Επιστημονικός Υπολογισμός
1η Εργαστηριακή Άσκηση

Κανελλόπουλος Χαράλαμπος, Α.Μ:4994

15 Οκτωβρίου, 2014

1 Εισαγωγικά

1.1

Σύστημα : Windows 7 x64
Επεξεργαστής : Intel Core i7 950 @ 3.07 GHz
Physical Memory: 6GBytes DDR3-SDRAM 3x 2048Mb
Virtual Memory:
Cache L1: 4x 32KBytes
Cache L2: 4x 256KBytes
Cache L3: 4x 8MBytes
Write Mode: Write back

1.2

Έκδοση:Matlab R2014a

1.3

Η εντολή tic toc καλέστηκε 31 φορές χωρίς να μετρηθεί η 1η και το αποτέλεσμα ήταν 2.4195e-07

1.4

Τρέξαμε την συνάρτηση bench 6 φορές.Ο μικρότερος χρόνος φένεται από κάτω:

This machine (run 1)	0.0858

2 Χρονομέτρηση Συναρτήσεων

2.1

- **Παραγοντοποίηση LU**:Όπου LU προέρχεται από το "Lower,Upper".Η εντολή[L,U]=lu(A) εκφράζει το μητρώο A ως γινόμενο 2 τριγωνικών μητρείων, το ένα κάτω τριγωνικό και το άλλο άνω τριγωνικό.Με την παραγωντοποίηση αυτή, ένα γενικό σύστημα εξισώσεων Ax=b μετατρέπεται σε ζεύγος τριγωνικών συστημάτων Ly=Pb, Ux=y.
  Τέλος η παραγοντοποίηση LU είναι η απαλοιφή του Gauss εκφρασμένη σε μητρώα.
- **Παραγοντοποίηση QR**:Αναφέρεται και ως διαχωρισμός QR.Για κάθε πείνακα mxn η εντολή qr(A) εκφράζει τον διαχωρισμό ενώς μητρώου A σε ένα ορθογώνιο μητρώο Q,τύπου mxn και σε ένα άνω τριγωνικό R τύπου mxn (A=QR).
- **Εντολή svd**:Η εντολή s=svd(n) μας επιστρέφει ένα διάνυσμα από ιδιόμορφες τιμές.
- **Εντολή det**:Η εντολή d=det(n) μας επιστρεφει την ορίζουσα του τετραγωνικού πίναχα n.Αν το n περιέχει μόνο ακέραιες καταχωρήσεις, τότε το αποτέλεσμα d θα είναι επίσης ένας ακέραιος.

- **Εντολή rank**:Η εντολή k=rank(A) μας επιστρέφει τον αριθμό των ιδιόμορφων τιμών του A που ειναι μεγαλύτερες απο το αρχικό.Η εντολή rank(A,tol) μας επιστρέφει τον αριθμό των ιδιόμορφων τιμών του A που είναι μεγαλύτερες απο το tol

2.2

α)

```
n=2.^[7:10];  %diastaseis n
time = zeros(2,4);       %preallocation twn dianusmatwn gia pio grhgorh ektelesh.
for i = [1:4]            %apo 1:4 giati to n pernei 4 times(128,256,512,1024).
    A= randn(n(i),n(i)) %Paragwgh tyxaiou pinaka A nxn.
    b= randn(n(i),1);   %Paragwgh tyxaiou pinaka b nx1.

    [L,U] = lu(A);         %prwth ektelesh ths prakshs lu.
    mtimes(A,b);           %prwth ektelesh tis prakshs A*b mesw ths mtimes.


    tic;                   %Xrhsh ths tic/toc gia thn xronometrhsh ths lu(A).
    [L,U] = lu(A);         %Deuterh eketelesh afou exoun metafer8ei ta dedomena stin mnhmh.
    time(1,i) = toc;

    tic;                   %Xrhsh ths tic/toc gia thn xronometrhsh ths mtimes(A,b).
    mtimes(A,b);           %Deuterh eketelesh afou exoun metafer8ei ta dedomena stin mnhmh.
    time(2,i) = toc;
end

figure
plot(n, time(1,:), '-*r', n, time(2,:),'-b*')
legend('LU', 'Mtimes','Location', 'Northwest')
title('Xronometrhsh a')
xlabel('n = 2. ^ [7:10]')
ylabel('time')
```

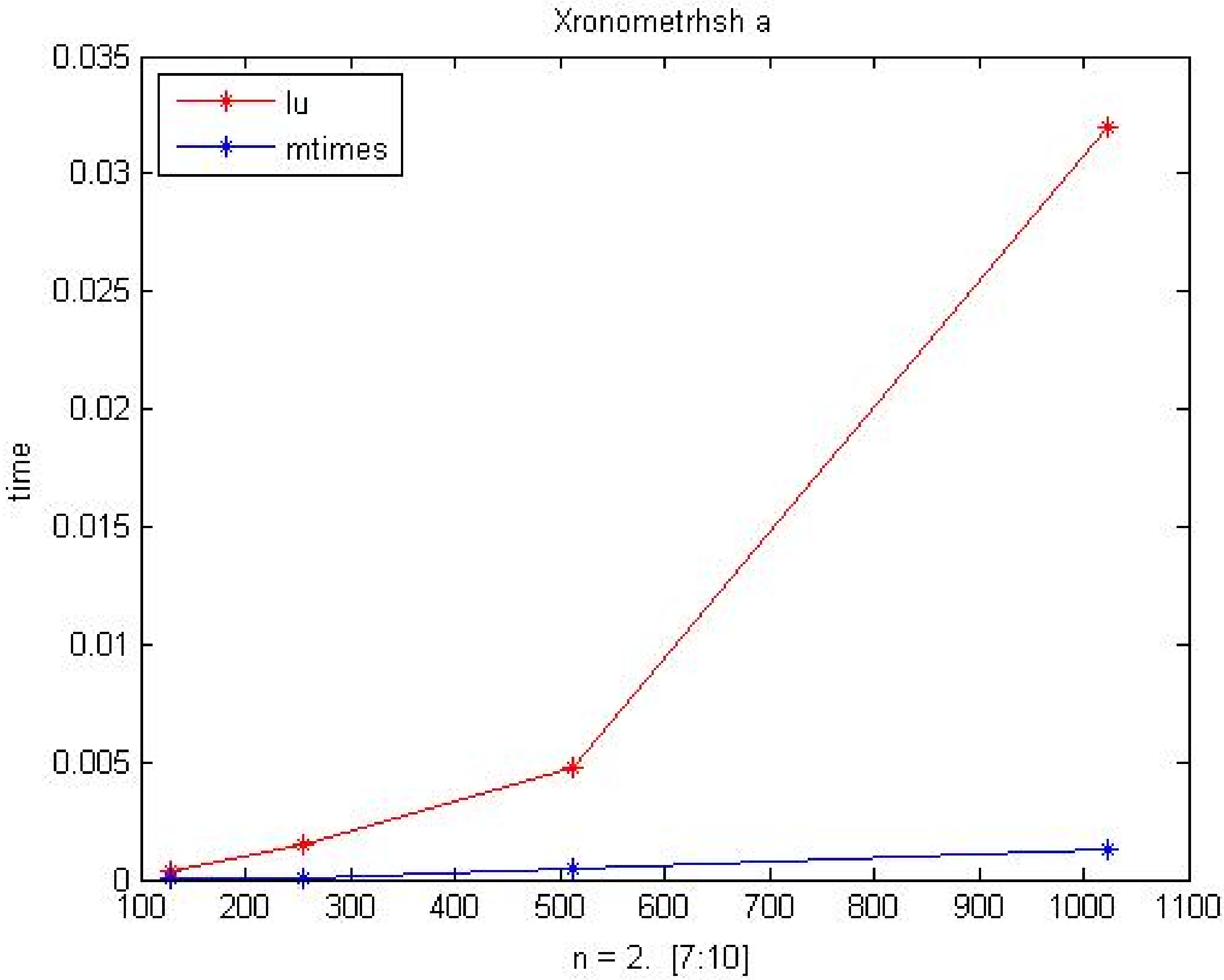
β)

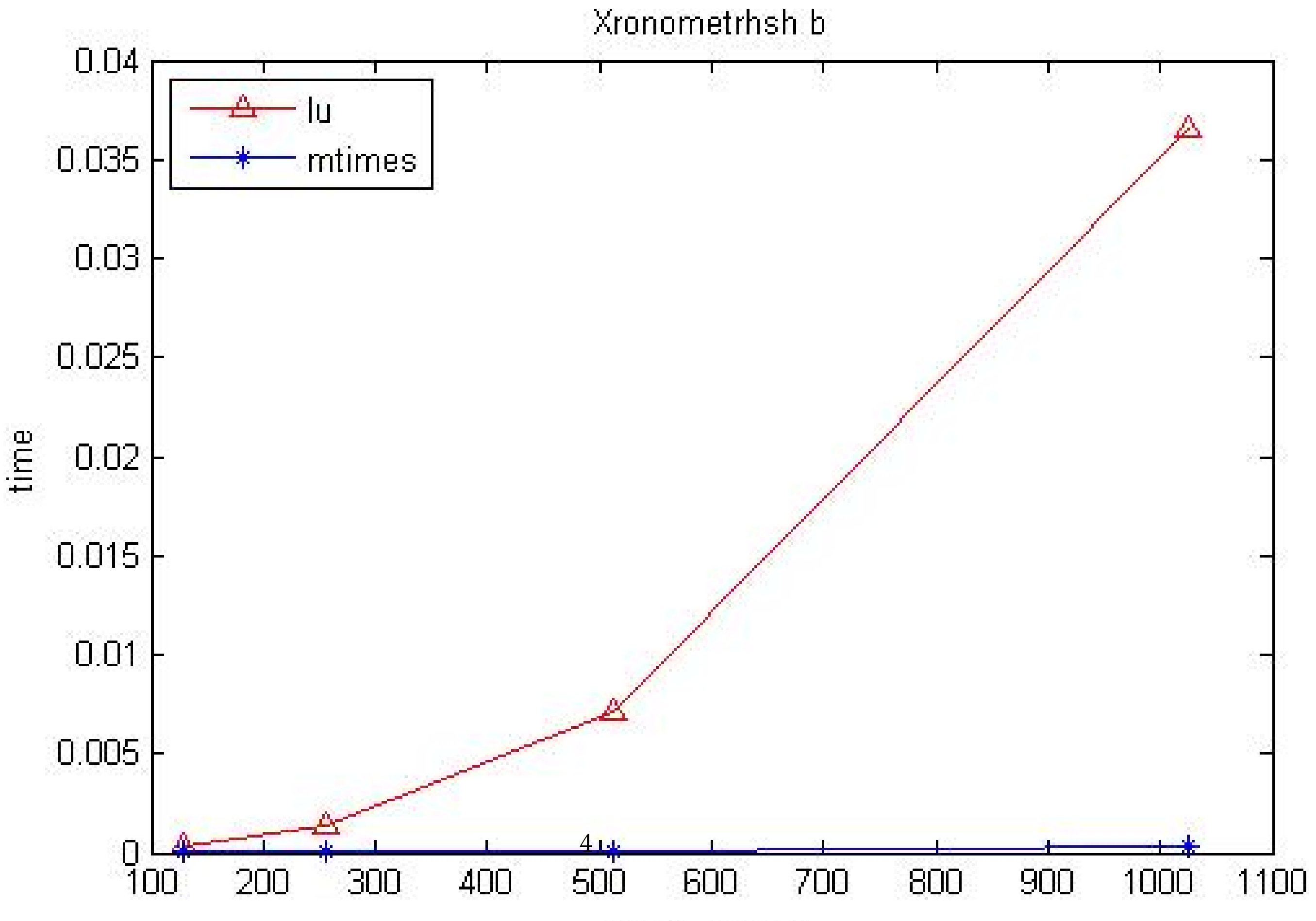
```
n=2.^[7:10];  %diastaseis n
accurate_time = zeros(2,4) %preallocation twn dianusmatwn gia pio grhgorh ektelesh
for i = [1:4]              %apo 1:4 giati to n pernei 4 times(128,256,512,1024).
    A= randn(n(i),n(i))    %Paragwgh tyxaiou pinaka A nxn.
    b= randn(n(i),1);      %Paragwgh tyxaiou pinaka b nx1.

    [L,U] = lu(A);         %prwth ektelesh ths prakshs lu.
    time = zeros(10,1);    %preallocation twn dianusmatwn gia pio grhgorh ektelesh.
    for j=[1:10]
        tic;
        [L,U] = lu(A);     %Deuterh eketelesh afou exoun metafer8ei ta dedomena stin mnhmh.
        time(j)=toc;
    end
    accurate_time(1,i) = sum(time)./10; %Mesos oros 10 xronwn kai eksagwgh sthn o8onh.

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    mtimes(A,b);           %prwth ektelesh ths prakshs A*b mesw ths mtimes.
    for j=[1:10]
        tic;
        mtimes(A,b);       %Deuterh eketelesh afou exoun metafer8ei ta dedomena stin mnhmh.
        time(j)=toc;
    end
    accurate_time(2,i) = sum(time)./10; %Mesos oros 10 xronwn kai eksagwgh sthn o8onh.

end

figure
plot(n, accurate_time(1,:), '-r^', n, accurate_time(2,:),'-b*')
legend('lu', 'mtimes','Location', 'Northwest')
title('Xronometrhsh b')
xlabel('n = 2. ^ [7:10]')
ylabel('time')
```

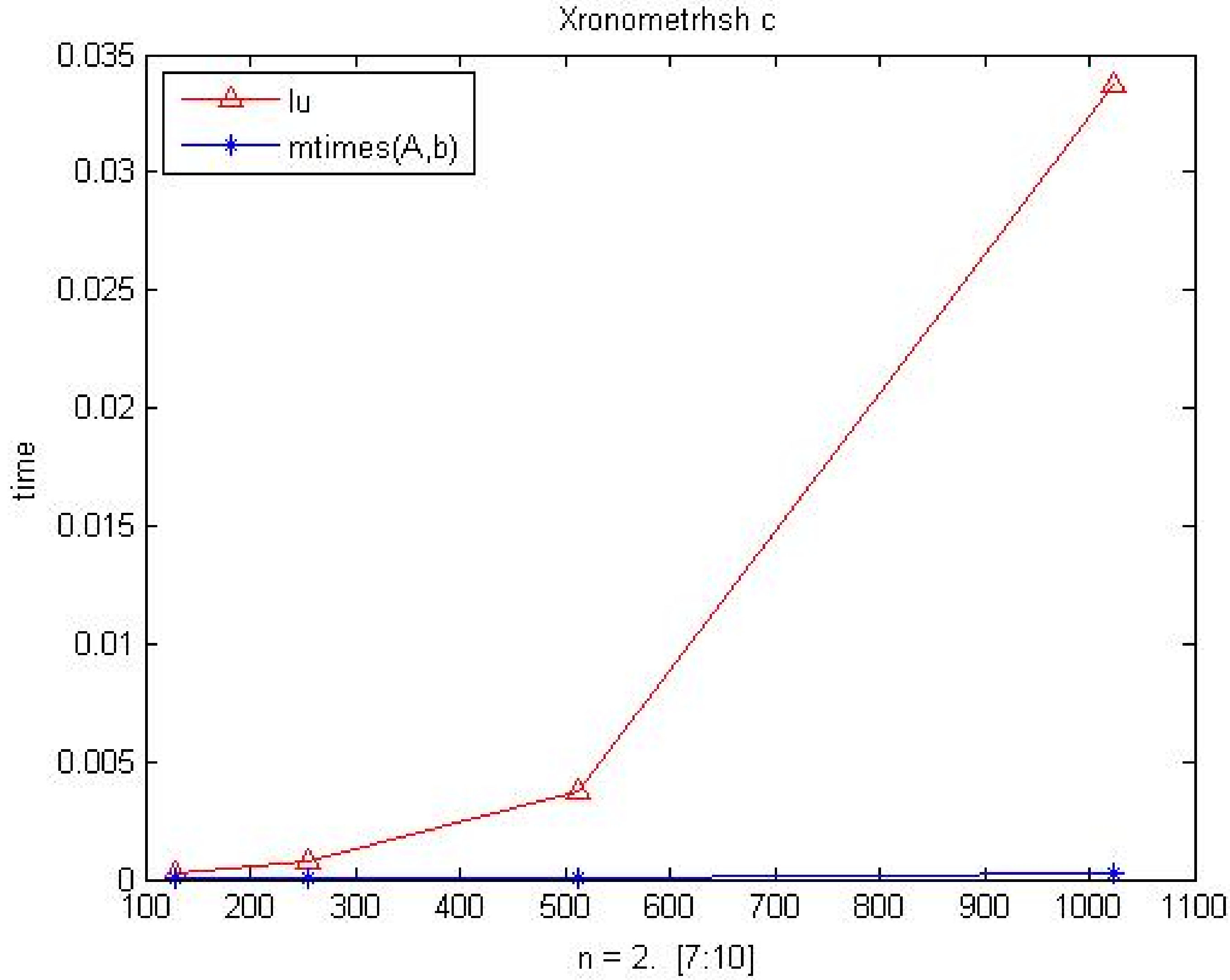
γ)

```
n=2.^[7:10];  %diastaseis n.
mat_time_timeit = zeros(2,4);%preallocation twn dianusmatwn gia pio grhgorh ektelesh.
for i = [1:4]               %apo 1:4 giati to n pernei 4 times(128,256,512,1024).
    A= randn(n(i),n(i))     %Paragwgh tyxaiou pinaka A nxn.
    b= randn(n(i),1);       %Paragwgh tyxaiou pinaka b nx1.

    f = @ () lu(A);                     %Klhsh ths timit gia thn xronometrhsh ths entolhs lu.
    mat_time_timeit(1,i) = timeit(f)    %Apothikeush ston pinaka mat_time_timeit ta ...
        apotelesmata ths timeit.


    f = @ () mtimes(A,b);               %Klhsh ths timit gia thn xronometrhsh ths entolhs ...
        mtimes(A,b).
    mat_time_timeit(2,i) = timeit(f)    %Apothikeush ston pinaka mat_time_timeit ta ...
        apotelesmata ths timeit.
end

figure
plot(n, mat_time_timeit(1,:), '-^r', n, mat_time_timeit(2,:),'-b*')
legend('LU', 'mtimes(A,b)','Location', 'Northwest')
title('Xronometrhsh c')
xlabel('n = 2. ^ [7:10]')
ylabel('time')
```

2.3


Xronometrhsh a
lu
mtimes
time
n = 2. [7:10]


Xronometrhsh b
lu
mtimes
time



Στο πρώτο διάγραμμα (xronometrhsh a) έχουμε 1 γραφικη παράσταση για την lua και την mtimes.Αρχικά εκτελούμε την συνάρτηση μία φορα για να μεταφερθούν τα δεδομένα και έπειτα πέρνουμε μία μέτρηση.Έπειτα όμως επειδή θέλουμε οι μετρήσεις μας να έχουν μεγαλύτερη ακρίβεια χρησιμοποιούμε επαναλήψεις και βγάζουμε τον μέσο όρο από τα αποτελέσματα των χρονομετρήσεων.Στο δεύτερο διάγραμμα (xronometrhsh b) εκτελούμε τις δύο συναρτήσεις πρώτα μια φορά έξω από τις επαναλήψεις και έπειτα πέρνουμε τον μέσο όρο των αποτελεσμάτων και τα αποθηκεύουμε σε έναν πίνακα.Μεταξύ των δύο γραφιμάτων παρατηρούνται διαφορές στις lu,mtimes με πιο αισθητή την διαφορα της απεικόνισης στην mtimes.Αυτό συμβαίνει διότι στο 2ο διάγραμμα έχουμε πάρει τον μέσο όρο και έτσι επιτυγχάνεται μεγαλύτερη αξιοπιστία στις μετρήσεις. Τέλος στο τρίτο διάγραμμα (xronometrhsh c) δημιουργούμε την γραφική απεικόνηση για τις 2 συναρτήσεις, χρησιμοποιώντας για την χρονομέτρηση την timeit.Με την χρήση της timeit υπολογίζουμε την μέση τιμή πολλών επαναλαμβανόμενων μετρήσεων.Η διαφορά με την 2η χρονομέτρηση είναι ότι η timeit γνωρίζει πόσες επαναλήψεις χρειαζόμαστε για την κάθε συνάρτηση.Παρατηρείται ότι και στα 3 διαγράματα μεγαλύτερο χρόνο εκτέλεσης κάνει η lu και ότι αυξάνεται ο χρόνος υπολογισμού όσο αυξάνεται και το μέγεθος του μητρώου.

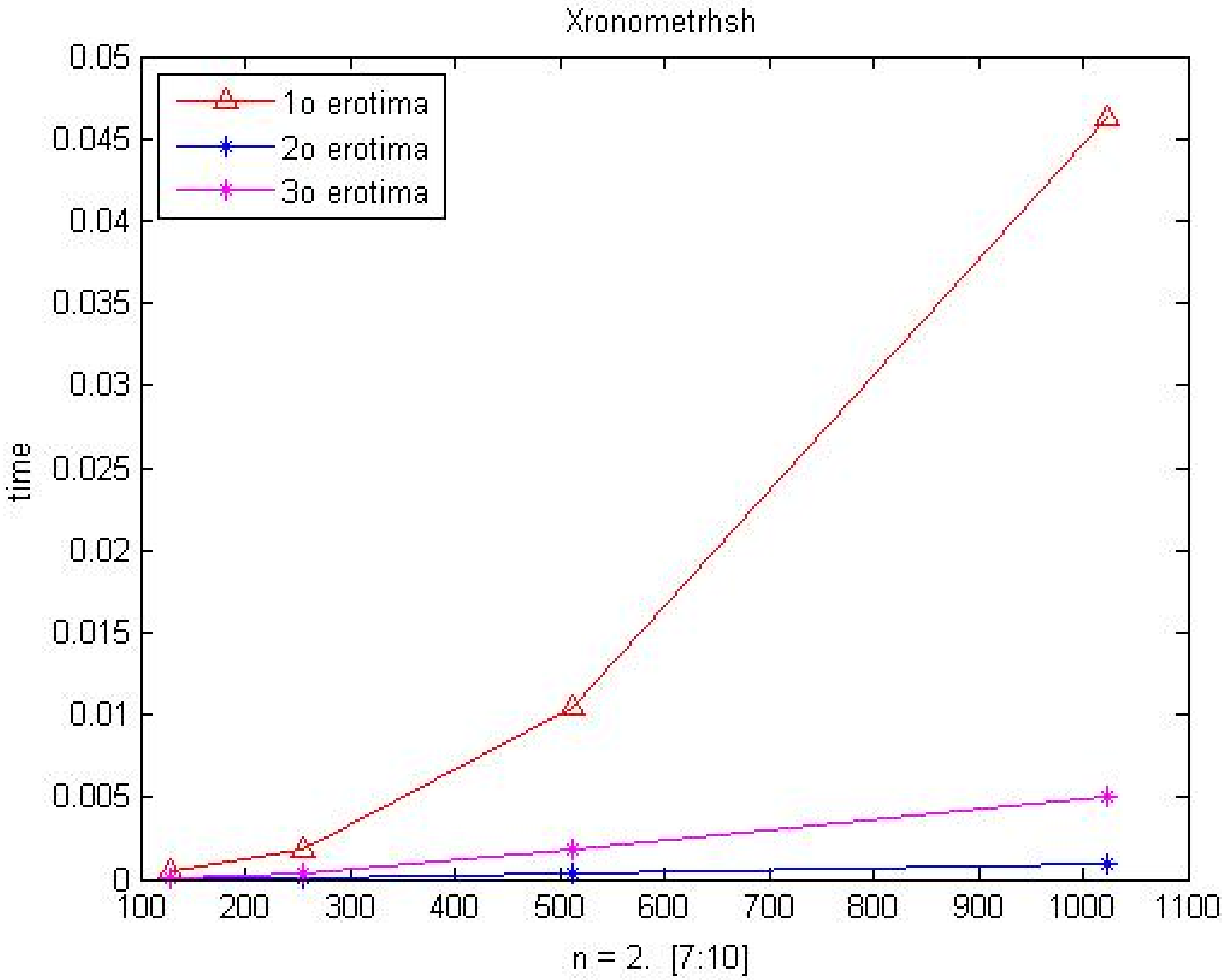
3 Αξιολόγηση Ενδογενών Συναρτήσεων

3.1

i,ii,iii)

```matlab
n=2.^[7:10];  %diastaseis n
mat_time_timeit = zeros(3,4); %preallocation twn dianusmatwn gia pio grhgorh ektelesh.
for i=[1:4]                    %apo 1:4 giati to n pernei 4 times(128,256,512,1024).
    A=randn(n(i),n(i));        %Paragwgh tyxaiou pinaka A nxn.
    b=randn(n(i),1);           %Paragwgh tyxaiou pinaka b nx1.

    c = zeros(n(i));          %preallocation twn dianusmatwn gia pio grhgorh ektelesh.
    s = zeros(n(i));          %preallocation twn dianusmatwn gia pio grhgorh ektelesh.

    c=tril(A);                %Dimiourgia katw trigonikou mitroou mesw tis tril.

    s = c(:,randperm(size(c,2))); %Dhmiourgia psixologikou katw trigonikou mitroou mesw ths ...
        randperm kai ths size gia thn antimetathesh
                                   %twn stulwn.


    f = @ () mldivide(A,b);
    mat_time_timeit(1,i) = timeit(f)  %Klhsh ths timeit gia xronometrhsh ths mldivide(A,b).


    f = @ () mldivide(c,b);
    mat_time_timeit(2,i) = timeit(f)  %Klhsh ths timeit gia xronometrhsh ths mldivide(c,d).


    f = @ () mldivide(s,b);
    mat_time_timeit(3,i) = timeit(f)  %Klhsh ths timeit gia xronometrhsh ths mldivide(s,d).


end




figure
plot(n, mat_time_timeit(1,:), '-^r', n, mat_time_timeit(2,:),'-b*', n, ...
    mat_time_timeit(3,:),'-m*')
legend('1o erotima', '2o erotima','3o erotima','Location', 'Northwest')
title('Xronometrhsh')
xlabel('n = 2. ^ [7:10]')
ylabel('time')
```

3.2

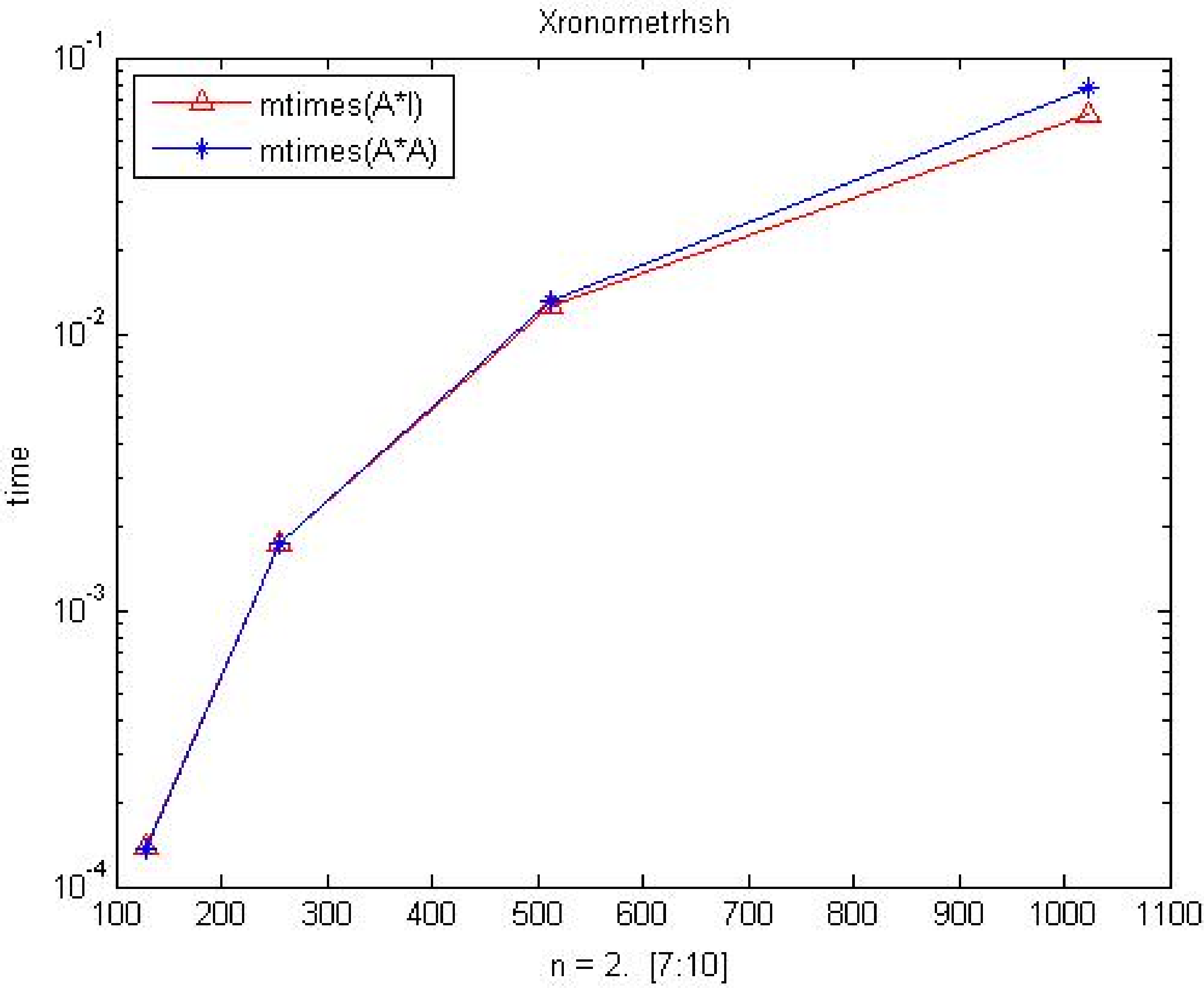


3.3

Αρχικά παρατηρούμε στο διάγραμμα πως τον μεγαλύτερο χρόνο τον κάνει η διαίρεση όταν έχουμε τυχαίο μητρώο Α.Αυτό συμβαίνει διότι το μητρώο Α είναι γεμάτο και η matlab εκτελεί όλες τις πράξεις.
Έπειτα παρατηρούμε πως τον μικρότερο χρόνο τον κάνει η διαίρεση όταν έχουμε κάτω τριγωνικά μητρώα.Η matlab όταν χρησιμοποιούμε την tril αυτόματα διχοτομεί το μητρώο σε 2 τριγωνικά μέρη ,γεμίζοντας το πάνω μέρος με μηδενικά και το κάτω με στοιχεία.Έπειτα όταν προχωράμε στην κλήση της mldivide για την διαίρεση του κάτω τριγονικού με ένα διάνυσμα διαστάσεων nx1 η matlab γνωρίζει ότι το πάνω τριγωνικό μέρος του μητρώου μας έχει παντού μηδενικά οπότε δεν εκτέλει καθόλου την διαίρεση για αυτό το μέρος και εκτελεί τις πράξεις μόνο για το κάτω τριγονικό ,όπου έχουμε και τα στοιχεία μας.
Τέλος παρατηρούμε ότι η διαίρεση για τα ψυχολογικά κάτω τριγονικά μητρώα κάνει λίγο περισσότερο χρόνο από την διαίρεση για τα κάτω τριγονικά και αυτό συμβαίνει γιατί έχοντας κάνει αντιμετάθεση στις στήλες του πίνακα το matlab δεν γνωρίζει όπως πριν ότι ένα μέρος του μητρώου είναι γεμάτο με μηδενικά.Παρόλαυτα είναι πολύ πιο γρήγορο από την περίπτωση με τα τυχαία μητρώα Α κι αυτό γιατί το matlab γλιτώνει πράξεις από τα κελία του μητρώου που περιέχουν μηδενικα.

3.4

```
n=2.^[7:10];  %diastaseis n
mat_time_timeit = zeros(2,4);%preallocation twn dianusmatwn gia pio grhgorh ektelesh
for  i=[1:4]                 %po 1:4 giati to n pernei 4 times(128,256,512,1024).
   A=randn(n(i),n(i));       %Paragwgh tyxaiou pinaka A nxn mesw ths randn.
   I= eye(n(i));             %Dimiourgiou tou tautotikou pinaka nxn mesw ths eye.

   f = @ () mtimes(A,I);
   mat_time_timeit(1,i) = timeit(f);%Klhsh ths timeit gia xronometrhsh ths mtimes(A,I).


   f = @ () mtimes(A,A);
   mat_time_timeit(2,i) = timeit(f);%Klhsh ths timeit gia xronometrhsh ths mtimes(A,A).


end

figure
semilogy(n, mat_time_timeit(1,:), '-^r', n, mat_time_timeit(2,:),'-b*' )
legend('mtimes(A*I)', 'mtimes(A*A)', 'Location', 'Northwest')
title('Xronometrhsh ')
xlabel('n = 2. ^ [7:10]')
ylabel('time')
```



Από το παραπάνω διάγραμμα παρατηρούμε ότι οι δύο χρόνοι για την εκτέλεση των mtimes(A*I) kai mtimes(A*A), όπου A τυχαίο μυτρώο nxn και I ταυτοτικό μυτρώο nxn , είναι ίδιοι με μια πολύ μικρή απόκλιση στο τέλος.

4 Σύγκριση Υλοποιήσεων

4.1

Από την εκφώνηση έχουμε το γινόμενο:

$$\prod_{1}^{p}(I - u * v^T)$$

Όπου I έχουμε το ταυτοτικό μητρώο nxn και όπου u,v τυχαία διανύσματα nx1.
Αρχικά η πρώτη πράξη που γίνεται μέσα στην παρένθεση είναι ο πολ/σμος του διανύσματος u με το transposed διάνυσμα v. Η ύπαρξη του εκθέτη T στο διάνυσμα v μας δείχνει ότι ο πολ/σμος του διανύσματος u γίνεται με το transposed του v,που είναι διάνυσμα 1xn.
Το πλήθος των πράξεων που χρησιμοποιούμε για τον πολ/σμο 2 διανυσμάτων είναι n^2.(1)
Από τον πολ/σμό $u*v^T$ έχουμε δημιουργήσει έναν νέο πίνακα διαστάσεων nxn που έπειτα θα αφαιρέσουμε από τον πίνακα I.
Το πλήθος των πράξεων που χρησιμοποιούμε για την αφαίρεση 2 πινάκων είναι n^2.(2)
Άρα από (1),(2) έχουμε ότι για την πράξη $(I - u * v^T)$ το απαιτούμενο πλήθος πράξεων είναι $2n^2$.(3)
Από την εκφώνηση έχουμε πως το όρισμα της παρένθεσης θα το πολ/σουμε με τον εαυτό του p φορές.Δηλαδή θα γίνουν $p - 1$ πολ/σμοι.
Έστω A ο πίνακας που δημιουργήσαμε στο (3).Το πλήθος των πράξεων που χρησιμοποιούμε για να πολ/σουμε 2 πίνακες nxn είναι $n^2 * (2n - 1)$.
Άρα το απαιτούμενο πλήθος πράξεων κινητής υποδιαστολής συναρτήσει των n και p είναι :
$2n^2 + (p - 1) * n^2 * (2n - 1)$.
όπου $2n^2$ είναι οι πράξεις για την δημιουργία του A από (3).

4.2

Από την εκφώνηση της 1) έχουμε το γινόμενο:

$$\prod_{1}^{p}(I - u * v^T)$$

Αρχικά πολ/ζουμε με e από δεξιά όπου e από την εκφώνηση θεωρούμε μία στήλη από το ταυτοτικό μητρώο I.
Οι πράξεις αναφέρεται ότι γίνονται από τα δεξιά προς τα αριστερά οπότε πολ/ζουμε επιμεριστικά το e με το όρισμα της παρένθεσης.
Από τον πολ/σμο $u * v^T * e$ παρατηρούμε πως ο πολ/σμος $v^T * e$ μας δείνει ένα μητρώο nxn .Το πλήθος των πράξεων για αυτόν τον πολ/σμο είναι n^2.(1)
Έπειτα πολ/ζουμε το διάνυσμα u με το μητρώο που έχουμε παραπάνω το οποίο κοστίζει $2n^2 - n$ πράξεις.(2)
Από την επιμεριστική που εκτελέσαμε παραπάνω έχουμε τον πολ/σμο του ταυτοτικού μητρώου I με το διάνυσμα e το οποίο μας δείνει σαν αποτέλεσμα ένα διάνυσμα.Το πλήθος των πράξεων για αυτόν τον πολ/σμο είναι $2n^2 - n$.(3)
Έπειτα η αφαίρεση που γίνεται κοστίζει n πράξεις γιατί αφαιρούμε 2 διανύσματα.(4) Τέλος υψώνοντας τα στοιχεία του διανύσματος στην p παρατηρούμε πως έχουμε $p - 1$ πολ/σμους και n πράξεις αφού κάθε φορά πολ/ζουμε κάθε στοιχείο του πίνακα με τον εαυτό του οπότε έχουμε $n * (p - 1)$ πράξεις.(5).
Προσθέτουμε τα (1)(2)(3)(4)(5) για να βρούμε το συνολικό αριθμό πράξεων και έχουμε:
$2n^2 + 2n^2 - 2n + n^2$ <=> $n(p - 1) + (5n^2 - n)$.

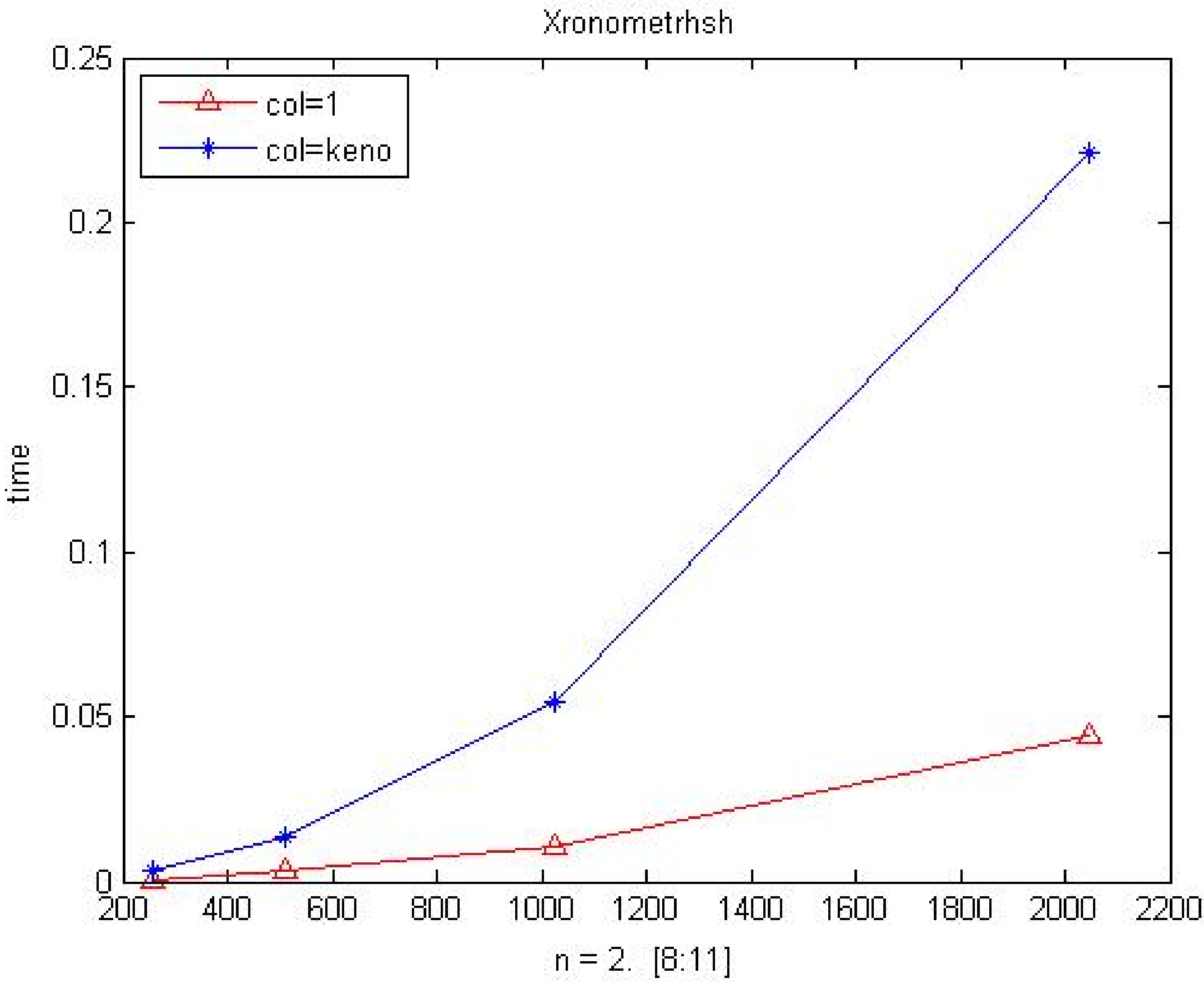
4.3

```
function [F] = myfunc(p,v,u,col)
cols = @(u)size(u,1);%Xrisimopoioume tin size gia na apothikeusoume sto n to mege8os
n = cols(u);          %tou pinaka.
I = eye(n);          %Dimiourgia tautotikou mitrwou mesw tis eye.
TF = isempty(col);   %Xrisimopoioume tin sinartisi isempty gia na doume
if TF == 1;          %se poia periptosi tou col vriskomaste.
    F = (I-u*v').^p;
else
    e = I(:,col);    %Epilegoume tin stili col kai tin apo8ikeuoume sto e.
    F = ((I-u*v')*e).^p;
    end
end
```

4.4

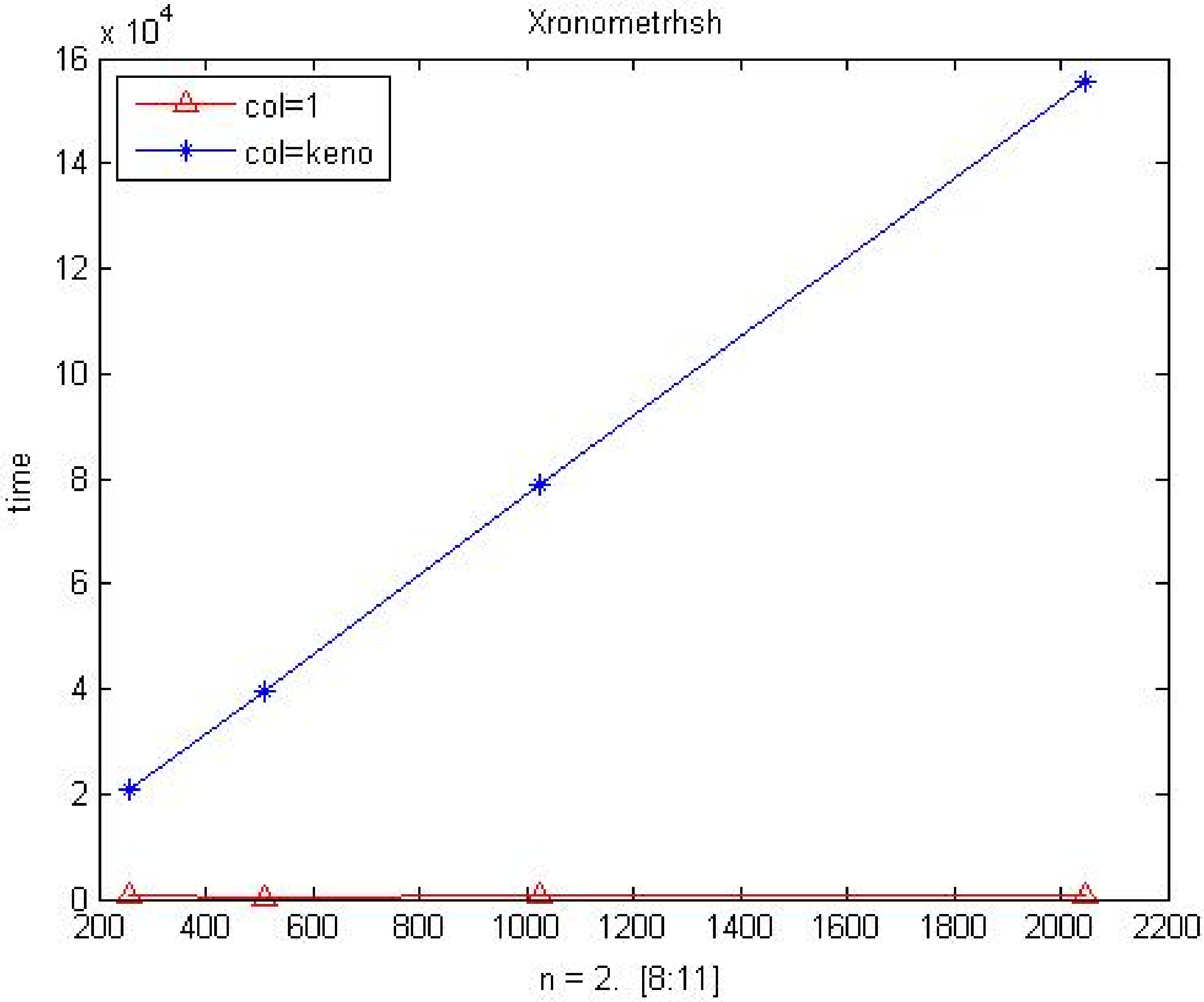
```
n=2.^[8:11];% To n pernei tis times (256 512 1024 2048).
mat_time_timeit = zeros(2,4);%preallocation tou mitroou gia pio grhgorh ektelesh.
mflops = zeros(2,4);%preallocation twn dianusmatwn gia pio grhgorh ektelesh.
for i = [1:4]% 4 epanalipseis gt exoume 4 times apo panw.
    u = randn(n(i),1);%Dimiourgia tyxaiou dianismanos nx1.
    v = randn(n(i),1);%Dimiourgia tyxaiou dianismanos nx1.
    col1 = 1;
    col2 = '';
    p = 3;
    myfunc(p,v,u,col1)
    f = @ () myfunc(p,v,u,col1);
    mat_time_timeit(1,i) = timeit(f)%Klhsh ths timit gia thn xronometrhsh ths myfunc.
                                    %gia to 1o column.
    omega = n(i)*(p-1)+(5*n(i)^2-n(i));%Ypologismos tou omega gia tin euresi twn m.flops.
    mflops(1,i) = (omega/mat_time_timeit(1,i))/10^6;%Ypologismos megaflops kai apothikeusi ...
        tis timit
                                                    %se enan peinaka mflopos(2,4).
    myfunc(p,v,u,col2)
    f = @ () myfunc(p,v,u,col2);
    mat_time_timeit(2,i) = timeit(f)%Klhsh ths timit gia thn xronometrhsh ths myfunc gia ...
        olokliro to mitrwo.
    omega = 2*n(i)^2+n(i)^2*(2*n(i)-1)*(p-1);%Ypologismos tou omega gia tin euresi twn m.flops.
    mflops(2,i) = (omega/mat_time_timeit(2,i))/10^6;%Ypologismos megaflops kai apothikeusi ...
        tis timit
end                                                 %se enan peinaka mflopos(2,4).

figure
plot(n, mat_time_timeit(1,:), '-^r', n, mat_time_timeit(2,:),'-b*' )
legend('col=1','col=keno','Location', 'Northwest')
title('Xronometrhsh ')
xlabel('n = 2. ^ [8:11]')
ylabel('time')

figure
semilogy(n,mflops(1,:),'-^r', n, mflops(2,:),'-b*' )
legend('col=1','col=keno','Location', 'Northwest')
title('Xronometrhsh ')
xlabel('n = 2. ^ [8:11]')
ylabel('time')
```

α)



Xronometrhsh
col=1
col=keno
time
0
0.05
0.1
0.15
0.2
0.25
200
400
600
800
1000
1200
1400
1600
1800
2000
2200
n = 2. [8:11]

β)



Xronometrhsh
x 10^4
16
14
12
10
8
6
4
2
0
time
col=1
col=keno
200
400
600
800
1000
1200
1400
1600
1800
2000
2200
n = 2. [8:11]