

Επιστημονικός Υπολογισμός

Εργαστηριακή Άσκηση Μέρος 1

Ανδρονίκη Πορτοκάλογλου
1067539
4ο Έτος

Περιεχόμενα

1. Εισαγωγικά

1.1 Στοιχεία υπολογιστικού συστήματος

2. Αραιές αναπαραστάσεις και κατασκευές μητρώων

2.1 Άσκηση 1

2.2 Άσκηση 2

2.3 Άσκηση 3

2.4 Άσκηση 4

2.5 Άσκηση 5

1 Εισαγωγικά

1.1 Στοιχεία υπολογιστικού συστήματος

1.

Ημερομηνία έναρξης ενασχόλησης με την άσκηση : 17/11/2021

Ημερομηνία λήξης της ενασχόλησης με την άσκηση : 10/01/2021

2.

(i)

model	προσωπικό λαπτοπ Dell Vostro 15-5568
O/S	Windows 10.0
processor name	Intel Core i7 7500U
processor speed	2.7GHz
number of processors	1
total # thead	2
total # cores	4
FMA instruction	yes
L1 cache	128 KiB L1I\$ 64 KiB 2x32 KiB 8-way set associative L1D\$ 64 KiB 2x32 KiB 8-way set associative write-back
L2 cache	512 KiB 2x256 KiB 4-way set associative write-back
L3 cache	4 MiB 2x2 Mi 16-way set associative write-back
Gflops/s	403.2
Memory	8GB
Memory Bandwidth	2133 MHZ
MATLAB Version	9.11.0.1811744 (R2021b) Update 1
BLAS	Intel(R) Math Kernel Library Version 2019.0.3 for Intel(R) 64 architecture applicationsCNRbranch AVX512
LAPACK	Intel(R) Math Kernel Library Version 2019.0.3 for Intel(R) 64 architecture applications CNR branch AVX512,Linear Algebra PACKage (LAPACK 3.7.0)

(ii)

Έκδοση MATLAB Version 9.11(R2021b) (Χρησιμοποιήθηκε η matlab online μέσω mathworks) δεν έχει γίνει χρήση κάποιας βιβλιοθήκης.

(iii)

Το αποτέλεσμα της εντολής bench.

MATLAB Benchmark (times in seconds)						
Computer Type	LU	FFT	ODE	Sparse	2-D	3-D
Windows 10, AMD Ryzen Threadripper(TM) 3970x @ 3.50 GHz	0.2008	0.1881	0.3469	0.4396	0.2029	0.1117
Debian 10(R), AMD Ryzen Threadripper 2950x @ 3.50 GHz	0.3122	0.2377	0.3219	0.5047	0.5941	0.1631
iMac, macOS 11.2.3, Intel Core i9 @3.6 GHz	0.3278	0.2648	0.2674	0.2763	0.6898	0.3946
Windows 10, Intel Xeon(R) W-2133 @ 3.60 GHz	0.4154	0.2991	0.4348	0.4574	0.3167	0.2184
Windows 10, Intel Xeon CPU E5-1650 v3 @ 3.50 GHz	0.4614	0.3030	0.4455	0.4433	0.3559	0.2623
Windows 10, AMD Ryzen(TM) 7 1700 @ 3.00 GHz	0.7507	0.5163	0.4884	0.5441	0.3397	0.1849
This machine	0.5749	0.3013	0.4680	0.6091	1.1511	0.2736
Windows 10, Intel Core i7-10610 @ 1.8 GHz	0.9218	0.4394	0.3666	0.3844	0.7386	0.6251
Surface Pro 3, Windows(R) 10, Intel(R) Core(TM) i7-5600U @ 2.6 GHz	1.7475	0.9090	0.6178	0.5711	0.5713	0.3623
MacBook Pro, macOS 10.15.2, Intel Core i5 @ 2.6 GHz	1.6237	0.9786	0.5446	0.6173	2.5214	2.0229

Place the cursor near a computer name for system and version details. Before using this data to compare different versions of MATLAB, or to download an updated timing data file, see the help for the bench function by typing "help bench" at the MATLAB prompt.

2 Αραιές αναπαραστάσεις και κατασκευές μητρώων

2.1 Άσκηση 1

Στη συνάρτηση αυτή ανάλογα με το `sp_type` εκτυπώνεται η αντίστοιχη μορφή. Αρχικά στο `val` αποθηκεύονται οι μη μηδενικοί αριθμοί του `A` και ανάλογα με το ποια μορφή έχει ζητήσει γίνεται ανά στήλες ή ανά σειρές. Έπειτα μέσω του `find` βρίσκω το `IA` και ανάλογα με την μορφή που έχει δοθεί στη είσοδο χρησιμοποιείται το `A` ή `AT`. Το `JA` βρίσκεται με τη παράγωγο της στήλης που ανήκει κάθε στοιχείο. Αφού έχουν βρεθεί όλοι οι απαραίτητοι πίνακες γίνεται `display` της σύνταξης των πινάκων σε `latex`.

```

1 function [val,JA,IA]=spmat2latex(A,sp_type)
2 %Author: ΑΠΟΡΤΟΚΑΛΟΓΛΟΥΤ. , AM:1067539 ,DATE:20/11/21
3 if sp_type == 'csr'
4     val= (nonzeros(A.')).'
5     [IA, col_ip]=find( A.' )
6     IA=IA.'
7     dCid=diff(col_ip)
8     answer=find(dCid)+1
9     JA=[1 answer.' length(val)+1]
10    disp(['$$ val= \begin { tabular } {',repmat('|1',1,length(val)), '|}\ hline',
11          num2str(val),'\\ \hline \end { tabular } $$'])
12    disp(['$$ JA= \begin { tabular } {',repmat('|1',1,length(JA)), '|}\ hline',
13          num2str(JA),'\\ \hline \end { tabular } $$'])
14    disp(['$$ IA= \begin { tabular } {',repmat('|1',1,length(IA)), '|}\ hline',
15          num2str(IA),'\\ \hline \end { tabular } $$'])
16 else
17     val= (nonzeros(A)).'
18     [IA, cId]=find( A )
19     dCid=diff(cId.')
20     answer=find(dCid)+1
21     JA=[1 answer length(val)+1]
22     val= (nonzeros(A)).'
23     disp(['$$ val= \begin { tabular } {',repmat('|1',1,length(val)), '|}\ hline',
24           num2str(val),'\\ \hline \end { tabular } $$'])
25     disp(['$$ JA= \begin { tabular } {',repmat('|1',1,length(JA)), '|}\ hline',
26           num2str(JA),'\\ \hline \end { tabular } $$'])
27     disp(['$$ IA= \begin { tabular } {',repmat('|1',1,length(IA)), '|}\ hline',
28           num2str(IA.').'\ \hline \end { tabular } $$'])
29 end
30 end

```

2.2 Άσκηση 2

Η συνάρτηση blkToeplitzTrid αρχικά δημιουργεί ένα τετραγωνικό πίνακα d διαστάσεων $m \times n$ με μηδενικά. Έπειτα, μέσω μιας for από το 1 έως το n που είναι το μήκος της διαγώνιου τοποθετεί το πίνακα A στη διαγώνιο. Μέσα στην ίδια for τοποθετούνται και οι πίνακες B και C ωστόσο πρέπει να ελέγξουμε ότι δεν θα ξεφύγουν από τα όρια του πίνακα οπότε σε κάθε περίπτωση πριν την αρχειοθέτηση του καθένα τοποθετείται μια if.

```

1 function blkToeplitzTrid(m,n,B,A,C)
2 %Author: ΑΠΟΡΤΟΚΑΛΟΓΛΟΥΤ. , AM:1067539 ,DATE:28/11/21
3
4 d=zeros(m*n)
5 for i=1:n
6
7
8     d(1+(i-1)*m:m+(i-1)*m,1+(i-1)*m:m+(i-1)*m)= A
9     if (m+(i-1)*m+m )<=m*n
10    d(1+(i-1)*m:m+(i-1)*m,1+(i-1)*m+m:m+(i-1)*m+m)= C
11    end
12
13    if (1+(i-1)*m-m)>=1
14    d(1+(i-1)*m:m+(i-1)*m,1+(i-1)*m-m:m+(i-1)*m-m)= B
15    end
16
17
18 end
19
20 end

```

2.3 Άσκηση 3

Για να υλοποιηθεί η συνάρτηση `sp_mx2bcs` αρχικά χωρίζεται ο πίνακας `A` σε blocks σύμφωνα με το `nb`. Μέσα σε δυο `for` οι οποίες διαπερνάνε όλα τα block στα οποία έχει σπάσει ο πίνακας `A` για κάθε block το οποίο δεν αποτελείται μόνο από μηδενικά γεγονός το οποίο ελέγχεται μέσω μιας `if` τοποθετούνται τα στοιχεία του block ανα στήλες και έτσι δημιουργείται το `val`. Για κάθε block που συμμετέχει στο `val` κρατάμε το `i` το οποίο μας δείχνει σε ποια block σειρά βρίσκεται έτσι ώστε να δημιουργηθεί το `brow_idx`. Μέσα στις δυο `for` αυτές κρατάμε για κάθε block που συμμετέχει στο `val` και την `j` διάσταση του κάθε block στο `d` και ένα `count` το οποίο μας δείχνει ποσά block πήραμε από όλο το πίνακα. Έπειτα, χρησιμοποιώντας το `diff` του `d` και το `find` φτιάχνουμε το `bcol_ptr` και βάζουμε ως τελευταίο στοιχείο το `count+1`.

```

1 function [val,brow_idx,bcol_ptr]= sp_mx2bcs(A,nb)
2 %Author: ΑΠΟΡΤΟΚΑΛΟΓΛΟΥ. , AM:1067539 ,DATE:3/12/21
3
4 val=[]
5 brow_idx=[]
6 d=[]
7 z=(ones(1,size(A,2)/nb))*nb
8
9 c = mat2cell(A, [ z ],[ z])
10 celldisp(c)
11
12 count=0
13 for j=1:size(A,2)/nb
14     for i=1:size(A,2)/nb
15
16         if sum(abs(c{i,j}(:)~=0))
17             newval=(c{i,j}(:).')
18             val = [val newval];
19             brow_idx=[brow_idx i]
20             d=[d j]
21             count=count+1
22         end
23     end
24 end
25 n=diff(d)
26 answer=find(n)+1
27 bcol_ptr=[1 answer count+1]
28 end

```

2.4 Άσκηση 4

Η συνάρτηση `spmv_bccs` δημιουργεί ένα `y_dot` στο μέγεθος του `y` που δίνεται στην είσοδο. Παίρνει το `val` και το κάνει σε blocks των `nb*nb` και την παραγωγή του `bcol_ptr`. Δημιουργείται μια `for` από το ένα έως το μήκος της παραγωγού `bcol_ptr`, πριν από την οποία αρχικοποιούνται 2 μετρητές ο `col` και ο `index`. Η παράγωγος `bcol_ptr` δείχνει ποσό έχει η κάθε στήλη οπότε φτιάχνετε άλλη μια `for` από το 1 έως την κάθε τιμή της παραγωγού. Στις δυο αυτές εμφωλευμένες `for` παίρνουμε ένα ένα τα blocks του `val` την σειρά τους μέσω του `brow_idx(index)` και στο `y_dot` κάνουμε την πράξη του πολλαπλασιασμού του κάθε block με το `x` (το σωστό στοιχείο του `x` κάθε φορά) βάζοντας το αποτέλεσμα στη σωστή θέση του `y_dot` και αθροίζοντας με το προηγούμενο αριθμό που μπορεί να έχει το `y_dot` μέσα. Αφού έχουμε κάνει τον πολλαπλασιασμό προσθέτουμε στο `y_dot` το `y` και το αποθηκεύουμε στο `y`.

```

1 function [y]= spmv_bccs(y,x,nb,val,brow_idx,bcol_ptr)
2 %Author: ΑΠΟΡΤΟΚΑΛΟΓΛΟΥ. , AM:1067539 ,DATE:5/12/21
3
4 y_dot= zeros(size(y,1),1)
5
6 C = reshape(val,nb,nb,length(val)/(nb*nb))
7
8 d=diff(bcol_ptr)
9
10 index=1
11 col=0
12 for i= 1:length(d)
13     k=d(i)
14     for j=1:k
15         block=C(:, :, index)
16         row=brow_idx(index)
17
18         y_dot(nb*row-1:(row)*nb,1)=y_dot(nb*row-1:(row)*nb,1)+block*x(nb*col+1:(col+1)*
19             nb,1)
20
21         index=index+1
22     end
23     col=col+1
24 end
25 y=y+y_dot
26 end

```


2.5 Άσκηση 5

Αρχικά, δημιουργείται ένα μητρώο T 32×32 έπειτα μέσω της συνάρτησης `blkToeplitzTrid` με είσοδο τρία μητρώα το A , B και το T . Το A είναι αντίστροφο του T και το B είναι το T στο τετράγωνο όλα έχουν διάσταση 32×32 . Έτσι, υπολογίζουμε το μητρώο S (2048×2048). Μέσω της κλήσης `[val,brow_idx,bcol_ptr]= sp_mx2bcss(S,nb)` βρίσκεται η μορφή `bcss` του S ενώ με το `[y]=spmv_bcss(y,x,nb,val,brow_idx,bcol_ptr)` υπολογίζεται το y (2048×1). Στη συνέχεια, υπολογίζεται το $y1$ κάνοντας την πράξη κατευθείαν και υπολογίζεται η νόρμα 2 του y και του $y1$. Τελος για να βρεθεί το σφάλμα τα αφαιρούμε.

```

1 %Author: ΑΠΟΡΤΟΚΑΛΟΓΛΟΤ. , AM:1067539 ,DATE:15/12/21
2 nb=m
3 n=64
4 y=eye(n*m,1)
5 x=ones(n*m,1)
6 T=toeplitz([4,-1,zeros(1,m-2)])
7 A=inv(T)
8 B=T^2
9 S=blkToeplitzTrid(32,64,A,B,T)
10 [val,brow_idx,bcol_ptr]= sp_mx2bcss(S,nb)
11 [y]=spmv_bcss(y,x,nb,val,brow_idx,bcol_ptr)
12 l2y = sqrt(sum(abs(y).^2))
13 y1=y+S*x
14 l2y1 = sqrt(sum(abs(y1).^2))
15 error=l2y1-l2y
16
17
18 function [d]=blkToeplitzTrid(m,n,B,A,C)
19 %Author: ΑΠΟΡΤΟΚΑΛΟΓΛΟΤ. , AM:1067539 ,DATE:28/11/12
20
21 d=zeros(m*n)
22 for i=1:n
23
24
25     d(1+(i-1)*m:m+(i-1)*m,1+(i-1)*m:m+(i-1)*m)= A
26     if (m+(i-1)*m+m )<=m*n
27     d(1+(i-1)*m:m+(i-1)*m,1+(i-1)*m+m:m+(i-1)*m+m)= C
28     end
29
30     if (1+(i-1)*m-m)>=1
31     d(1+(i-1)*m:m+(i-1)*m,1+(i-1)*m-m:m+(i-1)*m-m)= B
32     end
33
34
35 end
36
37 end
38 function [y]= spmv_bcss(y,x,nb,val,brow_idx,bcol_ptr)
39 %Author: ΑΠΟΡΤΟΚΑΛΟΓΛΟΤ. , AM:1067539 ,DATE:5/12/12
40
41 y_dot= zeros(size(y,1),1)
42
43 C = reshape(val,nb,nb,length(val)/(nb*nb))
44
45 d=diff(bcol_ptr)
46
47 index=1
48 col=0
49 for i= 1:length(d)
50     k=d(i)
51     for j=1:k

```

```

52     block=C(:, :, index)
53     row=brow_idx(index)
54
55     y_dot(nb*row-1:(row)*nb,1)=y_dot(nb*row-1:(row)*nb,1)+block*x(nb*col+1:(col+1)*
    nb,1)
56
57     index=index+1
58     end
59     col=col+1
60 end
61
62 y=y+y_dot
63 end
64
65 function [val,brow_idx,bcol_ptr]= sp_mx2bccs(A,nb)
66 %Author: ΑΠΟΡΤΟΚΑΛΟΓΛΟΥ. , AM:1067539 ,DATE:3/11/12
67
68 val=[]
69 brow_idx=[]
70 d=[]
71 z=(ones(1,size(A,2)/nb))*nb
72
73 c = mat2cell(A, [ z ],[ z])
74 cellldisp(c)
75
76 count=0
77 for j=1:size(A,2)/nb
78     for i=1:size(A,2)/nb
79
80         if sum(abs(c{i,j}(:)~=0))
81             newval=(c{i,j}(:)'.')
82             val = [val newval];
83             brow_idx=[brow_idx i]
84             d=[d j]
85             count=count+1
86         end
87     end
88 end
89 n=diff(d)
90 answer=find(n)+1
91 bcol_ptr=[1 answer count+1]
92
93 end

```