

Επιστημονικός Υπολογισμός Αναφορά Εξαμηνιαίας Εργασίας

Μπουραντάς Κωνσταντίνος 1041829 (236145) - 7ο Έτος(Επί πτυχίο)
bourantas@ceid.upatras.gr
students.ceid.upatras.gr/bourantas

Περιεχόμενα

| | |
|--|----|
| Ερώτημα 1: Στοιχεία υπολογιστικού συστήματος | 1 |
| Ερώτημα 2: Αραιή αναπαράσταση BCRS | 3 |
| 2.1 Ερώτημα 2.1 | 3 |
| 2.2 Ερώτημα 2.2 | 4 |
| 2.3 Ερώτημα 2.3 | 6 |
| Ερώτημα 3: Τανυστές και διαδρομές | 7 |
| 3.1 Ερώτημα 3.1 | 7 |
| 3.2 Ερώτημα 3.2 | 8 |
| 3.3 Ερώτημα 3.3 | 8 |
| Ερώτημα 4: Στατιστικά μητρώων | 9 |
| Ερώτημα 5: Επαναληπτικές μέθοδοι | 15 |
| 5.1 Ερώτημα 5.1 | 15 |

Ερώτημα 1

Στοιχεία υπολογιστικού συστήματος

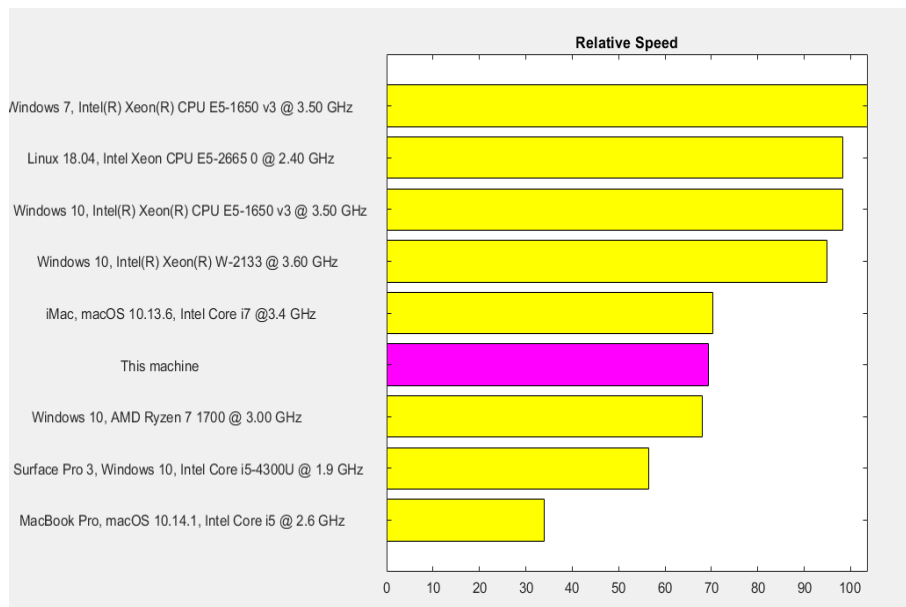
Η παρούσα εργαστηριακή άσκηση πραγματοποιήθηκε σε σύστημα laptop HP ProBook 450 G7. Τα χαρακτηριστικά του υπολογιστικού συστήματος το οποίο χρησιμοποιήθηκε για την υλοποίηση της παρούσας εργαστηριακής άσκησης βρέθηκαν με την χρήση των ακόλουθων προγραμμάτων: [cpuz](#), [speccy](#), της ιστοσελίδας [en.wikichip.org](#) και [ark.intel.com](#) καθώς και μέσω της built-in εφαρμογής System Information των Windows. Επίσης οι εκδόσεις των BLAS, LAPACK και Matlab βρέθηκαν τρέχοντας τις παρακάτω εντολές σε Matlab: "version -blas", "version -lapack", "version" αντίστοιχα.

| Χαρακτηριστικό | Ενδεικτική Απάντηση |
|----------------------|--|
| Έναρξη/Λήξη εργασίας | 14/01/2021 - 22/02/2021 |
| model | HP ProBook 450 G7 |
| O/S | Microsoft Windows 10 Pro |
| processor name | Intel Core i5 10210U |
| processor speed | 1.60GHz |
| number of processors | 1 |
| total cores | 4 |
| total threads | 8 |
| FMA instruction | yes |
| L1 cache | 32 KBytes (per core) |
| L2 cache | 256 KBytes (per core) |
| L3 cache | 6 MBytes (shared) |
| Gflops/s | 243.9 |
| Memory | 16 GBytes |
| Memory Bandwidth | 45.8 GBytes/s |
| Matlab Version | 9.7.0.1190202 (R2019b) |
| BLAS | Intel(R) Math Kernel Library Version 2018.0.3 Product Build 20180406 for Intel(R) 64 architecture applications, CNR branch AVX2 |
| LAPACK | Intel(R) Math Kernel Library Version 2018.0.3 Product Build 20180406 for Intel(R) 64 architecture applications, CNR branch AVX2 Linear Algebra PACKage Version 3.7.0 |

Πίνακας 1.1: Στοιχεία για τα πειράματα

| Computer Type | LU | FFT | ODE | Sparse | 2-D | 3-D |
|--|---------------|---------------|---------------|---------------|---------------|---------------|
| Windows 7, Intel(R) Xeon(R) CPU E5-1650 v3 @ 3.50 GHz | 0.0767 | 0.0979 | 0.0154 | 0.1007 | 0.2420 | 0.2746 |
| Linux 18.04, Intel Xeon CPU E5-2665 0 @ 2.40 GHz | 0.0766 | 0.0969 | 0.0147 | 0.1126 | 0.3538 | 0.2652 |
| Windows 10, Intel(R) Xeon(R) CPU E5-1650 v3 @ 3.50 GHz | 0.0766 | 0.0969 | 0.0147 | 0.1126 | 0.3538 | 0.2652 |
| Windows 10, Intel(R) Xeon(R) W-2133 @ 3.60 GHz | 0.0814 | 0.0870 | 0.0138 | 0.1185 | 0.3117 | 0.3877 |
| iMac, macOS 10.13.6, Intel Core i7 @3.4 GHz | 0.1501 | 0.1354 | 0.0250 | 0.1181 | 0.4019 | 0.3264 |
| This machine | 0.1448 | 0.0810 | 0.0184 | 0.0898 | 0.5770 | 0.7028 |
| Windows 10, AMD Ryzen 7 1700 @ 3.00 GHz | 0.1840 | 0.1441 | 0.0158 | 0.2030 | 0.2959 | 0.3000 |
| Surface Pro 3, Windows 10, Intel Core i5-4300U @ 1.9 GHz | 0.1957 | 0.1764 | 0.0227 | 0.1391 | 0.6722 | 0.4643 |
| MacBook Pro, macOS 10.14.1, Intel Core i5 @ 2.6 GHz | 0.2734 | 0.1980 | 0.0177 | 0.1423 | 2.0198 | 1.3889 |

Σχήμα 1.1: Αποτέλεσμα της bench σε HP ProBook 450 G7.



Σχήμα 1.2: Αποτέλεσμα της bench σε HP ProBook 450 G7.

Ερώτημα 2

Αραιή αναπαράσταση BCRS

2.1 Ερώτημα 2.1

```
1 function [val, col_idx, row_blk] = sp_mx2bcrs(A, nb)
2     %Author: Konstantinos Bourantas, AM: 236145, Date: 14/1/2021
3
4     %Apothikeuoume tis diastaseis tou A
5     [rows, columns] = size(A);
6
7     %Elegxoume an to mitrwo A einai tetragoniko
8     if rows ~= columns
9         error("[-]Error: Input matrix A must be a square matrix.")
10    end
11
12    %Vriskoume mexri posa blocks mporoun na iparxoun se kathe grammi tou A
13    maxBlocksPerRow = rows / nb;
14
15    %Thetoume tis diastaseis tw n blocks pou theloume na xorisoume to mitrwo
16    %A
17    xblockDim = nb * ones(1, maxBlocksPerRow); %poses grammes theloume na
18    %exei to kathe ipomitro
19    yblockDim = nb * ones(1, maxBlocksPerRow); %poses stiles theloume na
20    %exei to kathe ipomitro
21
22    %I sinartisi mat2cell leitourgei os eksis:
23    %Estw A kai xblockDim == [2 2 2 2] kai yblockDim==[2 2 2 2]
24    %paragontai 4 ipomitro me megethos (2,2)
25
26    %Apothikeuoume ta blocks pou prokiptoun apo to mitrwo A
27    blocksMatrix = mat2cell(A, xblockDim, yblockDim);
28
29    val = []; col_idx = []; row_blk = [];
30    blockIdx = 1;
31    %Apothikeuoume posa blocks antistoixoun se kathe grammi
32    blocksPerRow = zeros(maxBlocksPerRow, 1);
33
34    %Diatrexei ola ta blocks tou A pou vriscontai sto mitrwo blocksMatrix
35    for i = 1:size(blocksMatrix, 1)
36
37        for j = 1:size(blocksMatrix, 2)
38
39            %Ean to block periexei mi midenika stoixeia to apothikeuoume sti
40            %metavliti val
41            if (nnz(blocksMatrix{i, j}) > 0)
42                %Topothetoume to kathe block parallila me to allo (row wise)
43                val = [val blocksMatrix{i, j}];
```

```

41         %Apothikeuoume ti stili block pou vrisketai to sigkekrimeno
           block sto mitrwo A
42         col_idx(blockIdx) = j;
43         blockIdx = blockIdx + 1;
44
45         blocksPerRow(i) = blocksPerRow(i) + 1;
46     end
47
48 end
49
50 end
51
52 %To proto block tis protis grammis ipoxrewtika vrisketai stin proti
    thesi
53 %tou mitroou A
54 row_blk(1) = 1;
55
56 %Vriskoume se pio index vrisketai to proto block tw n ipoloipwn grammwn
    sto mitrwo A
57 for i = 2:size(blocksPerRow, 1) + 1
58     row_blk(i) = row_blk(i - 1) + blocksPerRow(i - 1);
59 end
60
61 end

```

Κώδικας 2.1: Κώδικας αρχείου sp_mx2bcrs.m

2.2 Ερώτημα 2.2

```

1 function [y] = spmv_bcrs(y, val, col_idx, row_blk, x)
2     %Author: Konstantinos Bourantas, AM: 236145, Date: 16/1/2021
3
4     %Apothikeuoume tis diastaseis tou pinaka val
5     [rows, columns] = size(val);
6
7     %O sinolikos arithmos apo mi midenika blocks antistoixei sto plithos tw n
        stoixeiwn
8     %tou dianismatos col_idx
9     totalBlocks = size(col_idx, 2);
10    %Ypologizoume tin diastasi nb kathe block
11    nb = columns / totalBlocks;
12
13    %Ypologizoume se posa meri tha spasoume to dianisma x
14    %gia paradeigma an nb==2 kai size(x,1)== 1000 tote vecSize== 500
15    %Ara thelontas na spasoume to x se dianismata me megethos 2 stoixeia
16    vecSize = fix(size(x, 1) / nb)
17
18    %Xorizoume to dianisma x se ipodianismata megethous iso me nb
19    %Gia paradeigma an nb==2 kai size(x,1)== 1000 spame to dianisma se 500
        ipodianismata
20    %mesw tis mat2cell legontas tis oti theloume 500 ipodianismata megethous
        (2,1)
21    x = mat2cell(x, fix(size(x, 1) / vecSize) * ones(1, vecSize), 1);
22
23    blocksPerRowVec = [];
24
25    %Ypologizoume posa blocks exei i kathe seira tou pinaka A
26    for i = 1:size(row_blk, 2) - 1
27        blocksPerRowVec(i) = row_blk(i + 1) - row_blk(i);
28    end

```

```

29
30 i = 1; blockPointer = 1;
31 blockCounter = 0; row = 1;
32 sum = 0;
33 sumVec = [];
34
35 %Gia kathe block tou pinaka val
36 while blockPointer <= totalBlocks
37     %Gia na ipologisoume to y=y+A*x ipologizoume arxika to A*x os eksis:
38     %An gia paradeigma:
39     % A =
40     % 1      2      0      1
41     % 3      4      3      4
42     % 0      0      5      6
43     % 1      0      0      0
44     %
45     % x = [1;2;3;4]
46     %Kai metatrepsoume ton pinaka A se morfi BCRS me blocks 2*2 tote
47     % val =
48     % 1      2      0      1      0      0      5      6
49     % 3      4      3      4      1      0      0      0
50     %
51     % Kai ipologizoume to imiathroisma pou antistoixei se kathe grammi
52     %apo blocks
53     %px. gia tin proti grammi apo blocks exoume
54     %[1 2 ;3 4]*[1;2]+[0 1 ;3 4]*[3;4]
55
56     sum = sum + val(1:nb, i:i + nb -1) * x{col_idx(blockPointer)};
57
58     %Auksanoume to i oste na vrethoume sto epomeno block tou pinaka val
59     i = i + nb;
60
61     %i metavliti blockPointer apothikeuei se pio block vriskomaste
62     blockPointer = blockPointer +1;
63     %i metavliti blockCounter apothikeui posa blocks exoume dei gia
64     %kathe row
65     blockCounter = blockCounter +1;
66
67     %ean ftasame sto teleutaio block enos row pigenoume sto epomeno row
68     if blockCounter == blocksPerRowVec(row)
69         %auksanoume ton metrity row oste na pame stin epomeni grammi
70         row = row +1;
71         %Midenizoume ton metrity twv blocks pou exoume dei gia to
72         %sigkekrimeno row
73         blockCounter = 0;
74         %apothikeuoume to imiathroisma tou proigoumenou row
75         sumVec = [sumVec; sum];
76         %midenizoume to imiathroisma gia to sigkekrimeno row
77         sum = 0;
78     end
79 end
80
81 %ipologizoume to telika apotelesma
82 y = y +sumVec;
83 end

```

Κώδικας 2.2: Κώδικας αρχείου spmv_bcrs.m

2.3 Ερώτημα 2.3

Οι παραπάνω υλοποιήσεις ελέγχθηκαν με την χρήση των μητρών Boeing/bcsstk35, Boeing/bcsstk36. Επειδή το μέγεθος των παραπάνω μητρών ξεπερνάει το διάστημα (1000,2000) επιλέχθηκε μέρος αυτών, έτσι για το πρώτο πήραμε τις πρώτες 1000 γραμμές και στήλες και για το δεύτερο τις πρώτες 2000 γραμμές και στήλες. Για να χρησιμοποιήσουμε την συνάρτηση `spmv_bcrs` τρέχουμε πρώτα `[val, col_idx, row_blk] = sp_max2bcrs(A, 2)` και έπειτα δίνουμε ως είσοδο το αποτέλεσμα της δεύτερης στην πρώτη. Το αποτέλεσμα της συνάρτησης `spmv_bcrs` είναι πολύ κοντά με το αποτέλεσμα που προκύπτει αν τρέξουμε την εντολή $y = y + A \cdot x$ σε Matlab με μικρές αποκλείσεις σε μερικές περιπτώσεις που οφείλονται σε σφάλματα τα οποία προκύπτουν μεταξύ των πράξεων. Πιο συγκεκριμένα για τον έλεγχο των συναρτήσεων θεωρούμε $x = \text{ones}(1000,1)$; $y = \text{zeros}(1000,1)$ και $nb=10$. Έστω ότι αποθηκεύουμε το αποτέλεσμα της `spmv_bcrs` στην μεταβλητή `y_bcrs` και το αποτέλεσμα της εντολής $y = y + A \cdot x$, αν πάρουμε την μέγιστη διαφορά μεταξύ των τιμών των δύο διανυσμάτων που προκύπτουν ως αποτέλεσμα κάνοντας `max(y-y_bcrs)` ισούται με $7.4506e-09$ δηλαδή περίπου 0.00000000745058 . Παρόμοια διαδικασία ελέγχου πραγματοποιήθηκε και για το Boeing/bcsstk36 (με διαστάσεις (2000,2000) , μέγεθος block $nb=10$ και $x = \text{ones}(2000,1)$; $y = \text{zeros}(2000,1)$). Και στην δεύτερη περίπτωση παρατηρούμε ότι υπάρχει διαφορά με το αποτέλεσμα της εντολής $y = y + A \cdot x$ όπου αν δούμε την μέγιστη διαφορά μεταξύ των τιμών των δύο διανυσμάτων που προκύπτουν ως αποτέλεσμα κάνοντας `max(y-y_bcrs)` ισούται με $1.2943e-08$ δηλαδή περίπου 0.000000012943 .

Ερώτημα 3

Τανυστές και διαδρομές

```

1 function [] = erotima3(A, k)
2     %Author: Konstantinos Bourantas, AM: 236145, Date: 18/1/2021
3     %A=[0,1,0,1;1,0,1,0;0,1,0,1;1,0,1,0]
4
5     %erotima1=====
6     G = [];
7
8     %Ftiaxnoume enan polidiastato pinaka kai topothetoume stin triti tou
9     %diastasi stin thesi k to A^k
10    for i = 1:k
11        G(:, :, i) = A^i
12    end
13
14    %Metatrepoume ton polidiastato pinaka G se tensor mesw tou MTT
15    X = tensor(G)
16
17    %To grafima pou antistoixei ston pinaka A
18    % g = graph(A);
19    % plot(g)
20
21    %erotima2=====
22    i = 1; j = 1;
23    %Athroizoume to plithos twn diadromwn metaksi twn komvwn i->j mikous
24    %mexri k
25    sum(G(i, j, 1:k))
26
27    %erotima3=====
28    %Kanoume collapse ton tensor X sto mode 3(tube fibers) mesw tou MTT
29    %se auti tin periptosi i sinartisi collapse athroizei ola ta stoixeia
30    %pou antistoixoun se kathe tube fiber tou kathe tensor:
31    %px. gia komvους (1) kai (2) me k = 3 tote -> X(1,2,1)+X(1,2,2)+X(1,2,3)
32    %etsi o telikos pinakas Y exei tis sinolikes diadromes metaksi kathe
33    %zeugous i,j gia mikos os k
34    Y = collapse(X, 3)
35 end

```

Κώδικας 3.1: Κώδικας αρχείου erotima3.m

3.1 Ερώτημα 3.1

```

1 %erotima1=====
2 G = [];
3
4 %Ftiaxnoume enan polidiastato pinaka kai topothetoume stin triti tou
5 %diastasi stin thesi k to A^k
6 for i = 1:k

```



```
6      G(:, :, i) = A^i
7  end
8
9  %Metatrepoume ton polidiastato pinaka G se tensor mesw tou MTT
10 X = tensor(G)
```

3.2 Ερώτημα 3.2

```
1  %erotima2=====
2  i = 1; j = 1;
3  %Athroizoume to plithos twn diadromwn metaksi twn komvwn i->j mikous
   mexri k
4  sum(G(i, j, 1:k))
```

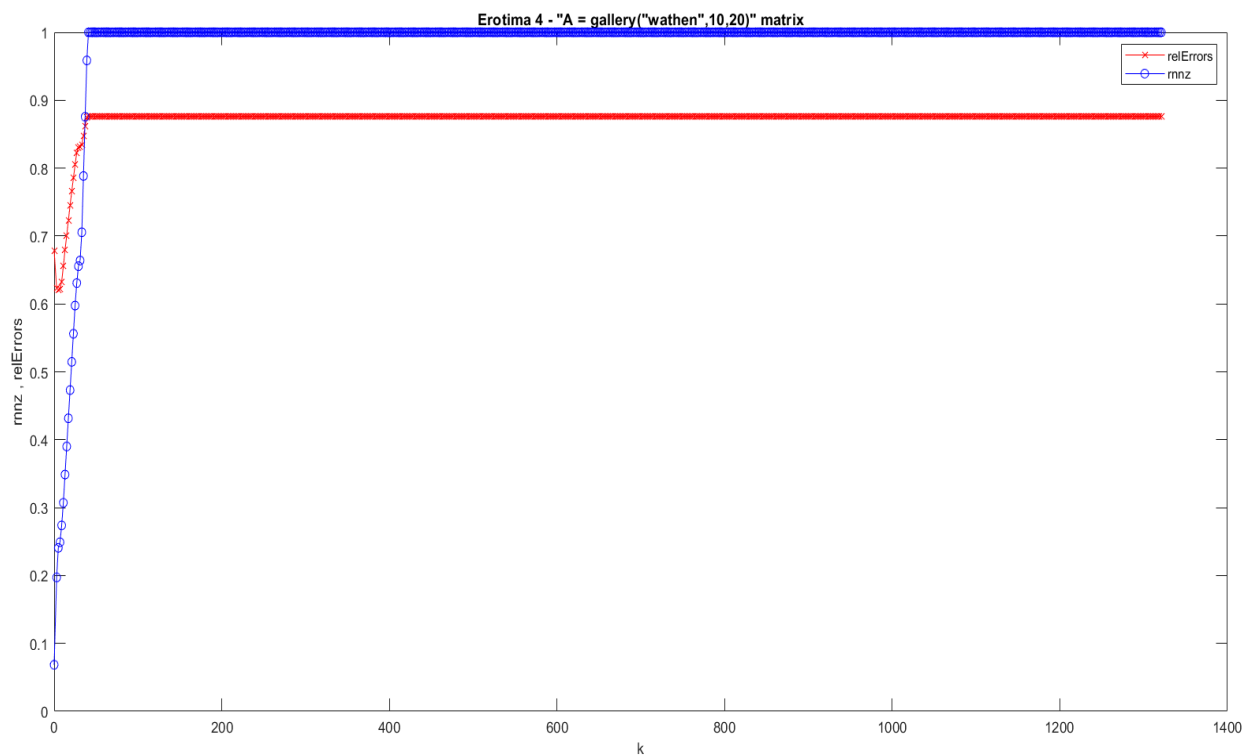
3.3 Ερώτημα 3.3

```
1  %erotima3=====
2  %Kanoume collapse ton tensor X sto mode 3(tube fibers) mesw tou MTT
3  %se auti tin periptosi i sinartisi collapse athroizei ola ta stoixeia
   pou antistoixoun se kathe tube fiber tou kathe tensor:
4  %px. gia komvous (1) kai (2) me k = 3 tote -> X(1,2,1)+X(1,2,2)+X(1,2,3)
5  %etsi o telikos pinakas Y exei tis sinolikes diadromes metaksi kathe
   zeugous i,j gia mikos os k
6  Y = collapse(X, 3)
```

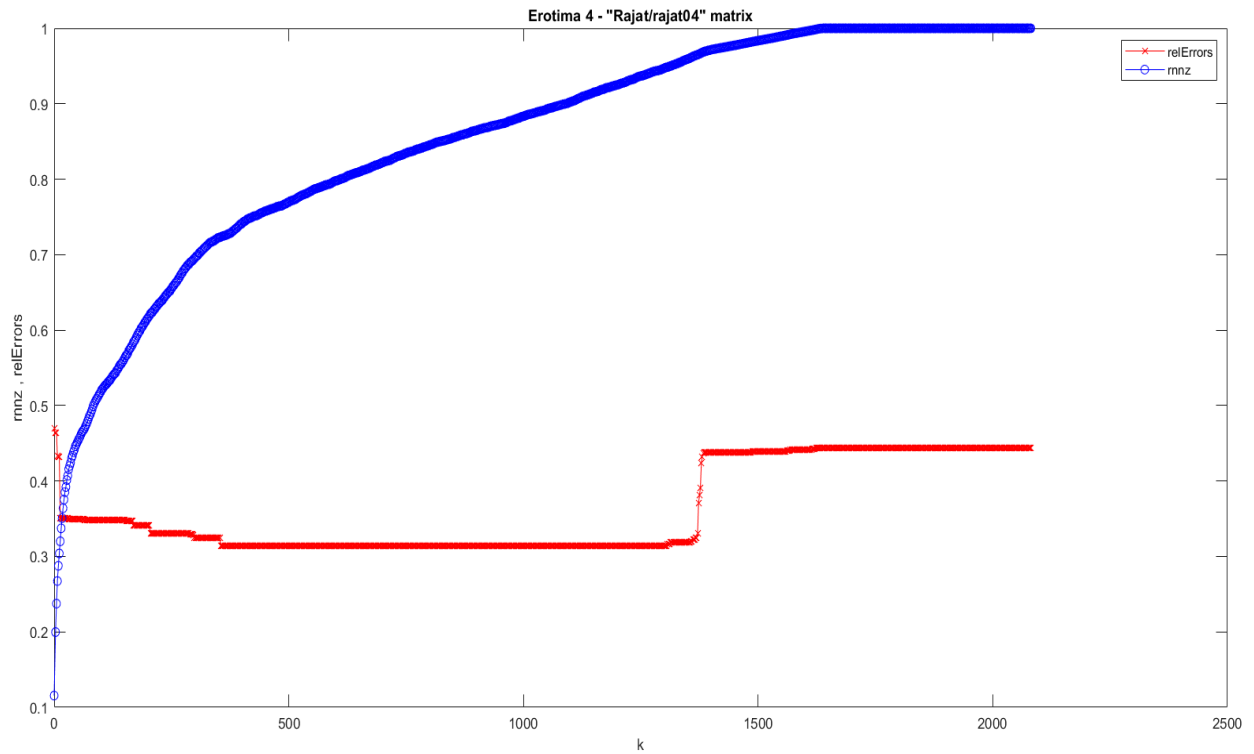
Ερώτημα 4

Στατιστικά μητρώων

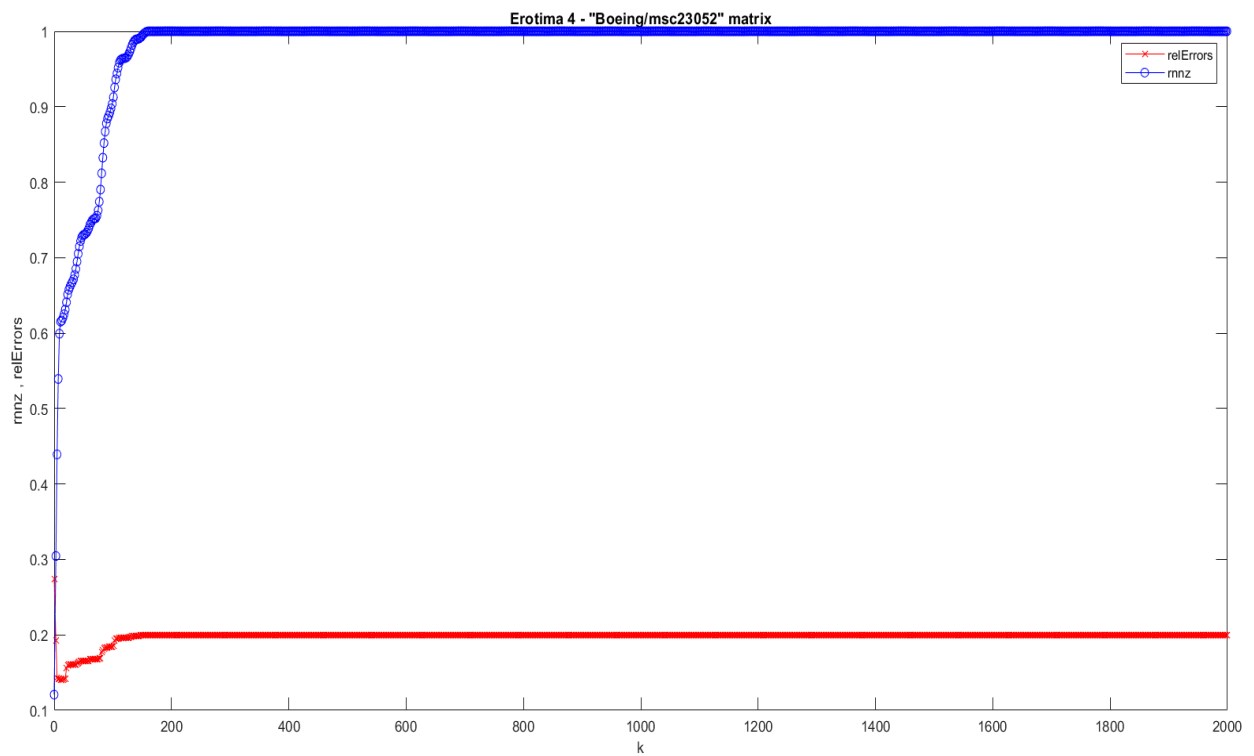
Σύμφωνα με την εκφώνηση τα τελευταία 4 ψηφία του αριθμού μητρώου 236145 είναι 6145 άρα $k = 1 + \text{mod}(6145, 2892) = 362$ άρα το μητρώο A που επιλέγεται είναι το Boeing/msc23052 το οποίο έχει μέγεθος (23052,23052) όποτε θέτουμε $A = A(1:1000, 1:1000)$. Ακολουθούν οι γραφικές παραστάσεις για τα τρία μητρώα ελέγχου:



Σχήμα 4.1: Αποτέλεσμα μετρήσεων για το μητρώο " $A = \text{gallery}(\text{"wathen"}, 10, 20)$ ".



Σχήμα 4.2: Αποτέλεσμα μετρήσεων για το μητρώο "Rajat/rajat04".



Σχήμα 4.3: Αποτέλεσμα μετρήσεων για το μητρώο "Boeing/msc23052"

```

1 function P = band_stats(mxid, p)
2     %Author: Konstantinos Bourantas, AM: 236145, Date: 22/1/2021
3     [rows, columns] = size(mxid);
4
5     rnnz = []; k = []; P = []; relErrors = [];
6
7     if isstring(mxid)
8         mxid = convertStringsToChars(mxid);
9     end
10
11     if isa(mxid, 'char') %An to mxid einai tupou string
12         %Dokimazoume na vroume to mitroo me onoma to string pou periexei to
13         %mxid mesw tis ssget
14         try
15             A = ssget(mxid).A;
16         catch %An to mitrwo me to sigkekrimeno onoma den vrethike
17             %emfanizoume sfalma
18             error('Matrix "%s" not found.', mxid)
19         end
20     else
21         %An to mxid exei arithmo grammwn kai stilwn megalitero apo 1 tote
22         %simainei einai ena mitrwo
23         if or((rows > 1), (columns > 1))
24             A = mxid;
25         elseif isa(mxid, 'double') % An to mxid einai arithmos psaxnoume na
26             %vroume to mitrwo pou antistixei to sigkekrimeno id mesw tis ssget
27             try
28                 A = ssget(mxid).A;
29             catch %An to mitrwo me to sigkekrimeno onoma den vrethike
30                 %emfanizoume sfalma
31                 error('Matrix "%d" not found.', mxid)
32             end
33         end
34     end
35
36     %Ean se auto to simeio prokipsei sfalma simenei oti to mitrwo A den
37     %arxikopoihthike sosta
38     try
39         [rows, columns] = size(A);
40     catch
41         error('Failed to load matrix A.')
42     end
43
44     %p = rows;
45
46     %0 arithmos tw'n mi midenikwn stoxeiwn tou A
47     nnz_A = nnz(A);
48
49     % Apothikeuoume oles tis diagonious tou mitrwou A os stiles ston pinaka
50     %diags
51     % xrisimopoiountas tin sinartisi spdiags. I sinartisi spdiags epistrefei
52     % kai ena index gia kathe
53     % diagonio kai to apothikeuei sto mitrwo diagsIndex gia paradeigma:
54     %
55     %A =

```

```

53 %
54 % 1      0      0      0
55 % 1      1      0      0
56 % 1      1      1      0
57 % 1      1      1      1
58 %diags =
59 %
60 % 1      1      1      1
61 % 0      1      1      1
62 % 0      0      1      1
63 % 0      0      0      1
64 %
65 %Οι diagonalioi pou vriskontai katw apo tin kiria diagonalio exoun arnitika
    indexes
66 %enw oi diagonalioi panw apo tin kiria diagonalio thetika. Episis i kiria
    diagonalios exei index 0.
67 % diagsIndex =
68 %-3
69 %-2
70 %-1
71 % 0
72 %
73 % Me auton ton tropo mporoume na gnorizoume pou vrisketai i kathe
    diagonalios ston pinaka A p.x
74 % I proti stili tou pinaka diags antistoixei to index -3 pou simenei oti
    einai i pio xamili ipodiagonalios tou pinaka A
75 [diags, diagsIndex] = spdiags(A);
76
77 %Apothikeuoume tin diagonalio tou pinaka A
78 mainDiag = diag(A);
79
80 %Arxikopoioyme ena pinaka B me midenika kai idies diastaseis me ton
    pinaka A
81 B = zeros(rows, columns);
82 mainDiagIndex = -1;
83
84 %0 arithmos tw n ipodiagwniwn
85 downDiagsNum = 0;
86 %0 arithmos tw n iperdiagwniwn
87 upDiagsNum = 0;
88 %downDiagIndex: To index tis protis ipodiagonaliou ston pinaka diags
89 downDiagIndexConst = -1;
90 %upDiagIndex: To index tis protis iperdiagonaliou ston pinaka diags
91 upDiagIndexConst = -1;
92
93 %Diatrexeume oles tis diagonalios pou vriskontai os stiles ston pinaka
    diags
94 for i = 1:size(diags, 2)
95
96     % Ean to index isoutai me miden einai i kiria diagonalios
97     if diagsIndex(i) == 0
98         mainDiagIndex = i;
99
100     elseif diagsIndex(i) < 0 % An index < 0 exoume ipodiagonalio
101         %Apothikeuoume to teleutaio arnitiko index pou simatodotei tin
            thesi tis protis
102         %ipodiagonaliou ston pinaka diagsIndex kai sinepws ston pinaka
            diags
103         downDiagsNum = downDiagsNum + 1;
104         downDiagIndexConst = i;
105

```

```

106     else %An index > 0 exoume iperdiagonio
107         upDiagsNum = upDiagsNum +1;
108         %Apothikeuoume to proto thetiko index pou simatodotei tin thesi
            tis protis
109         %iperdiagoniou ston pinaka diagsIndex kai sinepws ston pinaka
            diags
110         if (upDiagIndexConst == -1)
111             upDiagIndexConst = i;
112         end
113
114     end
115
116 end
117
118 %Ftiaxnoume ena mitrwo bandMatrix to opio periexei mono stin kiria
    diagonio tou A
119 bandMatrix = spdiags(mainDiag, 0, B);
120
121 bandwidth = 0; dimCheck = 0;
122
123 while bandwidth < 2 * p + 1
124     k = [k bandwidth];
125     dimCheck = dimCheck +1;
126
127     %Se periptosi pou to P ksepernaei tis diastaseis tou A
128     %epistrefoume ton pinaka bandMatrix me oles tis dinates diagonious
129     if dimCheck <= size(bandMatrix, 2)
130         diagsNum = fix(bandwidth / 2);
131     end
132
133     if bandwidth == 0
134         bandwidth = 1;
135     end
136
137     upIndex = 1;
138     upDiagIndex = upDiagIndexConst;
139
140     %An exoume iperdiagonious
141     if upDiagsNum > 0
142
143         %Dietrekse oles tis iperdiagonious me vasi to zitoumeno
            bandwidth
144         for i = 1:diagsNum
145
146             %Vriskoume tin sigkekrimeni diagonio
147             tempDiag = diags(1:rows, upDiagIndex);
148
149             %Ean i diagonios periexei mi midenika stoixeia
150             if nnz(tempDiag) > 0
151                 %Apothikeuoume tin diagonio tempDiag sti thesi pou
                    antistoixei sto index poy deixnei i metavliti
152                 %upIndex kai tin apothikeuoume sto bandMatrix
153                 bandMatrix = spdiags(tempDiag, upIndex, bandMatrix);
154             end
155
156             %Pigenoume stin amesws epomeni iperdiagonio
157             upIndex = upIndex +1;
158             upDiagIndex = upDiagIndex + 1;
159
160             if (upDiagIndex > size(diags, 2))
161                 break

```

```

162         end
163
164     end
165
166 end
167
168     downIndex = -1;
169     downDiagIndex = downDiagIndexConst;
170
171     %An exoume ipodiagonious
172     if downDiagsNum > 0
173
174         for i = 1:diagsNum
175             %Vriskoume tin sigkekrimeni diagonio
176             tempDiag = diags(1:rows, downDiagIndex);
177
178             %Ean i diagonios periexei mi midenika stoixeia
179             if nnz(tempDiag) > 0
180                 %Apothikeuoume tin diagonio tempDiag sti thesi pou
181                 %antistoixei sto index poy deixnei i metavliti
182                 %downIndex kai tin apothikeuoume sto bandMatrix
183                 bandMatrix = spdiags(tempDiag, downIndex, bandMatrix);
184             end
185
186             %Pigenoume stin amesws epomeni ipodiagonio
187             downIndex = downIndex - 1;
188             downDiagIndex = downDiagIndex - 1;
189
190             if (downDiagIndex < 1)
191                 break
192             end
193         end
194
195     end
196
197     %full(bandMatrix)
198
199     %To plithos tw'n mi midenikwn stoixeiwn tou bandMatrix
200     % se sxesi me to mitrwou A.
201     rnnz = [rnnz; nnz(bandMatrix) / nnz_A];
202     % To sxetika sfalma os pros tin norma Frobenius
203     relErrors = [relErrors; norm(A - bandMatrix, 'fro') / norm(A, 'fro')
204                 ];
205
206     bandwidth = bandwidth + 2;
207
208     P = [rnnz relErrors];
209
210 end

```

Κώδικας 4.1: Κώδικας αρχείου band_stats.m

Ερώτημα 5

Επαναληπτικές μέθοδοι

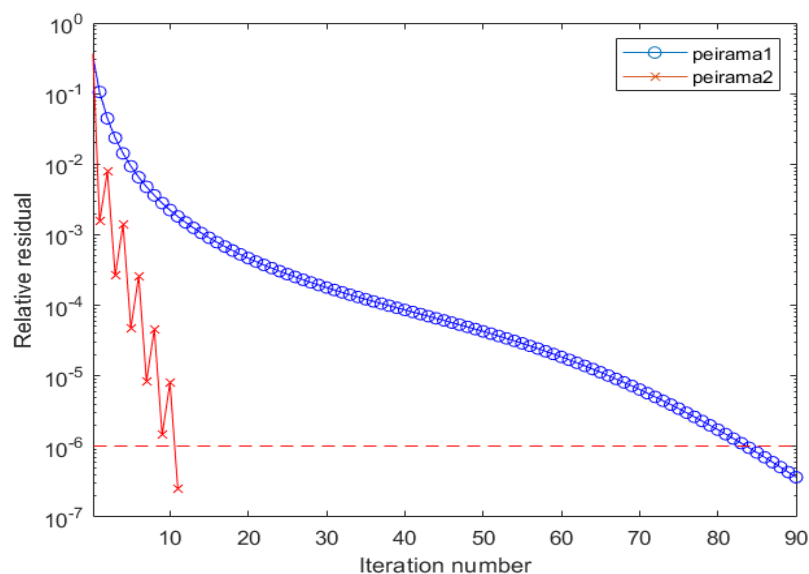
5.1 Ερώτημα 5.1

Υποερώτημα 5.1.1

Παρατηρώντας τα αποτελέσματα του πρώτου πειράματος βλέπουμε ότι η επαναληπτική μέθοδος PCG συγκλίνει. Επιπλέον παρατηρούμε ότι το σχετικό κατάλοιπο της λύσης (relative residual) στην τελευταία επανάληψη έχει μειωθεί αισθητά κάτι το οποίο σημαίνει ότι έχουμε πετύχει μια αρκετά καλή ακρίβεια στα αποτελέσματά μας. Επίσης παρατηρούμε ότι η ταχύτητα της σύγκλισης δεν είναι πολύ μεγάλη καθώς η επαναληπτική μέθοδος PCG τερματίζει μετά από 90 επαναλήψεις. Επίσης το τελικό σχετικό κατάλοιπο της λύσης ισούται με 8.9790×10^{-7} .

Υποερώτημα 5.1.2

Παρατηρώντας τα αποτελέσματα του δεύτερου πειράματος βλέπουμε ότι η επαναληπτική μέθοδος PCG και σε αυτή την περίπτωση συγκλίνει. Επιπλέον παρατηρούμε ότι το σχετικό κατάλοιπο της λύσης (relative residual) στην τελευταία επανάληψη έχει μειωθεί αισθητά κάτι το οποίο σημαίνει ότι έχουμε πετύχει μια αρκετά καλή ακρίβεια στα αποτελέσματά μας. Το σχετικό κατάλοιπο είναι μικρότερο από ότι στο πρώτο μας πείραμα και ισούται με 2.5204×10^{-7} . Επίσης στην συγκεκριμένη περίπτωση η επαναληπτική μέθοδος PCG συγκλίνει πολύ γρήγορα προς το αποτέλεσμα και τερματίζει μετά από 11 επαναλήψεις. Τέλος, βλέπουμε ότι η διακύμανση του σχετικού κατάλοιπου ακολουθεί ένα συγκεκριμένο μοτίβο όπου αφού τείνει να μειωθεί μετά ξανά αυξάνεται και μειώνεται πάλι κάτι το οποίο διαφέρει από το πρώτο πείραμα.



Σχήμα 5.1: Αποτέλεσμα μετρήσεων για τα πειράματα 5.1(Τελικό σχετικό κατάλοιπο: 8.9790×10^{-7}) και 5.2(Τελικό σχετικό κατάλοιπο: 2.5204×10^{-7}).

Υποερώτημα 5.1.3

Παρατηρώντας τα αποτελέσματα των δύο πειραμάτων βλέπουμε ότι η επαναληπτική μέθοδος PCG παρουσιάζει μεγαλύτερη ταχύτητα σύγκλισης στο δεύτερο πείραμα από ότι στο πρώτο. Στην πρώτη περίπτωση η επαναληπτική μέθοδος PCG, ο αλγόριθμος τερματίζει μετά από 90 επαναλήψεις ενώ στην δεύτερη περίπτωση μετά από 11 επαναλήψεις. Επίσης στο δεύτερο πείραμα έχουμε μικρότερο σχετικό κατάλοιπο στο τέλος της μεθόδου από ότι στο πρώτο κάτι το οποίο σημαίνει ότι έχουμε επιτύχει μεγαλύτερη ακρίβεια στο αποτέλεσμα μας. Επιπλέον το σχετικό κατάλοιπο μειώνεται πολύ πιο γρήγορα στο δεύτερο πείραμα από ότι στο πρώτο. Και στις δύο περιπτώσεις το μητρώο A είναι ένα αραιό μητρώο το οποίο περιέχει μη μηδενικά στοιχεία μόνο στην κύρια διαγώνιο. Στην πρώτη περίπτωση το μητρώο A(A(1:10,1:10)) έχει την μορφή:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 \end{bmatrix}$$

Ενώ στην δεύτερη περίπτωση έχουμε:

$$\begin{bmatrix} 1.0000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.0040 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0080 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0120 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0161 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0201 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.0241 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0281 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0321 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0361 \end{bmatrix}$$

Στο πρώτο πείραμα το μητρώο A έχει δείκτη κατάστασης ο οποίος ισούται με 500 ενώ στο δεύτερο πείραμα ο δείκτης κατάστασης του A ισούται με 1001. Ο δείκτης κατάστασης ενός μητρώου συνδέεται με την απώλεια των δεκαδικών ψηφίων, όσο μεγαλύτερη είναι η τιμή του τόσο περισσότερα ψηφία χάνουμε και συνεπώς τόσο μειώνεται και η ταχύτητα της σύγκλισης προς την τελική λύση του συστήματος Ax=b εφόσον δημιουργείται μεγαλύτερο σφάλμα. Επομένως θα αναμέναμε να είχαμε ταχύτερη σύγκλιση στην πρώτη περίπτωση από ότι στην δεύτερη ωστόσο αυτό δεν συμβαίνει.

```

1 %Author: Konstantinos Bourantas, AM: 236145, Date: 21/2/2021
2 %erotima
   5.1-----

3 n = 500;
4 tol = 1e-6;
5
6 %Ektelesi tou protou peiramatos 5.1.1
7 A = spdiags([1:n]', [0], n, n);
8 xsol = ones(n, 1);
9 b = A * xsol;
10
11 [x, fl1, rr1, it1, rv1] = pcg(A, b, tol, 4 * n);
12
13 %Ektelesi tou deuteron peiramatos 5.1.2
14 A = spdiags([linspace(1, 2, n / 2)'; linspace(1000, 1001, n / 2)'], [0], n,
   n);
15 xsol = ones(n, 1);
16 b = A * xsol;
17

```

```
18 [x, fl2, rr2, it2, rv2] = pcg(A, b, tol, 4 * n);
19
20 %Ftiaxnoume tin grafiki anaparastasi twm dio peiramatwn simfwna me tin
    ekfonisi
21 semilogy(0:length(rv0) - 1, rv0 / norm(b), 'b-o'); hold on;
22 semilogy(0:length(rv2) - 1, rv2 / norm(b), 'r-x'); hold on;
23
24 yline(tol, 'r--');
25 legend('peirama1', 'peirama2')
26 xlabel('Iteration number')
27 ylabel('Relative residual')
```

Κώδικας 5.1: Κώδικας αρχείου erotima5.m