

Επιστημονικός Υπολογισμός  
Εργαστηριακή Άσκηση 2019-20

Παναγιώτης Καββαδίας

ΑΜ:1054350

Περιεχόμενα

1.Εισαγωγικά.....	2
2.Τανυστές και μπλοκ τριδιαγώνια μητρώα.....	3
3.Πράξεις με μητρώα ειδικής μορφής, πολυώνυμα μητρώων.....	5
4.Εφαρμογές σε δίκτυα.....	6

## 1.Εισαγωγικά

Για την καταγραφή των στοιχείων του συστήματος χρησιμοποιήθηκε το CPU-Z

Έναρξη/λήξη εργασίας	22/12/2019-18/1/2020
Model	HP Pavilion 15
O/S	Windows 10 Education
Processor name	4-Core Intel Core i7(i7-6700HQ)
Processor Speed	2.6GHz(base)
Number of processors	1
Total # cores	4
Total # threads	8
FMA instruction	FMA3
L1 cache	128KB Instruction,128KB Data
L2 cache	256KB per core
L3 cache	6MB shared
Gflops/s	294.6
Memory	8GB
Memory Bandwidth	12.8 GB/s
MATLAB version	9.7.0.1190202(R2019b) update 2

Με την εκτέλεση της bench παράγεται το παρακάτω:

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
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
<b>This machine</b>	<b>0.2017</b>	<b>0.1793</b>	<b>0.0247</b>	<b>0.1444</b>	<b>0.8924</b>	<b>0.9456</b>
MacBook Pro, macOS 10.14.1, Intel Core i5 @ 2.6 GHz	0.2734	0.1980	0.0177	0.1423	2.0198	1.3889

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.

Στο πρώτο μέρος κατασκευάστηκαν οι συναρτήσεις `mask_band` για την κατασκευή μητρώων μάσκα, `dd_check` για τον έλεγχο διαγώνιας κυριαρχίας και η συνάρτηση `matrix2latex2` για την μετατροπή ενός μητρώου σε μορφή επιθυμητή για το LaTeX. Για την κατασκευή της βασίστηκε στην συνάρτηση `matrix2latex` του M.Koehler. Η συνάρτηση εκτός από την μορφή `tabular` που παράγει η συνάρτηση του

Koehler θα πρέπει μέσω της εισαγωγής του κατάλληλου ορίσματος `mstyle` να μπορεί να επιστρέψει δομές `pmatrix`, `bmatrix` και `vmatrix`

Στην υλοποίηση μου αν δεν δοθεί όρισμα `mstyle` η έξοδος είναι σε μορφή `tabular`. Ο κώδικας είναι σε μεγάλο βαθμό ίδιος με τον κώδικα του Koehler, έτσι όλα τα δυνατά ορίσματα της συνάρτησης `matrix2latex` είναι δεκτά και λειτουργικά και στην `matrix2latex2` που κατασκεύασα.

Στο `cell array` `okargs` προστέθηκε το `argument mstyle` ώστε να το δέχεται ως όρισμα στην μορφή που απαιτεί η εκφώνηση.

Έχω ορίσει μια `global` μεταβλητή, την `latex_envir` την οποία έχω αρχικοποιήσει ως `tabular`. Στο `switch statement` έχω ορίσει άλλη μια `case` επεκτείνοντας το `switch statement` του Koehler για την περίπτωση που το όρισμα είναι το έκτο του `cell array okargs`, δηλαδή το `mstyle`. Σε αυτή την περίπτωση η μεταβλητή `latex_envir` παίρνει την τιμή της `rval`, δηλαδή την τιμή που ακολουθεί αμέσως μετά το `argument`. Στην συγκεκριμένη περίπτωση δηλαδή η `rval` θα έχει ένα από τα αλφαριθμητικά `tabular`, `pmatrix`, `vmatrix`, `bmatrix`. Μετά από έρευνα των προαναφερόμενων δομών αναπαράστασης στο `latex` παρατήρησα ότι το `format` γραφής στο `latex` για τις δομές `pmatrix`, `vmatrix`, `bmatrix` είναι ίδιο με την μόνη αλλαγή ότι στο `begin` και `end` αλλάζει το όνομα της δομής.

Για παράδειγμα `begin{pmatrix}`. Οπότε μέσα στις αγκύλες μπαίνει η τιμή της μεταβλητής `latex_envir`.

Όμως στην δομή `tabular` δεν αρκεί αυτό καθώς χρειάζονται και `hlines` για τις στήλες. Έτσι ότι χρειάζεται αποκλειστικά για την δομή `tabular` έχει μπει σε `if statements` που εκτελούνται μόνο αν η μεταβλητή `latex_envir` έχει την τιμή `tabular`.

## 2. Τανυστές και μπλοκ τριδιαγώνια μητρώα

Ένα μπλοκ τριδιαγώνιο μητρώο μπορεί να αποθηκευτεί ως 4-mode  $m \times m \times n \times n$  tensor που απαιτεί χώρο αποθήκευσης συγκρίσιμο με αυτόν που απαιτεί το μητρώο αν δούμε τον tensor σαν ένα μητρώο με  $n^2$  μπλοκ διάστασης  $m \times m$ .

Για τις ανάγκες του ερωτήματος κατασκευάστηκε συνάρτηση `b2t` που δοθέντος μητρώου τέτοιου τύπου το μετατρέπει σε tensor. Αφού έχουμε μητρώο διαστάσεων  $(m*n) \times (m*n)$  δημιουργώ αρχικά τανυστή διαστάσεων  $m \times n \times m \times n$  και με το `permute` αλλάζω τις διαστάσεις ώστε να είναι  $m \times m \times n \times n$  όπως ζητείται.

Η συνάρτηση αυτή εφαρμόστηκε στο ζητούμενο μητρώο  $P_{(4,3)}$  και δημιουργήθηκε tensor διαστάσεων  $4 \times 4 \times 3 \times 3$  όπως φαίνεται παρακάτω. Η πρώτη `mode-1` ίνα είναι η 54,-16,1,0 και βρέθηκε με την εντολή `T(:,1,1,1)` ενώ η `mode-4` ίνα είναι η 54,-12,0 και βρέθηκε με την εκτέλεση της εντολής `T(1,1,1,:)` όπου `T` είναι ο τανυστής που προέκυψε από την συνάρτηση `b2t`.

```
val is a tensor of size 4 x 4 x 3 x 3
val(:,:,1,1) =
    54    -12     0   -16
   -16     0     0    54
     1     0     0   -16
     0     0     0     1
val(:,:,2,1) =
     0     0     1     0
   -12     0   -16     0
     0     0    54   -12
     0     0   -16     0
val(:,:,3,1) =
     0     0     0     0
     0     1     0     0
     0   -16     0     0
     0    54   -12     0
val(:,:,1,2) =
   -12    54   -12     0
     0   -16     0   -12
     0     1     0     0
     0     0     0     0
```

```

val(:, :, 2, 2) =
    -16     0     0     1
     54    -12     0    -16
    -16     0    -12    54
     1     0     0    -16
val(:, :, 3, 2) =
     0     0     0     0
     0     0     1     0
    -12     0    -16     0
     0    -12    54    -12
val(:, :, 1, 3) =
     0    -12    54     0
     0     0   -16     0
     0     0     1     0
     0     0     0     0
val(:, :, 2, 3) =
     0   -16     0     0
    -12    54     0     0
     0   -16     0   -12
     0     1     0     0
val(:, :, 3, 3) =
     1     0     0     0
    -16     0     0     1
     54     0     0   -16
    -16     0   -12    54

```

### 3.Πράξεις με μητρώα ειδικής μορφής, πολυώνυμα μητρώων και χρήση διεπαφής mex

Ο ζητούμενος πίνακας παρατίθεται παρακάτω:

matrix	N	ΔΚ	Συμμετρικό	Ζώνης	Αντιστρέψιμο	δ.κ κl(A)
C <sub>1000</sub>	1000	OXI	NAI	OXI	NAI	41.2165
P <sub>(100,10)</sub>	100x10	OXI	NAI	(100,100)	OXI	139.9998
P <sub>(10,100)</sub>	10x100	OXI	NAI	(10,10)	OXI	135.6683
P <sub>(100,100)</sub>	100x100	OXI	NAI	(100,100)	OXI	7001.8122
bcsstm07	420x420	OXI	NAI	(47,47)	OXI	11220.5540
email	1133x1133	OXI	NAI	(605,605)	NAI	infinite

Για να κατασκευαστεί ο παραπάνω πίνακας πρώτα δημιουργήθηκαν τα μητρώα.

Το πρώτο δημιουργήθηκε με δύο for loops και συγκεκριμένα με αυτές τις εντολές:

```
C1000: for i=1:1000
A(i,i)=1+(i^(1/2));
end
for i=1:1000
for j=1:1000
if j~=i
A(i,j)=1/((abs(i-j))^2);
end
end
end
```

Το δεύτερο με την εντολή `kron(eye(10),toeplitz([30 -16 1 zeros(1,100-3)]))+kron(toeplitz([24 -12 zeros(1,10-2)]),eye(100));`

Το τρίτο με την εντολή `kron(eye(100),toeplitz([30 -16 1 zeros(1,10-3)]))+kron(toeplitz([24 -12 zeros(1,100-2)]),eye(10));`

Το τέταρτο με την εντολή `kron(eye(100),toeplitz([30 -16 1 zeros(1,100-3)]))+kron(toeplitz([24 -12 zeros(1,100-2)]),eye(100));`

Για τα μητρώα `bscstm07` και `email` πρώτα έγιναν `load` τα `.mat` αρχεία και μετά ανατέθηκαν τα `matrices` σε μεταβλητές της `matlab`.

Δηλαδή `load('bscstm07')` και μετά `bscstm07 = Problem.A`

Ομοίως και με το μητρώο `email`. Αυτό συμβαίνει καθώς τα μητρώα της σουίτας `SuitSparse` δίνονται σε `.mat` αρχεία τα οποία περιέχουν ένα Matlab object με όνομα `Problem` και ο ζητούμενος πίνακας βρίσκεται στην μεταβλητή `A` του object.

Για τον έλεγχο διαγώνιας κυριαρχίας χρησιμοποιήθηκε η συνάρτηση `dd_check` που κατασκευάστηκε στο πρώτο ερώτημα. Για τον έλεγχο συμμετρικότητας χρησιμοποίησα την συνάρτηση `issymmetric` της Matlab που επιστρέφει 1 αν το μητρώο είναι συμμετρικό και 0 αν δεν είναι.

Για την ζώνη του μητρώου χρησιμοποιήθηκε η συνάρτηση `bandwidth` της Matlab ενώ για τον έλεγχο αντιστρεψιμότητας η συνάρτηση `inv`. Η συνάρτηση `inv` επιστρέφει τον αντίστροφο του εισαχθέντος μητρώου ή μήνυμα σφάλματος αν το μητρώο δεν είναι αντιστρέψιμο.

Για την εύρεση του δείκτη κατάστασης ως προς τη νόρμα 1 χρησιμοποιήθηκε η συνάρτηση `condest` της Matlab όπως προτείνει η εκφώνηση.

Επίσης δημιουργήθηκε η συνάρτηση `polyvalm_MV` η οποία δοθέντος τετραγωνικού μητρώου  $A$ , διανύσματος  $c$  και συντελεστών  $p$  της δυναμομορφής επιστρέφει το  $p(A)c$  κάνοντας μόνο BLAS-2 πράξεις.

## 4.Εφαρμογές σε δίκτυα

Στο τέταρτο μέρος έπρεπε να δημιουργηθεί συνάρτηση που υπολογίζει τους δείκτες Katz για ένα δεδομένο μητρώο  $A$ . Στις μετρήσεις που παρατίθενται χρησιμοποιήθηκε το μητρώο `email`.

Σημαντικό ζήτημα ήταν να βρω την μέγιστη τιμή του  $\alpha$  για την οποία ο υπολογισμός έχει νόημα. Για να έχει νόημα πρέπει το  $\alpha$  να είναι μικρότερο του κλάσματος  $1/\lambda_{\max}$  όπου  $\lambda_{\max}$  η μέγιστη ιδιοτιμή του μητρώου  $A$ , δηλαδή στην περίπτωση μας η μέγιστη ιδιοτιμή του μητρώου `email`. Οι ιδιοτιμές

βρέθηκαν με την εντολή `eigs` και υπολογίστηκε ότι πρέπει  $a_{\max} < 0.048$ . Οπότε για  $a$  που παίρνει τιμές στο διάστημα  $[0.01:0.01:1]$  οι τιμές που θα πάρει θα είναι 0.01, 0.02, 0.03 και 0.04.

Δημιουργήθηκε συνάρτηση `multiKatz` η οποία δοθέντος του μητρώου γειτνίασης και διανύσματος  $\alpha$  μήκους  $s \geq 1$  η κλήση της επιστρέφει τις τιμές των δεικτών Katz για το αντίστοιχο  $\alpha = \alpha(i)$ .

Για την εκτέλεση όλων των υπολογισμών απαιτούμενων για την συμπλήρωση του παρακάτω πίνακα καθώς και την παρουσίαση των αποτελεσμάτων σε γραφικές παραστάσεις δημιουργήθηκε script με την ονομασία `Script_katz` (εσωκλείεται στα παραδοτέα) το οποίο καλεί την `multiKatz` με τις κατάλληλες παραμέτρους.

I-alpha(1)*email	a	runtime (sec)	επαναλήψεις	κατάλοιπα	top-5 (hi to lo)
explicit	0.01	0.0347	0	0	(105,333,42,16,23)
	0.02	0.0347	0	0	(105,16,42,333,23)
	0.03	0.0327	0	0	(105,16,42,333,196)
	0.04	0.0339	0	0	(105,16,42,196,333)
serial+PCG	0.01	0.0279	7	3.2280e-07	(105,333,42,16,23)
	0.02	0.0246	9	1.5029e-06	(105,16,42,333,23)
	0.03	0.0275	12	1.2408e-06	(105,16,42,333,196)
	0.04	0.0406	16	9.2830e-07	(105,16,42,196,333)
serial+PCG+prec(ichol)	0.01	0.0263	3	2.8257e-06	(105,333,42,16,23)
	0.02	0.0210	5	5.813e-08	(105,16,42,333,23)
	0.03	0.0232	6	2.8743e-07	(105,16,42,333,196)
	0.04	0.0259	8	3.3260e-07	(105,16,42,196,333)

Ακολουθούν οι ζητούμενες γραφικές παραστάσεις.







