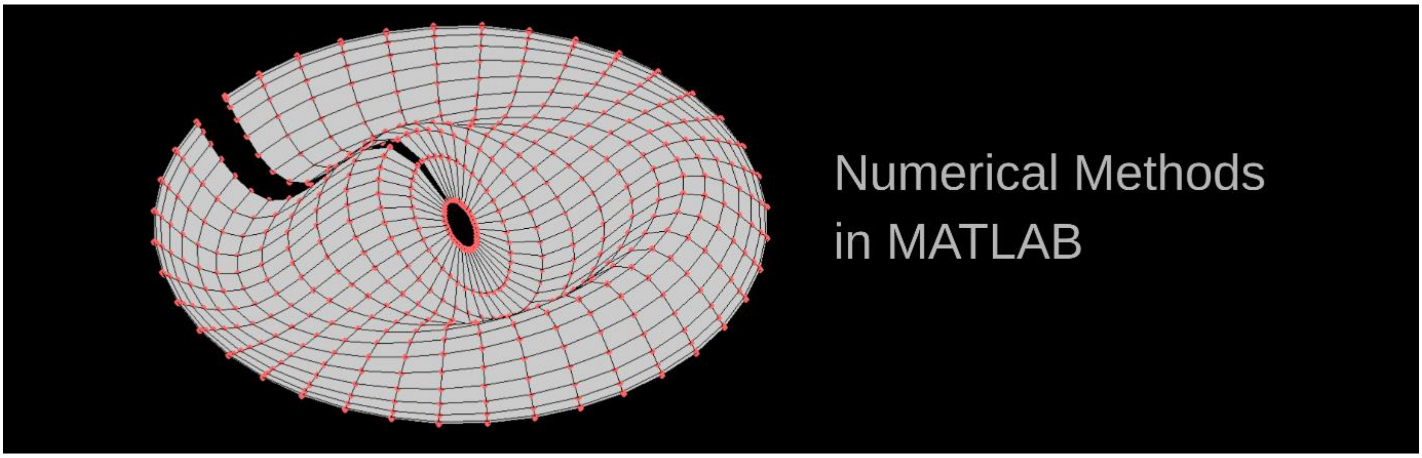
ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ & ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
Τομέας Μαθηματικών
Πολυτεχνειούπολη – Ζωγράφου ΑΘΗΝΑ - 157 80
ΤΗΛ. : 772 1774
FAX : 772 1775

ΑΡΙΘΜΗΤΙΚΗ ΑΝΑΛΥΣΗ ΙΙ : 2023-2024

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ



ΕΓΑΣΤΗΡΙΑΚΗ ΟΜΑΔΑ Α : 15.00-17.00

ΚΟΝΤΟΘΑΝΑΣΗ ΣΩΤΗΡΙΑ
ΑΜ : 09120080

Πρόβλημα Αρχικών Τιμών:

Τα προβλήματα που θα λύσετε στα ερωτήματα 1,2,3 αφορούν το ΠΑΤ του παρακάτω πίνακα όπου η παράμετρος α αντιστοιχεί στον αριθμό του τελευταίου ψηφίου του αριθμού μητρώου σας.

$$\Pi AT \quad \left\{ \begin{array}{c} y' = y^3 - (1 + \boldsymbol{a}) t exp\left(-\frac{1 + \boldsymbol{a}}{2} t^2\right) - \exp\left(-\frac{3 + 3\boldsymbol{a}}{2} t^2\right), t \in [0,2] \\ y(0) = 1 \end{array} \right\}$$

Δίνεται πραγματική λύση:

$$y(t) = \exp\left(-\frac{1 + \boldsymbol{a}}{2} t^2\right)$$

Το τελευταίο τελευταίου ψηφίου του αριθμού μητρώου μου είναι α=0 άρα :

$$\Pi AT \quad \left\{ \begin{array}{c} y' = y^3 - 1 \cdot t \cdot exp\left(-\frac{1}{2} \cdot t^2\right) - \exp\left(-\frac{3}{2} \cdot t^2\right), t \in [0,2] \\ y(0) = 1 \end{array} \right\} \quad (1)$$

Δίνεται πραγματική λύση:

$$y(t) = \exp\left(-\frac{1}{2} t^2\right)$$

Επομένως ορίζω στο MATLAB το παραπάνω ΠΑΤ:

```
%PAT ORISMOS
a=0.0; % Deksi akro p.o. tou t
b=2.0; % Aristero akro p.o. tou t
f = @(t,y) y.^3-t*exp(-0.5*t.^2)-exp(-1.5*t.^2); %y'(t)=f(t,y)
yexact = @(t) exp(-0.5*t.^2); %Pragmatikh Lysh
y0=1; %Arxikh Synthhkh
```

Ερώτημα 1:

Να κατασκευαστεί συνάρτηση που υλοποιεί μία άμεση μέθοδο Runge-Kutta που αντιστοιχεί στο ακόλουθο ταμπλό του Butcher, 4ης τάξης και 5 σταδίων

$$
\begin{array}{c|ccccc}
0 & 0 & 0 & 0 & 0 & 0 \\
1/3 & 1/3 & 0 & 0 & 0 & 0 \\
1/3 & 1/6 & 1/6 & 0 & 0 & 0 \\
1/2 & 1/8 & 0 & 3/8 & 0 & 0 \\
1 & 1/2 & 0 & -3/2 & 2 & 0 \\
\hline
 & 1/6 & 0 & 0 & 2/3 & 1/6
\end{array}
$$

και να εφαρμοστεί για (πλήθος υποδιαστημάτων της διαμέρισης) N=10, 20, 40 και 80. Για κάθε ένα από αυτά τα N να υπολογίσετε το μέγιστο απόλυτο σφάλμα καθώς και την πειραματική τάξη ακρίβειας της μεθόδου Runge-Kutta χρησιμοποιώντας τα σφάλματα που υπολογίσατε.

ΛΥΣΗ

ΜΕΘΟΔΟΙ Runge-Kutta

Αριθμητικές μέθοδοι στις οποίες πάμε από την προσέγγιση : $y_n \approx y(t_n)$ στην $y_{n+1} \approx y(t_{n+1})$ μέσω ενδιάμεσων προσέγγισεων :

$$y_{ni} \approx y(t_{ni})$$

για κάποια : $t_n \leq t_{ni} \leq t_{n+1}, \; i = 1, \dots, q \quad , \; q \in \mathbb{N}$

- Ορίζονται από το tableau του Butcher:

$$
\begin{array}{cccc|c}
a_{11} & a_{12} & \cdots & a_{1q} & \tau_1 \\
a_{21} & a_{22} & & a_{2q} & \tau_2 \\
\vdots & & \ddots & \vdots & \vdots \\
a_{q1} & a_{q2} & \cdots & a_{qq} & \tau_q \\
\hline
b_1 & b_2 & \cdots & b_q &
\end{array}
\; = \;
\begin{array}{c|c}
A & T \\
\hline
b^T &
\end{array}
$$

- Με τα τ_i κατασκευάζονται τα t_{ni} ως :

$$t_{ni} = t_n + \tau_i h \qquad (2)$$

- Με τα α_{ij}, t_{ni} υπολογίζουμε τα $k_{ni} \approx f(t_{ni}, y(t_{ni}))$:

$$k_{ni} = f\left(t_{ni}, y_n + h \sum_{j=1}^{q} a_{ij} k_{n_j}\right) \quad (3)$$

- Με τα k_{ni}, b_i υπολογίζουμε το :

$$y_{n+1} = y_n \sum_{i=1}^{q} b_i k_{n_j} \quad (4)$$

ΣΧΌΛΙΟ : Όταν ο πίνακας A που δίνεται στην υπόθεση του ερωτήματος 1 είναι **κάτω τριγωνικός με μηδενικά στην διαγώνιο** τότε έχω:

$$\alpha_{ij} = 0 \;\forall\, j > i-1 \;\Leftrightarrow \sum_{j=1}^{q} a_{ij}k_{n_j} = \sum_{j=1}^{i-1} a_{ij}k_{n_j}$$

Άρα η (3) γίνεται :

$$k_{ni} = f\left(t_{ni}, y_n + h\sum_{j=1}^{i-1} a_{ij}k_{n_j}\right)$$

όπου: k_{n1} , $k_{n2}, \dots, k_{n(i-1)}$ θα είναι ήδη γνωστά. Σε αυτήν την περίπτωση , η αντίστοιχη RK , λέγεται **Άμεση** (βλπ ερ1). Αλλιώς λέγεται **Πεπλεγμένη** ή **Έμμεση.**

Επομένως η μέθοδος Runge-Kutta που μας δόθηκε στο ερώτημα αυτό είναι Άμεση. (Explicit_RK)

- Ξεκινάμε με το να ορίσουμε την Άμεση **Runge-Kutta** στο Matlab:

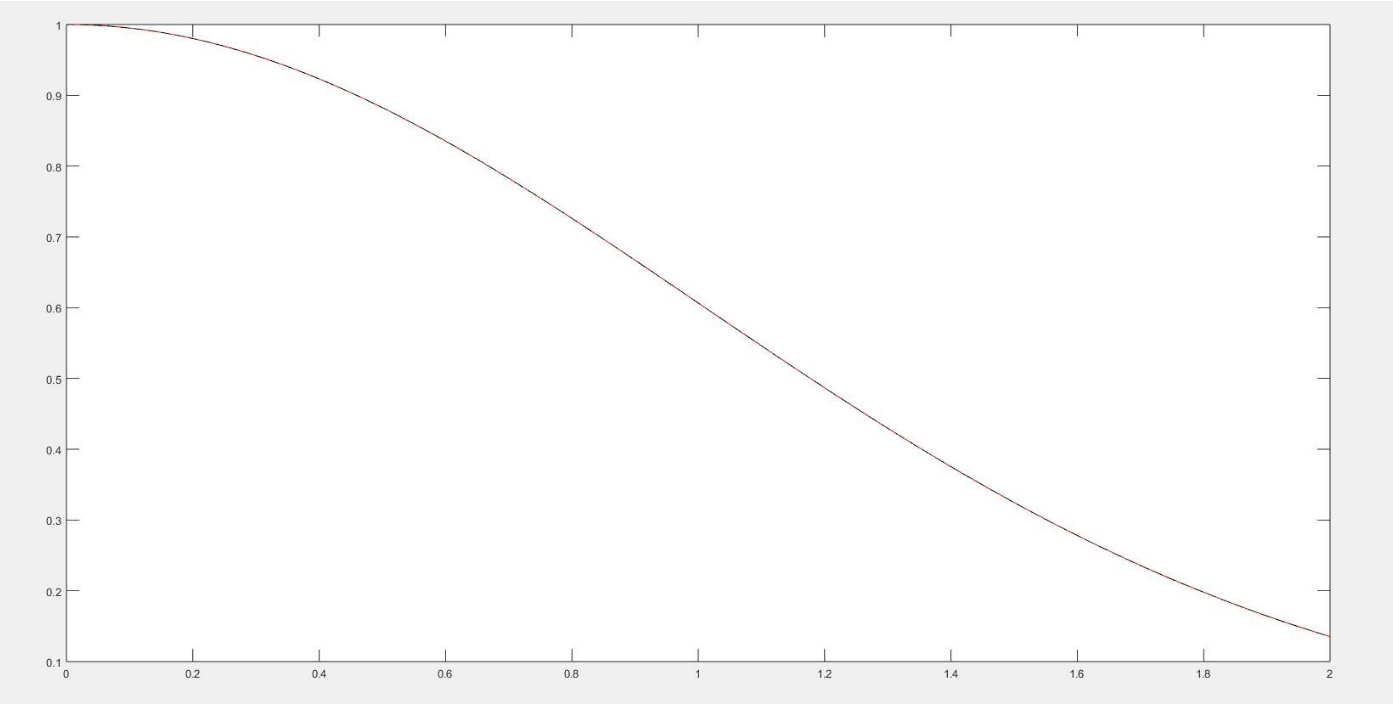
```
function sol = Explicit_RK(tinit,tend,y0,A,b,tau,N,f)

h=(tend - tinit)/N;
t=linspace(tinit,tend,N+1);

sol = zeros(1,N+1); %[0 0 0 ... 0]
sol(1)=y0;

q = length(b);
k = zeros(q,1);

    for n = 1:N

        y =sol(n);
        k(1)=f(t(n),y);

        for i=2:q

            tni = t(n) +tau(i)*h;
            k(i)=f(tni,y +h*A(i,1:i-1)*k(1:i-1));
        end

        y =y + h*b'*k;
        sol(n+1)=y;

    end
end
```

- Έπειτα ορίζουμε νέο παράθυρο στο , στο οποίο θα φορτώσουμε τα δεδομένα και με τα οποία θα τρέξουμε την συνάρτηση που ορίσαμε παραπάνω :

```
%PAT ORISMOS

tinit=0.0;% Deksi akro p.o. tou t
tend=2.0;% Aristero akro p.o. tou t
f = @(t,y) y.^3-t*exp(-0.5*t.^2)-exp(-1.5*t.^2);%y'(t)=f(t,y)
yexact = @(t) exp(-0.5*t.^2);%Pragmatikh Lysh
y0=1.0;%Arxikh Synthhkh
Ns=[10,20,40,80,100, 120, 140 ,200,250,300];

errs=zeros(length(Ns),1);
rates=zeros(length(Ns)-1,1);

%b = [0.25;0;0.75];
%tau = [0;1/3;2/3];
%A = [0,0,0;1/3,0,0;0,2/3,0];

b = [1/6;0;0;2/3;1/6];
tau = [0;1/3;1/3;1/2;1];
A = [0,0,0,0,0;1/3,0,0,0,0;1/6,1/6,0,0,0;0.125,0,0.375,0,0;0.5,0,-1.5,2,0];

for i =1:length(Ns)
    sol = Explicit_RK(tinit,tend,y0, A,b,tau,Ns(i),f);
    t = linspace(tinit, tend, Ns(i)+1);
    errs(i) = max(abs(yexact(t)-sol));
end

for i = 1:length(Ns)-1
    denom = log(Ns(i+1)/Ns(i));
    rates(i)=log( errs(i)/errs(i+1) )/denom;
end

figure(1)
plot(t, yexact(t),'r',t, sol, 'k--')

errs
rates
```

- Τα αποτελέσματα που μου δίνει ο παραπάνω κώδικας είναι τα εξής:

```
>> ergastiriaki_ergasia_Art_2023_2024_erwt1

errs =

   1.0e-04 *

    0.1888
    0.0169
    0.0021
    0.0002


rates =

    3.4835
    2.9836
    3.6493
```



ΕΡΜΗΝΕΙΑ ΑΠΟΤΕΛΕΣΜΆΤΩΝ

- Η **τάξη ακρίβειας** αυτής της μεθόδου είναι κανονικά **4**. Αλλά για τόσο μικρή διαμέριση δεν μπορεί κάτι τέτοιο να φανεί . Επομένων το διάνυσμα rates που είναι το διάνυσμα το οποίο προσεγγίζει την τάξη ακρίβειας της μεθόδου , δεν προλαβαίνει να προσεγγίσει την λύση. Αυξάνοντας την διαμέριση του διαστήματος [0,2] μπορούμε να έχουμε καλύτερη εικόνα για την τάξη ακρίβειας .Εάν παραδείγματος χάριν χρησιμοποιούσαμε αυτό το διάνυσμα διαφορετικών διαμερίσεων : Ns=[10,20,40,80,100, 120, 140 ,200,250,300] η τάξη ακρίβειας της μεθόδου θα προσεγγιζόταν πολύ καλύτερα.

- Παρατηρώ για το διάνυσμα των σφαλμάτων ότι όσο αυξάνεται η διαμέριση του διαστήματος [0,2] τόσο το σφάλμα της προσέγγισης που υπολογίζει η Άμεση μέθοδος μειώνεται.

- Βλέπουμε στο παραπάνω διάγραμμα την γραφική παράσταση της πραγματικής λύσης με την γραφική παράσταση της πειραματικής λύσης να ταυτίζονται, πράγμα πολύ ευχάριστο γιατί βλέπουμε ότι η μέθοδος μας δουλεύει πολύ καλά.

Ερώτημα 2:

Να κατασκευαστεί η έμμεση μέθοδος Runge-Kutta 4 σταδίων Gauss-Radau, που αντιστοιχεί στο ταμπλό του Butcher

$$
\begin{array}{c|cc}
\frac{1}{3} & \frac{5}{12} & \frac{-1}{12} \\
1 & \frac{3}{4} & \frac{1}{4} \\
\hline
 & 3/4 & 1/4
\end{array}
$$

και να εφαρμοστεί στο πρόβλημα αρχικών τιμών (1),για (πλήθος υποδιαστημάτων της διαμέρισης) N=10, 20, 40 και 80. Για κάθε ένα από αυτά τα N υπολογίσετε το μέγιστο απόλυτο σφάλμα καθώς και την πειραματική τάξη ακρίβειας της μεθόδου Runge-Kutta χρησιμοποιώντας τα σφάλματα που υπολογίσατε. Η επίλυση του μη-γραμμικού συστήματος να γίνει μέσω της μεθόδου σταθερού σημείου για Nfp=2 και <u>7 αριθμό επαναλήψεων</u>. Τι παρατηρείτε ως προς τα σφάλματα που προκύπτουν και την τάξη ακρίβειας της μεθόδου για τιμές Nfp=2 και 7;

ΛΥΣΗ

Σύμφωνα με το θεωρητικό υπόβαθρο που δόθηκε στο προηγούμενο ερώτημα τώρα δεν έχω κάτω τριγωνικό πίνακα A και άρα η μέθοδος Runge-Kutta που μας δόθηκε στο ερώτημα αυτό είναι Έμμεση. (Implicit_RK)

- Ξεκινάμε με το να ορίσουμε την Άμεση **Runge-Kutta** στο Matlab:

```
%Implicit_RK
function sol = Implicit_RK_FP(tinit,tend,y0,A,b,tau,N,f, maxits)

h=(tend - tinit)/N;
t=linspace(tinit,tend,N+1);
sol =zeros(1,N+1); %[0 0 0 ... 0]
sol(1)=y0;

%q = length(b);
k = ones(2,1);

    for n = 1:N
        y =sol(n);
   %----------------------------------------------
        for i=1:2
            tni = t(n) +tau(i)*h;
           %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
           %Poses fores ta treksei i loopa wste na dw pou syglinei
           %to k mas deixnei to m
            for m=1:maxits

                for j=1:2 %ypologizodai ola ta k se kathe vhma tis loopas
                k(j)=f(tni,y +h*A(j,1:2)*k(1:2));
                end

            end
           %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
         end
    %----------------------------------------------
        y =y + h*b'*k;
```

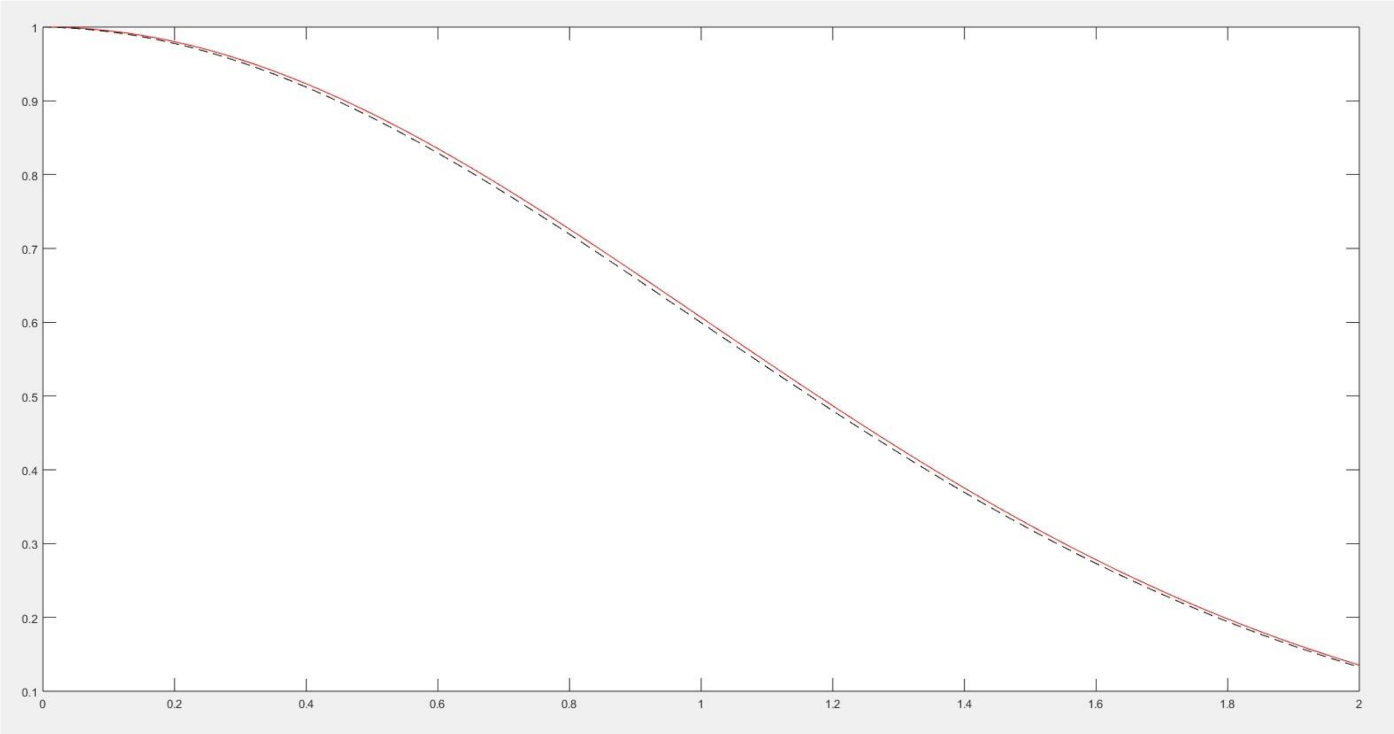
```
        sol(n+1)=y;
    end
end
```

```
%PAT ORISMOS
tinit=0.0;% Deksi akro p.o. tou t
tend=2.0;% Aristero akro p.o. tou t
f = @(t,y) y.^3-t*exp(-0.5*t.^2)-exp(-1.5*t.^2);%y'(t)=f(t,y)
yexact = @(t) exp(-0.5*t.^2);%Pragmatikh Lysh
y0=1.0;%Arxikh Synthhkh
Ns=[10,20,40,80];

maxits=7;

errs=zeros(length(Ns),1);
rates=zeros(length(Ns)-1,1);



b = [0.75;0.25];
tau = [1/3;1];
A = [5/12,-1/12;0.75,0.25];

for i =1:length(Ns)
    sol = Implicit_RK_FP(tinit,tend,y0, A,b,tau,Ns(i),f,maxits);
    t = linspace(tinit, tend, Ns(i)+1);
    errs(i) = max(abs(yexact(t)-sol));
end

for i = 1:length(Ns)-1
    denom = log(Ns(i+1)/Ns(i));
    rates(i)=log( errs(i)/errs(i+1) )/denom;
end

figure(1)
plot(t, yexact(t),'r',t, sol, 'k--')


errs
rates
```

```
errs =

    0.0356
    0.0239
    0.0135
    0.0072


rates =

    0.5760
    0.8192
    0.9161
```

```
%PAT ORISMOS
tinit=0.0;% Deksi akro p.o. tou t
tend=2.0;% Aristero akro p.o. tou t
f = @(t,y) y.^3-t*exp(-0.5*t.^2)-exp(-1.5*t.^2);%y'(t)=f(t,y)
yexact = @(t) exp(-0.5*t.^2);%Pragmatikh Lysh
y0=1.0;%Arxikh Synthhkh
Ns=[10,20,40,80];

maxits=2;

errs=zeros(length(Ns),1);
rates=zeros(length(Ns)-1,1);



b = [0.75;0.25];
tau = [1/3;1];
A = [5/12,-1/12;0.75,0.25];

for i =1:length(Ns)
    sol = Implicit_RK_FP(tinit,tend,y0, A,b,tau,Ns(i),f,maxits);
    t = linspace(tinit, tend, Ns(i)+1);
    errs(i) = max(abs(yexact(t)-sol));
end

for i = 1:length(Ns)-1
    denom = log(Ns(i+1)/Ns(i));
    rates(i)=log( errs(i)/errs(i+1) )/denom;
end

figure(1)
plot(t, yexact(t),'r',t, sol, 'k--')


errs
rates
```

```
>> ergastiriaki_ergasia_Art_2023_2024_erwt2

errs =

    0.0333
    0.0235
    0.0135
    0.0072


rates =

    0.5049
    0.8015
    0.9116
```



ΕΡΜΗΝΕΙΑ ΑΠΟΤΕΛΕΣΜΆΤΩΝ

- Η **τάξη ακρίβειας** αυτής της μεθόδου συγκλίνει στο **1 είτε το maxits=2 είτε το maxits=7**.
- Με την βοήθεια της γραφικής παράστασης της πραγματικής λύσης μαζί με την προσεγγιστική λύση βλέπουμε πως η αριθμητικής μας μέθοδος προσεγγίζει καλά προσεγγιστική λύση μας, οι δύο γραφικές ταυτίζονται, .
- Παρατηρώ για το διάνυσμα των σφαλμάτων ότι όσο αυξάνεται η διαμέριση του διαστήματος [0,2] τόσο το σφάλμα της προσέγγισης που υπολογίζει η Άμεση μέθοδος μειώνεται.

Ερώτημα 3:

Να κατασκευαστεί η πολυβηματική μέθοδος BDF3:

$$\mathbf{11}\boldsymbol{y}_{\boldsymbol{n+3}} - \mathbf{18}\boldsymbol{y}_{\boldsymbol{n+2}} + \mathbf{9}\boldsymbol{y}_{\boldsymbol{n+1}} - \mathbf{2}\boldsymbol{y}_{\boldsymbol{n}} = \mathbf{6}\boldsymbol{h}\boldsymbol{f}_{\boldsymbol{n+3}}$$

όπου , για (πλήθος υποδιαστημάτων της διαμέρισης) N=20, 40, 80 και 160. Για κάθε ένα από αυτά τα N υπολογίστε το μέγιστο απόλυτο σφάλμα καθώς και την πειραματική τάξη ακρίβειας της μεθόδου BDF3 χρησιμοποιώντας τα σφάλματα που υπολογίσατε. Για τον υπολογισμό κατάλληλων αρχικών συνθηκών να χρησιμοποιηθεί η έμμεση μέθοδος του τραπεζίου, ενώ για την επίλυση της μη-γραμμικής εξίσωσης, τόσο στην BDF3 όσο και στην έμμεση μέθοδο του τραπεζίου, να χρησιμοποιηθεί η μέθοδος Newton-Raphson, με αριθμό επαναλήψεων Nnr=3. Τι παρατηρείτε;

ΛΥΣΗ

Κατασκευή πολυβηματικής μεθόδου BDF3:

```
function sol = BDF3(a,b,y0,N,f,Df,maxits1,maxits2)

    h=(b-a)/N;
    t=linspace(a,b,N+1);
    sol=zeros(1,N+1);
    sol(1)=y0;

    %$$$$$$$$$$$$$ IMPLICIT TRAPEZIOU $$$$$$$$$$$$$$$$$
    for n=1:2
        yold = sol(n);
        ynew = sol(n);
        for k = 1:maxits1
           f12=f(t(n),yold)+f(t(n+1),ynew);
           ynew=yold + 0.5*h*f12;
           yold = ynew;
        end
        sol(n+1)= ynew;
    end
    %$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

    for n = 1:N-2
        Y0=sol(n);
        Y1=sol(n+1);
        Y2=sol(n+2);
        Y3=sol(n+2);

        %$$$$$$$$$$$$$ NEWTON RAPHSON $$$$$$$$$$$$$$$$$$$$
        for m = 1:maxits2
            f3 = f(t(n+3),Y3);
             g = 11*Y3 - 18*Y2 + 9*Y1 - 2*Y0 -6*h*f3;
            Dg = 11.0 - 6*h*Df(t(n+3),Y3);
            Y3 = Y3 - g/Dg;
        end
        %$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

        sol(n+3) = Y3;
    end

end
```

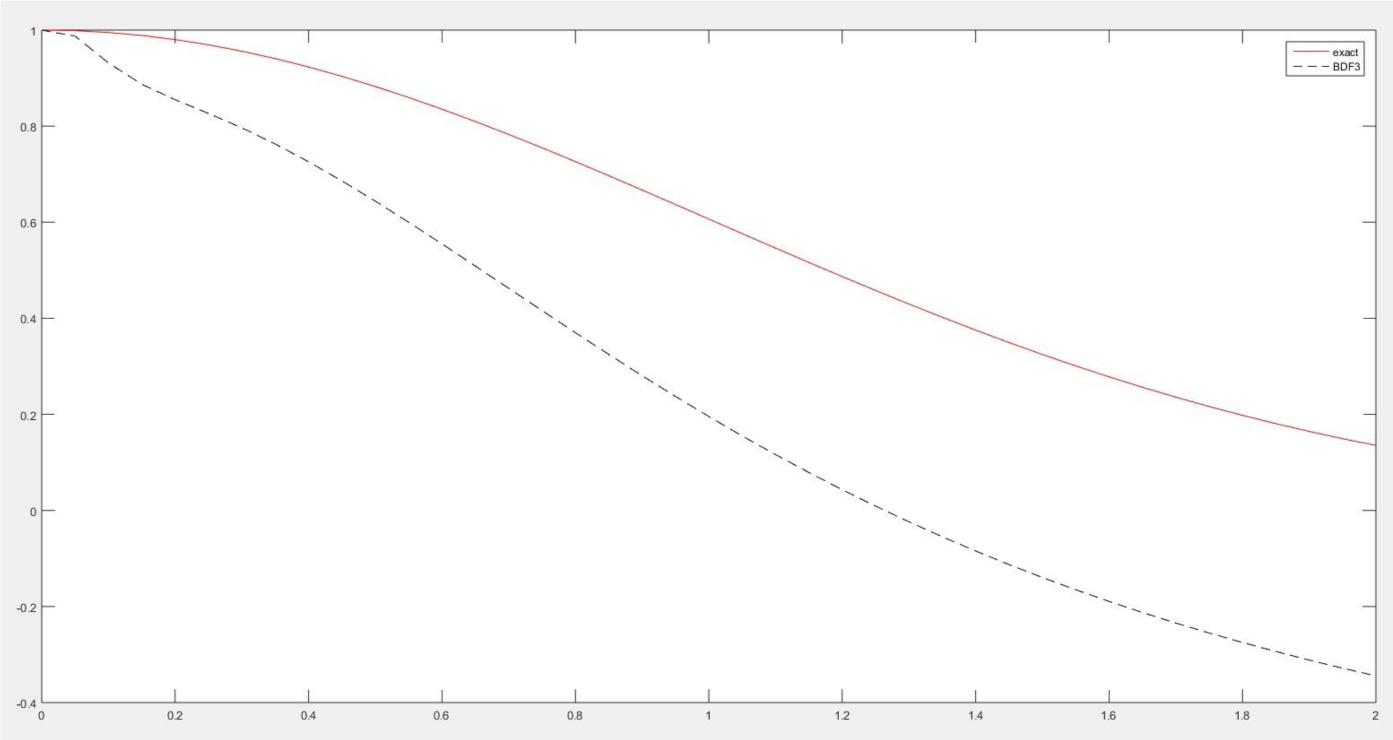
Έπειτα ορίζουμε νέο παράθυρο στο , στο οποίο θα φορτώσουμε τα δεδομένα και με τα οποία θα τρέξουμε την συνάρτηση που ορίσαμε παραπάνω :

```
a = 0.0;% Deksi akro p.o. tou t
b = 2.0;% Aristero akro p.o. tou t
y0 = 1.0;%Arxikh Synthhkh

%Ns=[20,40,80,160];
Ns=[10 , 20, 40,80,160];

f = @(t,y) y.^3-t*exp(-0.5*t.^2)-exp(-1.5*t.^2);%y'(t)=f(t,y)
Df = @(t,y) 3*y.^2;

yexact = @(t) exp(-0.5*t.^2);%Pragmatikh Lysh
maxits1=7;
maxits2=3;


errsBDF3=zeros(length(Ns),1);
ratesBDF3=zeros(length(Ns)-1,1);

for i =1:length(Ns)

    solBDF3 = BDF3(a,b,y0,Ns(i),f,Df,maxits1,maxits2);
    t = linspace(a, b, Ns(i)+1);
    errsBDF3(i) = max(abs(yexact(t)-solBDF3));

end

for i = 1:length(Ns)-1
   denom=log(Ns(i+1)/Ns(i));
   ratesBDF3(i)=log(errsBDF3(i)/errsBDF3(i+1))/denom;
end

errsBDF3
ratesBDF3

figure(1)% gia na mas dwsei ayta poy grapsame se m;ia eikona
plot( t, yexact(t),'r', t, solBDF3, 'k--')
%mou deixnei se legada poia einai poia
legend('exact','BDF3')
```

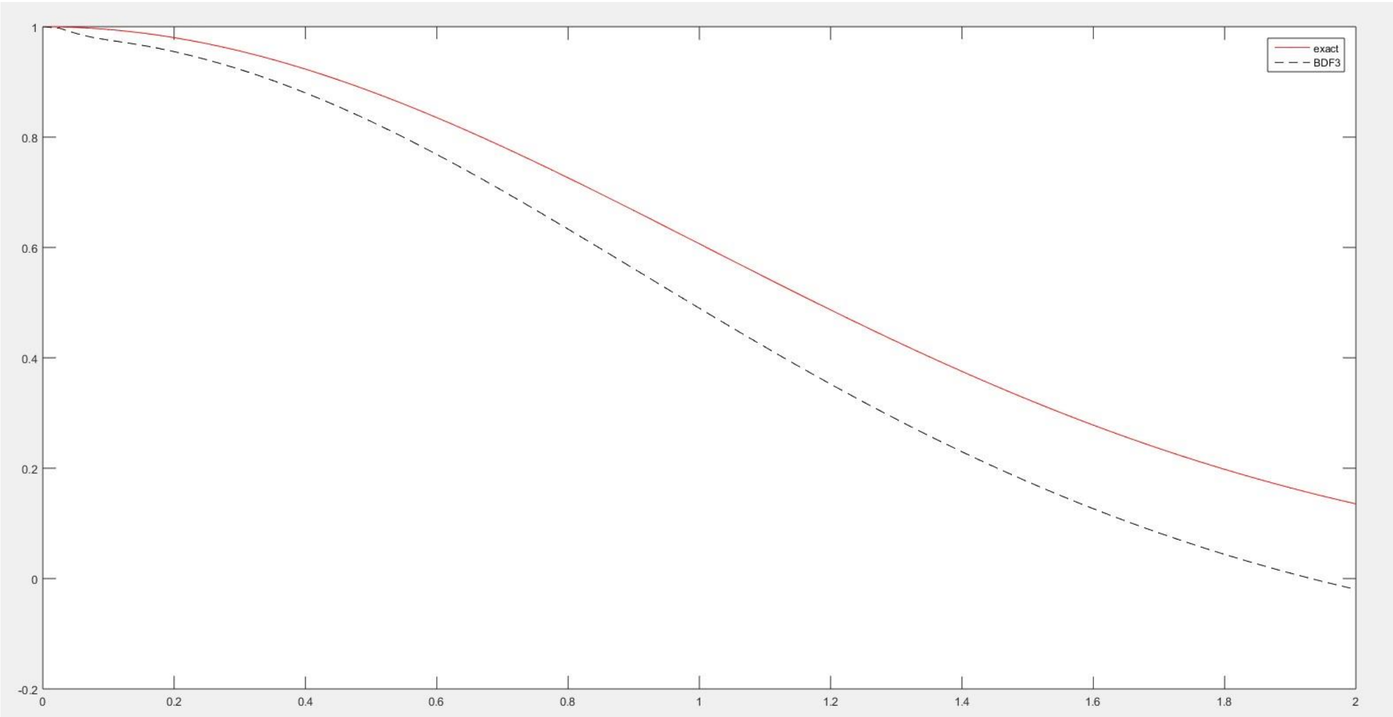
```
>> ergastiriaki_ergasia_Art_2023_2024_erwt3

errsBDF3 =

   17.4084
   91.0446
    0.4799
    0.1550
    0.0420


ratesBDF3 =

   -2.3868
    7.5677
```

```
        1.6305
        1.8843
```

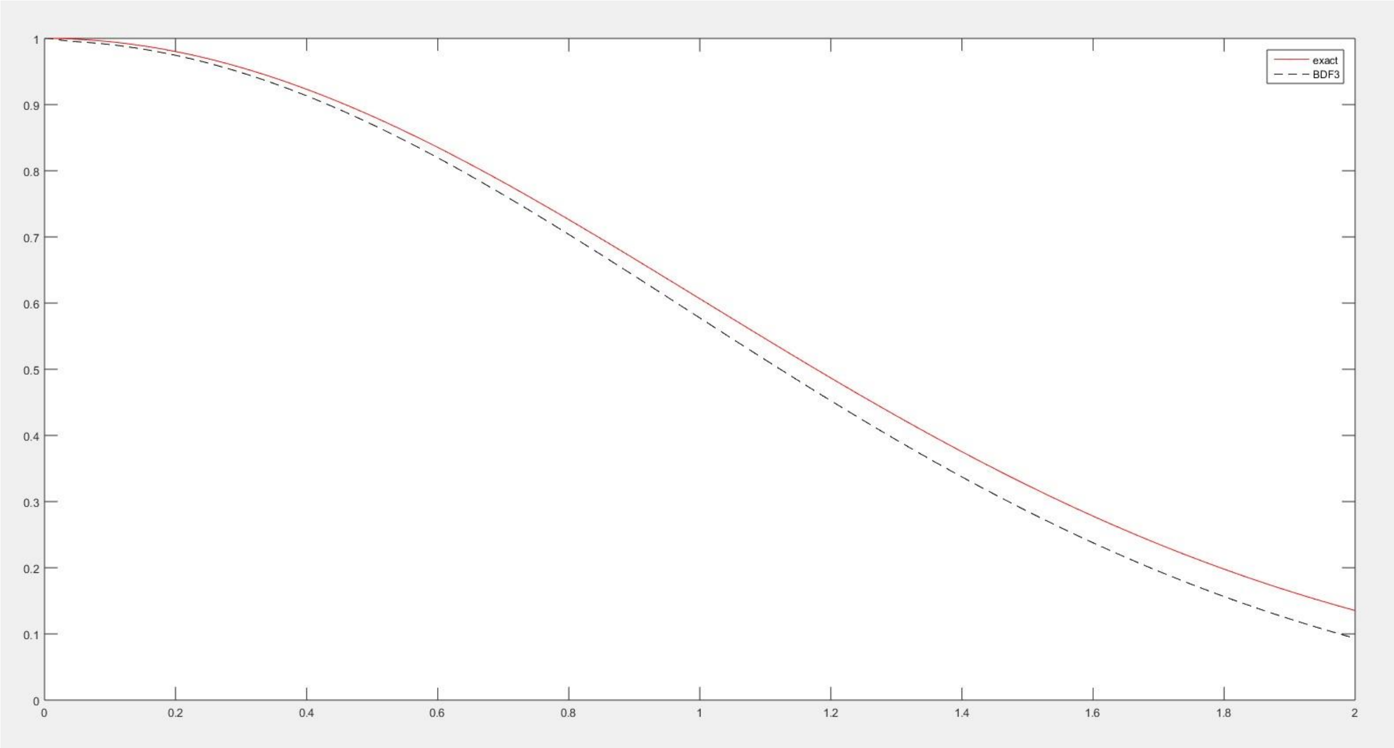
N=40



N=80



N=160



ΕΡΜΗΝΕΙΑ ΑΠΟΤΕΛΕΣΜΆΤΩΝ

- Το διάνυσμα errsBDF3 δίνει τα αποτελέσματα του μέγιστου απόλυτου σφάλματος για τις διαφορετικές διαμερίσεις οι οποίες ορίζονται με το διάνυσμα Ns=[20,40,80,160]. Παρατηρούμε ότι όσο οι διαμερίσεις αυξάνονται , δηλαδή η τιμή του h μικραίνει, τόσο το σφάλμα που έχει η αριθμητική προσέγγιση από την πραγματική λύση συνεχώς μειώνεται και άρα τόσο πιο κοντά έρχεται η προσεγγιστική λύση στην πραγματική . Αυτό φαίνεται επειδή η μέγιστη απολυτή τιμή του σφάλματος μικραίνει σταδιακά , με την αύξηση της διαμέρισης.

- Μπορούμε ακόμη να δούμε και από τις διαφορετικές γραφικές παραστάσεις των διαφορετικών διαμερίσεων πως προσεγγίζουμε την πραγματική λύση, όσο μεγαλώνει η διαμέριση.

- Με την βοήθεια της γραφικής παράστασης της πραγματικής λύσης (N=160) μαζί με την προσεγγιστική λύση βλέπουμε πως η αριθμητικής μας μέθοδος προσεγγίζει σχετικά καλά προσεγγιστική λύση μας, σχεδόν οι δύο γραφικές ταυτίζονται, .

- Η τάξη ακρίβειας φαίνεται να είναι κοντά στο 2 (το οποίο θα φανεί άπαξ και αυξήσουμε ακόμη περισσότερο το μήκος της διαμέρισης), συγκλίνει στο 2. Λόγω ότι η αριθμητική μέθοδος που χρησιμοποιήσαμε δεν είναι η ιδανική, η τάξη ακρίβειας που προκύπτει δεν είναι η αυτή που περιμέναμε.

Ερώτημα 4:

Δίνεται το ακόλουθο πρόβλημα αρχικών τιμών

$$\boldsymbol{u'(t) = 998u(t) + 2023v(t)}$$
$$\boldsymbol{v'(t) = -1001u(t) - 2024v(t)}$$
$$\boldsymbol{u(0) = 1\,, v(0) = 1}$$

A. Να λύσετε το πρόβλημα χρησιμοποιώντας την μέθοδο Runge-Kutta που αντιστοιχεί στο ταμπλό του Butcher του Ερωτήματος 1, με ομοιόμορφη διαμέριση, στο διάστημα [0,1] με h=0.1, h=0.01, h=0.001. (ερ 6)

ΛΥΣΗ

$$\left\{ \begin{array}{c} \begin{pmatrix} u'(t) \\ v'(t) \end{pmatrix} = \begin{pmatrix} \mathbf{988} & \mathbf{2023} \\ \mathbf{-1001} & \mathbf{-2024} \end{pmatrix} \cdot \begin{pmatrix} u(t) \\ v(t) \end{pmatrix}, t \in (0,1] \\ \begin{pmatrix} u(0) \\ v(0) \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \end{array} \right\} \Leftrightarrow \left\{ \begin{array}{cc} y'(t) = \boldsymbol{A} \cdot y(t), & t \in (0,1] \\ y(0) = y_0 & \end{array} \right\}$$

Όπου: $A = \begin{pmatrix} 988 & 2023 \\ -1001 & -2024 \end{pmatrix}$, $y_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

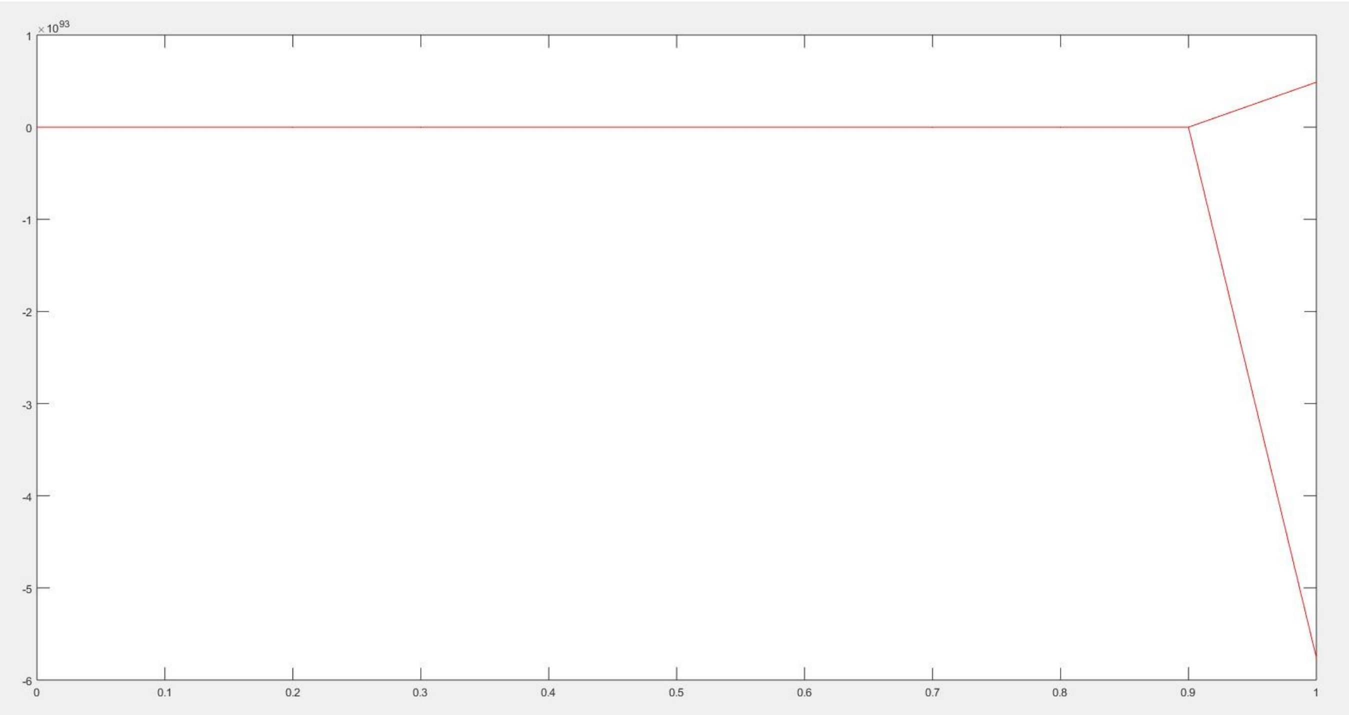
Κατασκευή μέθοδο Runge-Kutta για επίλυση συστήματος διαφορικών εξισώσεων :

```
function sol = RK_SYS(tinit,tend,Y0,N,AA,b,tau,F)
h=(tend-tinit)/N;
t=linspace(tinit,tend,N+1);
d=length(Y0);


sol = zeros(d,N+1);
sol(:,1)= Y0 ;
q= length(b);
k = zeros(d,q);
%Y=Y0; % <-----
for n= 1:N
     Y = sol(:,n);
     k(:,1)=F(t(n),Y);

    for i=2:q
        tni=t(n)+h*tau(i);
        AAk= k(:,1:i-1)*AA(i,1:i-1)';
        k(:,i)=F(tni,Y+h*AAk);
    end
    Y = Y + h*k*b;
    sol(:,n+1)= Y;
end
end
```

Έπειτα ορίζουμε νέο παράθυρο στο , στο οποίο θα φορτώσουμε τα δεδομένα και με τα οποία θα τρέξουμε την συνάρτηση που ορίσαμε παραπάνω.

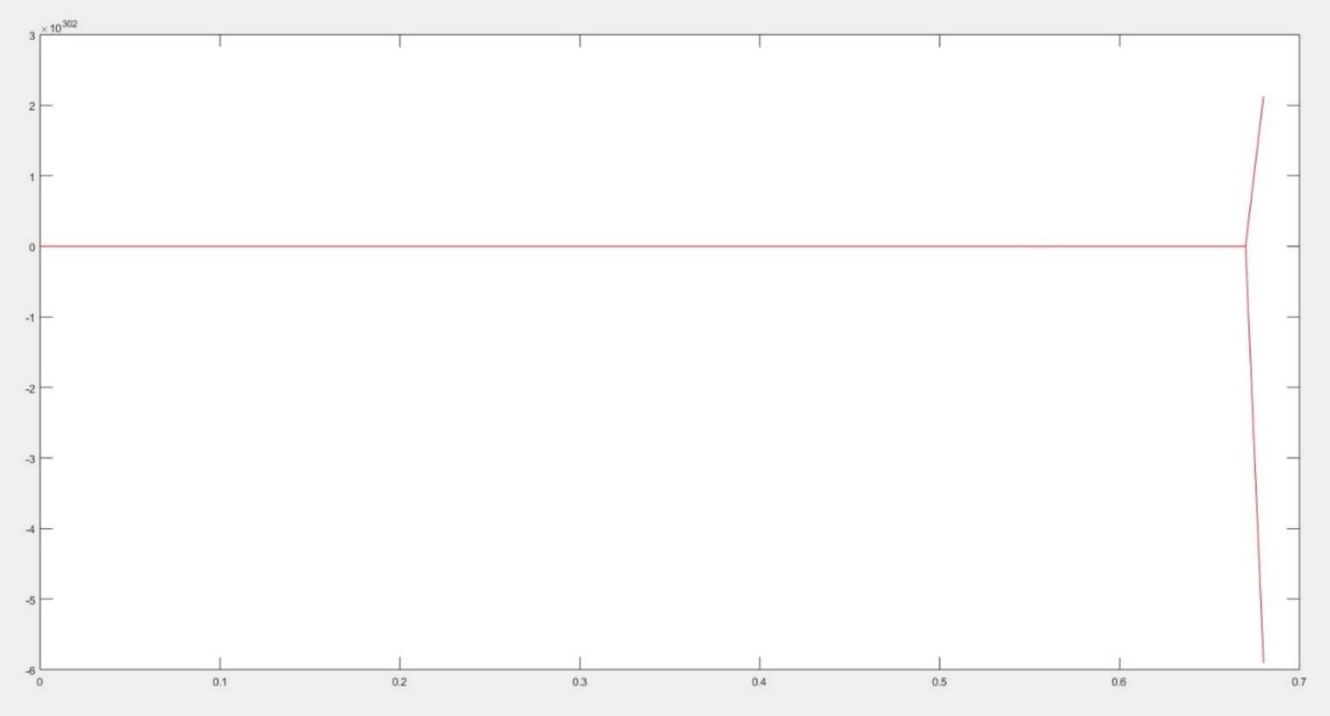
- **Υλοποίηση γίνεται για h = 0.1**

```
%$$$$$$$$ ORISMOS PAT : SYSTIMA SDE $$$$$$$$$$$$$$$$$$$$$$$
tinit = 0.0; tend = 1.0;
Y0 = [1;1];
A=[988,2023;-1001,2024];
F = @(t,Y) A*Y;
JF = @(t,Y) A;
%Yexact = @(t)[sin(t); cos(t)]; <- MALLON DEN TA XREIASTEI

%$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

N=10;

t=linspace(tinit,tend,N+1);

%$$$$$$$$$$$$$$$$$$ DEDOMENA IA BUTCER TABLEAU $$$$$$$$$$$$$$$$$$$$$$$$$$$$$
b = [1/6;0;0;2/3;1/6];
tau = [0;1/3;1/3;1/2;1];
AA = [0,0,0,0,0;1/3,0,0,0,0;1/6,1/6,0,0,0;0.125,0,0.375,0,0;0.5,0,-1.5,2,0];
%$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

solRK=RK_SYS(tinit,tend,Y0,N,AA,b,tau,F);

plot(t,solRK,'r')
```



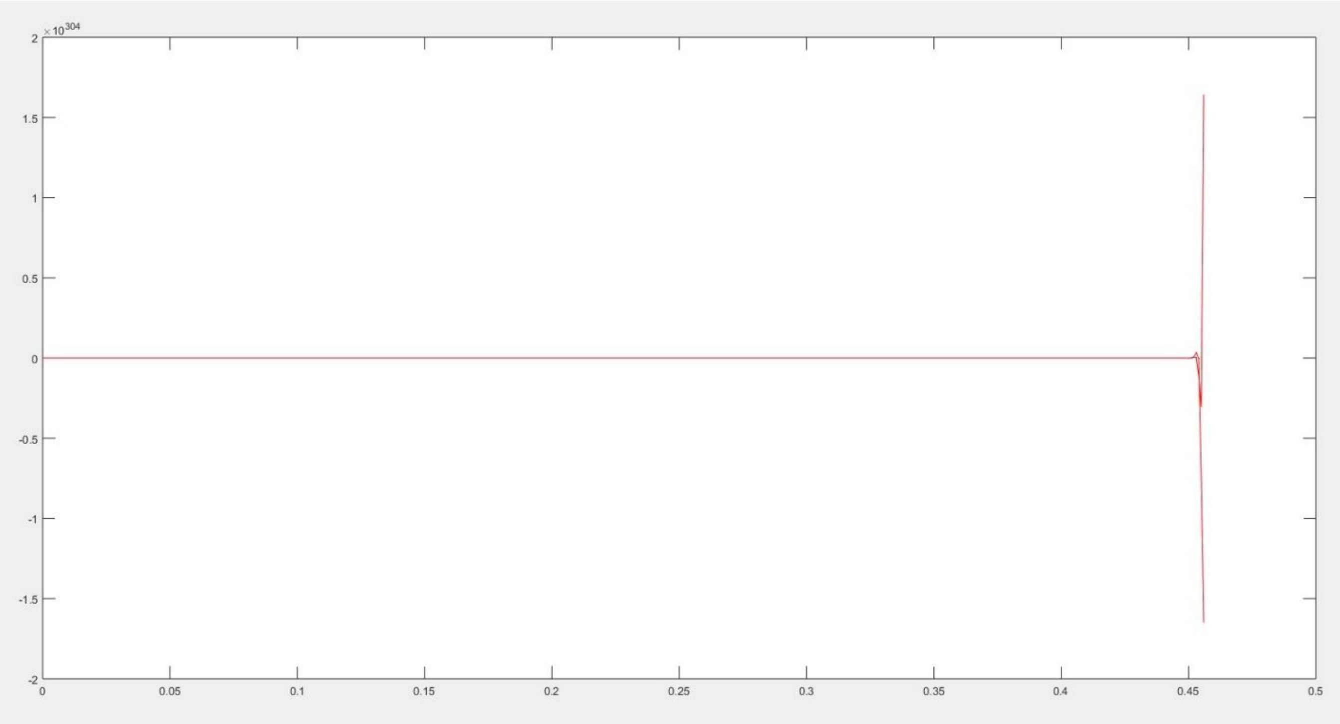
- **Υλοποίηση γίνεται για h = 0.01**

```
%$$$$$$$$$ ORISMOS PAT : SYSTIMA SDE $$$$$$$$$$$$$$$$$$$$$$$
tinit = 0.0; tend = 1.0;
Y0 = [1;1];
A=[988,2023;-1001,2024];
F = @(t,Y) A*Y;
JF = @(t,Y) A;
%Yexact = @(t)[sin(t); cos(t)]; <- MALLON DEN TA XREIASTEI

%$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

N=100;

t=linspace(tinit,tend,N+1);

%$$$$$$$$$$$$$$$$$$ DEDOMENA IA BUTCER TABLEAU $$$$$$$$$$$$$$$$$$$$$$$$$$$$$
b = [1/6;0;0;2/3;1/6];
tau = [0;1/3;1/3;1/2;1];
AA = [0,0,0,0,0;1/3,0,0,0,0;1/6,1/6,0,0,0;0.125,0,0.375,0,0;0.5,0,-1.5,2,0];
%$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

solRK=RK_SYS(tinit,tend,Y0,N,AA,b,tau,F);

plot(t,solRK,'r')
```



- **Υλοποίηση γίνεται για h = 0.001**

```
%$$$$$$$$ ORISMOS PAT : SYSTIMA SDE $$$$$$$$$$$$$$$$$$$$$$
tinit = 0.0; tend = 1.0;
Y0 = [1;1];
A=[988,2023;-1001,2024];
F = @(t,Y) A*Y;
JF = @(t,Y) A;
%Yexact = @(t)[sin(t); cos(t)]; <- MALLON DEN TA XREIASTEI

%$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

N=1000;

t=linspace(tinit,tend,N+1);

%$$$$$$$$$$$$$$$$$ DEDOMENA IA BUTCER TABLEAU $$$$$$$$$$$$$$$$$$$$$$$$$$$$$
b = [1/6;0;0;2/3;1/6];
tau = [0;1/3;1/3;1/2;1];
AA = [0,0,0,0,0;1/3,0,0,0,0;1/6,1/6,0,0,0;0.125,0,0.375,0,0;0.5,0,-1.5,2,0];
%$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

solRK=RK_SYS(tinit,tend,Y0,N,AA,b,tau,F);

plot(t,solRK,'r')
```



ΕΡΜΗΝΕΙΑ ΑΠΟΤΕΛΕΣΜΆΤΩΝ

- Παρατηρούμε ότι όσο οι διαμερίσεις αυξάνονται , δηλαδή η τιμή του h μικραίνει, τόσο το σφάλμα που έχει η αριθμητική προσέγγιση από την πραγματική λύση συνεχώς μειώνεται και άρα τόσο πιο κοντά έρχεται η προσεγγιστική λύση στην πραγματική. Αυτό φαίνεται από την όλο και μεγαλύτερη συσσώρευση των γραμμών σε ένα σημείο στις γραφικές παραστάσεις.

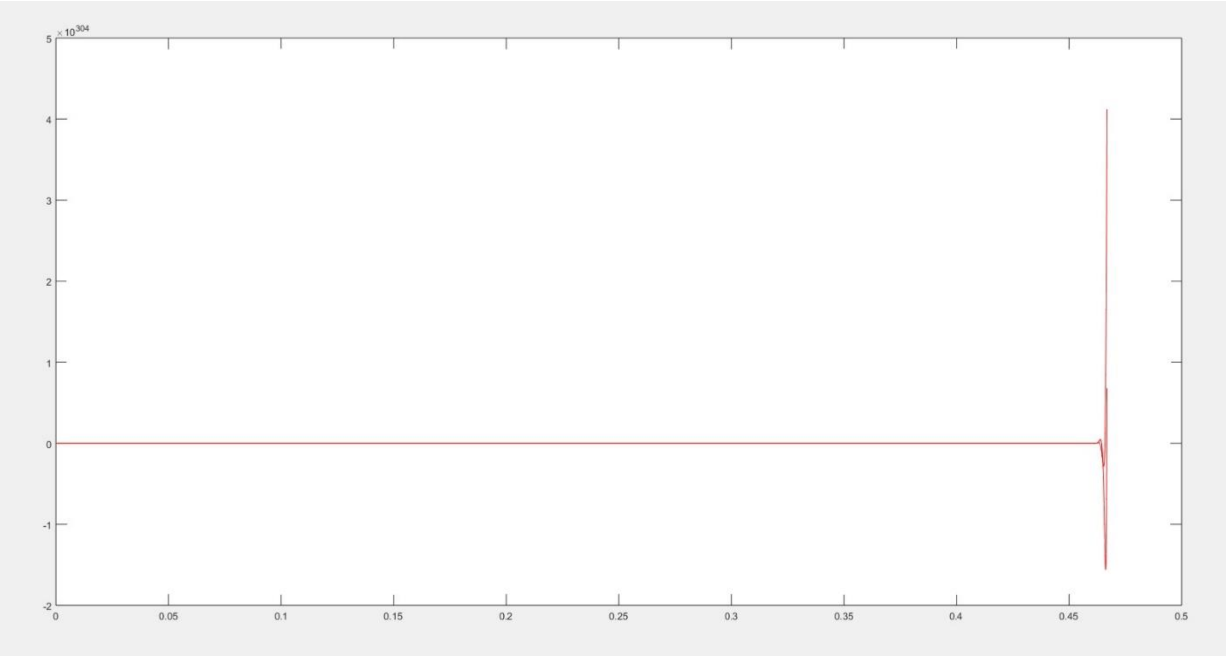
B. Να λύσετε το πρόβλημα με h=0.0001 με την έμμεση μέθοδο του τραπεζίου με ομοιόμορφη διαμέριση και να θεωρήσετε αυτό το αποτέλεσμα ως την πραγματική λύση.

```
function sol = ImplTrap_sys(a,b,Y0,N,F,JF,maxits)

    h=(b-a)/N;
    t=linspace(a,b,N+1);
    d=length(Y0);
    sol=zeros(d,N+1);
    sol(:,1)=Y0;

    for n=1:N
        Yold = sol(:,n);%TIM? LYS?S TOY PRO?OYMENOY B?MATOS
        Ynew = sol(:,n);%TREXOYSA EKTIMWMEN? TIM? LYS?S

        for k=1:maxits %Mesa se ayto to loop yold paramenei stathero
                       %enw ynew allazei synexws maxits fores
            F12=f(t(n),Yold)+F(t(n+1),Ynew);

            g  = Ynew-Yold - 0.5*h*F12;
            Jg = eye(d) - 0.5*h*Df(t(n+1),Ynew);

            Ynew = Ynew-inv(Jg)*g;
        end
        sol(:,n+1)=Ynew;
    end


end
```

Έπειτα ορίζουμε νέο παράθυρο στο , στο οποίο θα φορτώσουμε τα δεδομένα και με τα οποία θα τρέξουμε την συνάρτηση που ορίσαμε παραπάνω :

Υλοποίηση για h=0.0001

```
%$$$$$$$$ ORISMOS PAT : SYSTIMA SDE $$$$$$$$$$$$$$$$$$$$$$
tinit = 0.0; tend = 1.0;
Y0 = [1;1];
A=[988,2023;-1001,2024];
F = @(t,Y) A*Y;
JF = @(t,Y) A;
%Yexact = @(t)[sin(t); cos(t)]; <- MALLON DEN TA XREIASTEI
maxits =5;
%$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

N=10000;
t=linspace(tinit,tend,N+1);

solImplTrap_1 = ImplTrap_sys(tinit,tend,Y0,N,F,JF,maxits);

plot(t,solImplTrap_1,'r')
```

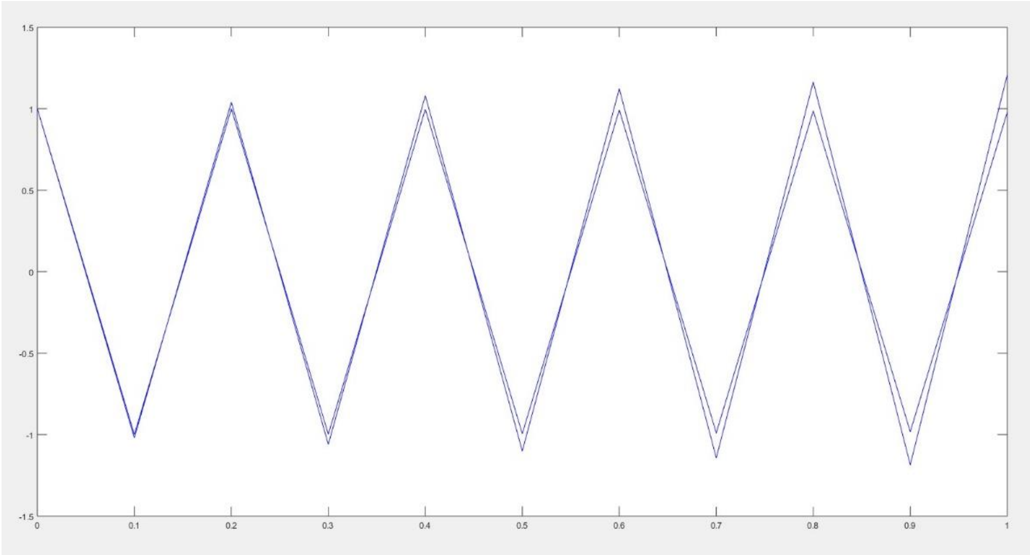


Θεωρώ **solImplTrap_1** ως πραγματική λύση.

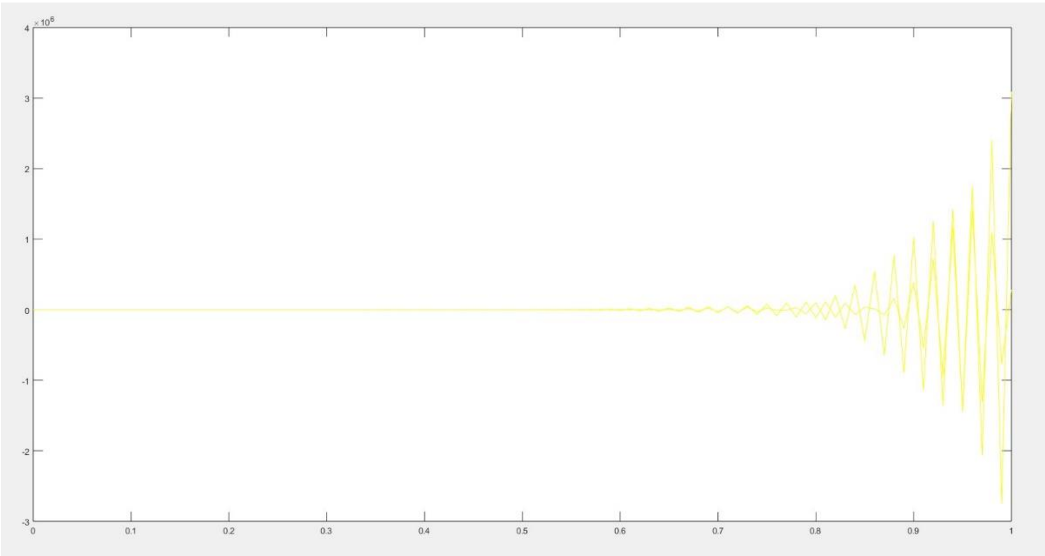
C. Να λύσετε το πρόβλημα με την έμμεση μέθοδο του τραπεζίου με ομοιόμορφη διαμέριση για h=0.1, h=0.01 και h=0.001. Να συγκρίνετε τα αποτελέσματα ερωτήματος A) με τα αποτελέσματα του ερωτήματος B) και Γ). Τι παρατηρείτε;

```
%$$$$$$$$ ORISMOS PAT : SYSTIMA SDE $$$$$$$$$$$$$$$$$$$$$$$
tinit = 0.0; tend = 1.0;
Y0 = [1;1];
A=[988,2023;-1001,2024];
F = @(t,Y) A*Y;
JF = @(t,Y) A;
%Yexact = @(t)[sin(t); cos(t)]; <- MALLON DEN TA XREIASTEI
maxits =5;
%$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
N1=10000;
N2=10;
N3=100;
N4=1000;

t1=linspace(tinit,tend,N1+1);
t2=linspace(tinit,tend,N2+1);
t3=linspace(tinit,tend,N3+1);
t4=linspace(tinit,tend,N4+1);

solImplTrap_1 = ImplTrap_sys(tinit,tend,Y0,N1,F,JF,maxits); %h=0.0001
solImplTrap_2 = ImplTrap_sys(tinit,tend,Y0,N2,F,JF,maxits);   %h=0.1
solImplTrap_3 = ImplTrap_sys(tinit,tend,Y0,N3,F,JF,maxits);  %h=0.01
solImplTrap_4 = ImplTrap_sys(tinit,tend,Y0,N4,F,JF,maxits); %h=0.001


figure(1)
plot(t1,solImplTrap_1,'r')
figure(2)
plot(t2,solImplTrap_2,'b')
figure(3)
plot(t3,solImplTrap_3,'y')
figure(4)
plot(t4,solImplTrap_4,'g')
```

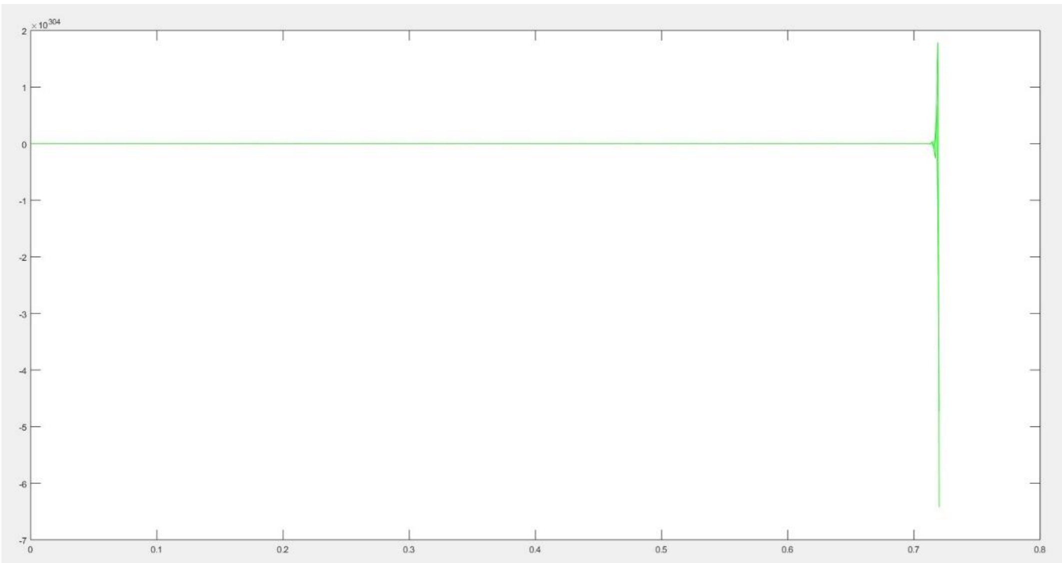
figure(2)



figure(3)



figure(4)



ΕΡΜΗΝΕΙΑ ΑΠΟΤΕΛΕΣΜΆΤΩΝ

- Παρατηρούμε ότι όσο οι διαμερίσεις αυξάνονται , δηλαδή η τιμή του h μικραίνει, τόσο το σφάλμα που έχει η αριθμητική προσέγγιση από την πραγματική λύση συνεχώς μειώνεται και άρα τόσο πιο κοντά έρχεται η προσεγγιστική λύση στην πραγματική. Συγκεκριμένα, βλέπουμε τις γραφικές παραστάσεις από μία απλή ταλάντωση που δεν δίνει κάποια σαφής τιμή λύσης στο figure(2) , όσο αυξάνονται οι διαμερίσεις να συσσωρεύεται οι πολλαπλές ταλαντώσεις γύρω από την λύση ενώ όσο απομακρυνόμαστε από την λύση οι ταλαντώσεις να σβήνουν και να εξαφανίζονται .

Ερώτημα 5:

Χρησιμοποιώντας **τις κεντρικές διαφορές δεύτερης τάξης (και πρώτης τάξης για τη συνοριακή συνθήκη του δεξιού άκρου του διαστήματος)** να γράψετε ένα πρόγραμμα Octave-Matlab το οποίο να κατασκευάζει προσεγγίσεις για προβλήματα της μορφής

$$\begin{gathered} -u(x) + r(x)u(x) = f(x) \\ u(a) = A, \qquad u'(b) = B \end{gathered}$$

Υποθέτουμε ότι οι συναρτήσεις **r(x)>0**, και **f(x)** είναι **ομαλές** και ότι χρησιμοποιούμε ισαπέχοντα σημεία διαμέρισης.

Να χρησιμοποιήσετε το πρόγραμμα σας για να λύσετε το πρόβλημα συνοριακών τιμών της μορφής:

$$\begin{Bmatrix} -\mathbf{10^{-k}} \boldsymbol{u''(x)} + \boldsymbol{u(x)} = \; -\mathbf{10^{-k}} \times \mathbf{12}\boldsymbol{x^2} + \boldsymbol{x(x^3-2)}, \qquad \mathbf{0} < \boldsymbol{x} < \mathbf{1} \\ \boldsymbol{u(0)} = \mathbf{0}, \qquad \boldsymbol{u'(1)} = \mathbf{2} \end{Bmatrix}$$

Συγκεκριμένα, να τρέξετε το πρόγραμμα σας χρησιμοποιώντας **διαμερίσεις μήκους h=0.1, h=0.001** για κ= -2,-1,0,1,2 αντιστοίχως και να τυπώσετε τα γραφήματα των προσεγγιστικών λύσεων καθώς και της πραγματικής λύσης $\boldsymbol{u(x) = x(x^3-2)}$. Να εκτιμήσετε πειραματικά την τάξη ακρίβειας της μεθόδου και να υπολογίσετε το δείκτη κατάστασης του πίνακα σε όλες τις περιπτώσεις. Τι παρατηρείτε;

ΛΥΣΗ

$$\begin{Bmatrix} -u(x) + r(x)u(x) = f(x) \\ u(a) = A, \qquad u'(b) = B \end{Bmatrix}$$

Όπου : $u(a) = A$ συνοριακή συνθήκη Dirichlet

και $\quad u'(b) = B$ συνοριακή συνθήκη Neumann

- Για την σ.σ. Neumann εισάγουμε βοηθητικό κόμβο: $\boldsymbol{x_{N+1} = x_N + h = b + h}$
- Παρατηρούμε ότι :

$$\begin{aligned} B = \; & u'(b) = u'(x_N) \\ & = \lim_{\varepsilon \to 0} \frac{u(x_N+\varepsilon) - u(x_N-\varepsilon)}{2\varepsilon} \approx \frac{u(x_N+h) - u(x_N-h)}{2h} = \frac{u(x_{N+1}) - u(x_{N-1})}{2h} \\ & \Leftrightarrow \boldsymbol{u(x_{N+1}) = u(x_{N-1}) + 2Bh} \end{aligned}$$

~ Δείκτης Κατάστασης ~

1. **Μικρός Δείκτης Κατάστασης (Κοντινός στο 1):** Αυτό υποδηλώνει ότι ο πίνακας είναι λίγο ευαίσθητος σε μικρές αλλαγές στα δεδομένα εισόδου. Σε γενικές γραμμές, όσο πιο κοντά στο 1 είναι ο δείκτης, τόσο καλύτερη η συμπεριφορά του πίνακα.
2. **Μεγάλος Δείκτης Κατάστασης:** Ένας μεγάλος δείκτης υποδηλώνει ότι ο πίνακας είναι ευαίσθητος σε μικρές αλλαγές. Αυτό μπορεί να οδηγήσει σε αστάθεια κατά την επίλυση συστημάτων γραμμικών εξισώσεων.
3. **Υπερβολικά Μεγάλος Δείκτης Κατάστασης:** Εάν ο δείκτης είναι πολύ μεγάλος, μπορεί να υποδηλώνει ότι ο πίνακας είναι πολύ κακά συνθηκευμένος (ill-conditioned), και η ακρίβεια των αριθμητικών υπολογισμών μπορεί να είναι περιορισμένη.

Ορίζω συναρτηση :

```
function U = FDM_Dirichlet_Neumann(a,b,A,B,r,f,N)
h=(b-a)/(N+1);
x=linspace(a,b,N+2); %x=linspace(a,b,N+2);
U=zeros(1,N+2); %<------ oxi U=zeros(1,N+2);
U(1)=A;
%U(N+1)= B;%<---------- mipws U(N+2)= U(N+1)+2*B*h;


a1     =(-1/h^2)*ones(N,1); %katw diag
a1(N-1)= -2/(h^2);

a2=(2/h^2)*ones(N,1)+r(x(2:N))' ; %diag

a3= (-1/h^2)*ones(N,1); %anw diag


M = spdiags([a1,a2,a3],[-1,0,1],N,N);

%Ta afksisa ola kata mia (+1) diastasi
F = f(x(2:N+1))';
F(1)  = f(x(2)) + A/(h^2);
F(N)  = f(x(N+1)) + 2*B/h;

Uint = M\F; %lisi systimatos

U(2:N+1)=Uint';
U(N+2)= U(N)+2*B*h;
end%Uint dianysma stili enw to U einai dianysma rammi
```

Περίπτωση 1 : κ= -2

$$\begin{cases} -10^{+2}u''(x) + u(x) = -10^{+2} \times 12x^2 + x(x^3-2), & 0 < x < 1 \\ u(0)=0, \quad u'(1) = 2 \end{cases}$$

```
%$$$$$$ ORISMOS PAT $$$$$$$$$$$
a=0.0;
b=1.0;
A=0.0;
B=2.0;

k = -2.0;
r =@(x) 10^k;
f =@(x) -12*x.^2 + 10^(k)*x.*(x.^(3) -2);

uexact=@(x) x.*(x.^(3) -2);
%$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

%Ns=[10,1000,2000,3000];
Ns=[5,9,999];
errs =zeros(length(Ns),1);
rates=zeros(length(Ns)-1,1);

for i=1:length(Ns)
    U = FDM_Dirichlet_Neumann(a,b,A,B,r,f,Ns(i));
    x = linspace(a,b,Ns(i)+2);
    errs(i)  = max(abs(uexact(x)-U));
end

for i=1:length(Ns)-1

    den=log(Ns(i+1)/Ns(i));
    rates(i)=log(errs(i)/errs(i+1))/den;

end

errs
rates
cond_U = cond(U);
cond_U
```

```
>> ergastiriaki_ergasia_Art_2023_2024_erwt5

errs =

    1.6081
    1.0535
    0.0119


rates =

    0.7196
    0.9512


cond_U =

     1
```

<u>Περίπτωση 2</u>: κ= -1

$$\begin{Bmatrix} -10^{+1}u''(x) + u(x) = -10^{+1} \times 12x^2 + x(x^3 - 2), \qquad 0 < x < 1 \\ u(0) = 0, \qquad u'(1) = 2 \end{Bmatrix}$$

```
%$$$$$$ ORISMOS PAT $$$$$$$$$$$
a=0.0;
b=1.0;
A=0.0;
B=2.0;

k = -1.0;
r =@(x) 10^k;
f =@(x) -12*x.^2 + 10^(k)*x.*(x.^(3) -2);

uexact=@(x) x.*(x.^(3) -2);
%$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

%Ns=[10,1000,2000,3000];
Ns=[5,9,999];
errs =zeros(length(Ns),1);
rates=zeros(length(Ns)-1,1);

for i=1:length(Ns)
    U = FDM_Dirichlet_Neumann(a,b,A,B,r,f,Ns(i));
    x = linspace(a,b,Ns(i)+2);
    errs(i) = max(abs(uexact(x)-U));
end

for i=1:length(Ns)-1

    den=log(Ns(i+1)/Ns(i));
    rates(i)=log(errs(i)/errs(i+1))/den;

end

errs
rates
cond_U = cond(U);
cond_U
```

```
>> ergastiriaki_ergasia_Art_2023_2024_erwt5

errs =

    1.5819
    1.0314
    0.0116


rates =

    0.7277
    0.9529


cond_U =

     1
```

Περίπτωση 3 : κ= 0

$$\begin{cases} -10^{0}u''(x) + u(x) = \; -10^{0} \times 12x^2 + x(x^3 - 2), \qquad 0 < x < 1 \\ \qquad\qquad u(0) = 0, \qquad u'(1) = 2 \end{cases}$$

```
%$$$$$$ ORISMOS PAT $$$$$$$$$$$
a=0.0;
b=1.0;
A=0.0;
B=2.0;

k = 0.0;
r =@(x) 10^k;
f =@(x) -12*x.^2 + 10^(k)*x.*(x.^(3) -2);

uexact=@(x) x.*(x.^(3) -2);
%$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

%Ns=[10,1000,2000,3000];
Ns=[5,9,999];
errs =zeros(length(Ns),1);
rates=zeros(length(Ns)-1,1);

for i=1:length(Ns)
    U = FDM_Dirichlet_Neumann(a,b,A,B,r,f,Ns(i));
    x = linspace(a,b,Ns(i)+2);
    errs(i) = max(abs(uexact(x)-U));
end

for i=1:length(Ns)-1

    den=log(Ns(i+1)/Ns(i));
    rates(i)=log(errs(i)/errs(i+1))/den;

end

errs
rates
cond_U = cond(U);
cond_U
```

```
>> ergastiriaki_ergasia_Art_2023_2024_erwt5

errs =

    1.3768
    0.8641
    0.0091


rates =

    0.7926
    0.9660


cond_U =

     1
```

Περίπτωση 4: κ= 1

$$\begin{cases} -10^{-1}u''(x) + u(x) = -10^{-1} \times 12x^2 + x(x^3 - 2), & 0 < x < 1 \\ u(0) = 0, \quad u'(1) = 2 \end{cases}$$

```
%$$$$$$ ORISMOS PAT $$$$$$$$$$$
a=0.0;
b=1.0;
A=0.0;
B=2.0;

k = 1.0;
r =@(x) 10^k;
f =@(x) -12*x.^2 + 10^(k)*x.*(x.^(3) -2);

uexact=@(x) x.*(x.^(3) -2);
%$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

%Ns=[10,1000,2000,3000];
Ns=[5,9,999];
errs =zeros(length(Ns),1);
rates=zeros(length(Ns)-1,1);

for i=1:length(Ns)
    U = FDM_Dirichlet_Neumann(a,b,A,B,r,f,Ns(i));
    x = linspace(a,b,Ns(i)+2);
    errs(i) = max(abs(uexact(x)-U));
end

for i=1:length(Ns)-1

    den=log(Ns(i+1)/Ns(i));
    rates(i)=log(errs(i)/errs(i+1))/den;

end

errs
rates
cond_U = cond(U);
cond_U
```

```
>> ergastiriaki_ergasia_Art_2023_2024_erwt5

errs =

    0.8187
    0.4479
    0.0038


rates =

    1.0261
    1.0134


cond_U =

     1
```

Περίπτωση 5 : κ= 2

$$\begin{cases} -10^{-2}u''(x) + u(x) = -10^{-2} \times 12x^2 + x(x^3 - 2), & 0 < x < 1 \\ u(0) = 0, \quad u'(1) = 2 \end{cases}$$

```
%$$$$$$ ORISMOS PAT $$$$$$$$$$$
a=0.0;
b=1.0;
A=0.0;
B=2.0;

k = 2.0;
r =@(x) 10^k;
f =@(x) -12*x.^2 + 10^(k)*x.*(x.^(3) -2);

uexact=@(x) x.*(x.^(3) -2);
%$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

%Ns=[10,1000,2000,3000];
Ns=[5,9,999];
errs =zeros(length(Ns),1);
rates=zeros(length(Ns)-1,1);

for i=1:length(Ns)
    U = FDM_Dirichlet_Neumann(a,b,A,B,r,f,Ns(i));
    x = linspace(a,b,Ns(i)+2);
    errs(i)  = max(abs(uexact(x)-U));
end

for i=1:length(Ns)-1

    den=log(Ns(i+1)/Ns(i));
    rates(i)=log(errs(i)/errs(i+1))/den;

end

errs
rates
cond_U = cond(U);
cond_U
```

```
>> ergastiriaki_ergasia_Art_2023_2024_erwt5

errs =

    0.5583
    0.2456
    0.0012


rates =

    1.3969
    1.1281


cond_U =

     1
```

ΕΡΜΗΝΕΙΑ ΑΠΟΤΕΛΕΣΜΆΤΩΝ

- Παρατηρούμε ότι όσο οι διαμερίσεις αυξάνονται , δηλαδή η τιμή του h μικραίνει, τόσο το σφάλμα (errs) που έχει η αριθμητική προσέγγιση από την πραγματική λύση συνεχώς μειώνεται και άρα τόσο πιο κοντά έρχεται η προσεγγιστική λύση στην πραγματική.
- Η τάξη ακρίβειας της παραπάνω μεθόδου φαίνεται να είναι 1 καθώς το διάνυσμα των τάξεων δείχνει να τείνει στο 1. Συγκεκριμένα , όσο το $k \leq 0$, η τάξη ακρίβειας 1 της μεθόδου φαίνεται να προσεγγίζεται από κάτω (δηλαδή όσο αυξάνεται η διαμέριση όλο ένα και αυξάνεται η τιμή της προσεγγιστικής τάξης ακρίβειας), ενώ για $k>0$ η τάξη ακρίβειας 1 της μεθόδου φαίνεται να προσεγγίζεται από πάνω (δηλαδή όσο αυξάνεται η διαμέριση όλο ένα και μειώνεται η τιμή της προσεγγιστικής τάξης ακρίβειας).
- Ο δείκτης κατάστασης του πίνακα U φαίνεται να είναι ίσος με 1 , που σημαίνει ότι ο πίνακας είναι λίγο ευαίσθητος σε μικρές αλλαγές στα δεδομένα εισόδου αλλά πολύ καλό εργαλείο προσέγγισής. Σε γενικές γραμμές, όσο πιο κοντά στο 1 είναι ο δείκτης, τόσο καλύτερη η συμπεριφορά του πίνακα.