Επιστημονικός Υπολογισμός Ι

3η Εργαστηριακή Άσκηση

Παπαρροδοπούλου Αναστασία ΑΜ 3873

01/23/14

Εισαγωγή – Χαρακτηριστικά υπολογιστικού συστήματος:

Για τον υπολογισμό των παρακάτω χαρακτηριστικών του υπολογιστικού συστήματος στο οποίο υλοποιήθηκε η εργαστηριακή άσκηση, χρησιμοποιήθηκαν δύο ειδικά προγράμματα(τα οποία κατεβάστηκαν από τη διεύθυνση http://www.cpuid.com/), το CPU-Z και το PC Wizard.

Τύπος και συχνότητα λειτουργίας επεξεργαστή :
Intel Mobile Core 2Duo P8600 @ 2.40GHz

Μέγεθος και αριθμός πιπέδων κρυφής μνήμης :
L1 D-Cache : Size 32 Kbytes x2, Descriptor 8-way set associative, 64-byte line size
L1 I-Cache: Size 32 Kbytes x2, Descriptor 8-way set associative, 64-byte line size
L2 Cache : Size 3072 Kbytes, Descriptor 12-way set associative, 64-byte line size

Το **είδος της πολιτικής εγγραφής** στην κρυφή μνήμη είναι Write-Back.

Το λειτουργικό σύστημα του μηχανήματος είναι Windows 7 Professional Service Pack 1 (64-bit).

Επίσης η **έκδοση Matlab** που χρησιμοποιήθηκε είναι η R2012b(64-bit).

Ερώτημα 1 – Κατασκευή Μητρώων με Ορισμένο Δείκτη Κατάστασης:

----

(1) Στο πρώτο αυτό υποερώτημα, μας ζητάται να κατασκευάσουμε μία συνάρτηση που θα ονομάζεται condrand_AM(condrand_3873 στη δική μου περίπτωση), η οποία θα παράγει μητρώα με δεδομένο δείκτη κατάστασης ως προς τη νόρμα-2, για τον έλεγχο αλγορίθμων. Σύμφωνα και με τα υπόλοιπα δεδομένα της εκφώνησης λοιπόν, ο κώδικας που συντάχθηκε είναι ο ακόλουθος:

```
function [A] = condrand_3873(n,kappa)

% Creating two random matrices.
M1 = rand(n);
M2 = rand(n);
% Calculating QR for matrices M1 and M2.
[Q1,R1] = qr(M1);
[Q2,R2] = qr(M2);
% Creating the diagonal of matrix D using a vector T.
T = zeros(n,1);
for i=1:n
T(i) = kappa^(-(i-1)/(n-1));
end
D = diag(T);
% Calculating final matrix A.
A = Q1*D*Q2;
% Printing the given condition number and the generated one of matrix A.
fprintf('Given condition number: %d\n',kappa);
g_c_n = cond(A);
fprintf('Generated condition number: %d\n',g_c_n);

end
```

Σαν επιβεβαίωση σωστής λειτουργίας του κώδικα, απλά βάλαμε 2 τιμές στα n και kappa, 8 και 2 αντίστοιχα, ώστε να αποτυπώσουμε το παραγόμενο μητρώο και να δούμε την ταύτιση των δύο δεικτών κατάστασης. Ακολουθεί σχετική εικόνα:

```
Command Window
>> A = condrand_3873(8,2)
Given condition number: 2
Generated condition number: 2.000000e+00

A =

    0.0981    0.3126   -0.4693   -0.3173    0.1218   -0.1399    0.2294    0.2263
    0.3618   -0.0381    0.0938    0.0528   -0.1053   -0.2861    0.3046    0.4959
    0.1690   -0.2584   -0.2433   -0.0689   -0.0840   -0.1510   -0.5673    0.2143
    0.4305   -0.1951   -0.2153    0.0298    0.4422   -0.0272    0.2501   -0.1506
   -0.2308   -0.1921   -0.3452    0.2870    0.2027   -0.0555    0.2101    0.2798
   -0.2032   -0.3280    0.1844   -0.4184    0.1995    0.1626    0.1567    0.2215
   -0.0196   -0.5654   -0.2873   -0.0801   -0.3637    0.0830    0.2120   -0.0837
    0.5014    0.0697   -0.0217    0.2440    0.0453    0.5005   -0.0667    0.2095

fx >>
```

Παρατηρούμε πως στον παραγόμενο δείκτη κατάστασης υπάρχει ένα σφάλμα, το οποίο όμως είναι αποδεκτό της περίπτωση αυτή, μιας και η απαίτηση ο δ.κ. του διαγώνιου μητρώου D να είναι "ακριβώς" kappa, είναι στα όρια σφάλματος κινητής υποδιαστολής.

(2) Στη συνέχεια μας ζητάται ο σχεδιασμός ενός προγράμματος st_condrand_3873 που θα παράγει τριδιαγώνια συμμετρικά μητρώα, που θα έχουν δείκτη κατάστασης ως προς τη νόρμα-2 ακριβώς kappa.

Σύμφωνα τώρα με την υπόδειξη της εκφώνησης, δηλαδή με το ότι πρέπει να χρησιμοποιήσουμε την προηγούμενη συνάρτηση που κατασκευάσαμε(ίσως και να την τροποποιήσουμε πρώτα), και μετά τη συνάρτηση Matlab hess, καταλήγουμε σε κάποια συμπεράσματα.

Από το help της Matlab όσον αφορά της συνάρτηση hess, βλέπουμε αρχικά ότι θέλουμε τη Hessenberg μορφή ενός μητρώου A, η οποία έχει μηδενικά κάτω από την πρώτη υποδιαγώνιο, καθώς και τις ίδιες ιδιοτιμές με το αρχικό μητρώο A. Επίσης αν το μητρώο είναι συμμετρικό(ή ερμιτιανό), τότε η μορφή αυτή δίνει ένα τριδιαγώνιο μητρώο.
Αυτή η τελευταία λειτουργία της συνάρτησης hess μας βοηθάει σε αυτό που θέλουμε να επιτύχουμε(την παραγωγή τριδιαγώνιων συμμετρικών μητρώων), μόνο όμως αν το μητρώο μας είναι ήδη συμμετρικό, κάτι το οποίο δε μας εξασφαλίζει η συνάρτηση του προηγούμενου ερωτήματος. Παρ'όλα αυτά, με τη χρήση πρώτα της condrand_3873 που υλοποιεί την παραγοντοποίηση QR, κρατάμε το παραγόμενο unitary μητρώο Q και το αντρίστροφό του Q', και πολλαπλασιάζοντάς τα με το D από δεξιά και αριστερά(Q*D*Q'), παίρνουμε ένα συμμετρικό μητρώο. Αυτό είναι κάτι το οποίο γνωρίζουμε από τη θεωρία μας της αριθμητικής ανάλυσης. Οπότε σε αυτό το σημείο η χρήση της hess, που θα πάρει σαν είσοδο ένα συμμετρικό πλέον μητρώο, θα μας παράξει ένα τριδιαγώνιο συμμετρικό μητρώο, όπως μας ζητάει η εκφώνηση.

Οπότε επιλέγουμε αρχικά να τροποποιήσουμε την condrand_3873, ώστε να παράγεται απ'ευθείας συμμετρικό μητρώο. Έτσι έχουμε πλέον τον κώδικα:

```
function [A] = sym_condrand_3873(n,kappa)

% Creating a random matrix.
M1 = rand(n);
% Calculating QR for matrix M1.
[Q1,R1] = qr(M1);
% Creating the diagonal of matrix D using a vector T.
T = zeros(n,1);
for i=1:n
    T(i) = kappa^(-(i-1)/(n-1));
end
D = diag(T);
% Calculating final symmetric matrix A.
A = Q1*D*Q1';
% Printing the given condition number and the generated one of matrix A.
fprintf('Given condition number: %d\n',kappa);
g_c_n = cond(A);
fprintf('Generated condition number: %d\n',g_c_n);

end
```

Και στη συνέχεια, ο κώδικας λοιπόν της καινούργιας μας συνάρτησης st_condrand_3873 για την υλοποίηση που έχουμε περιγράψει παραπάνω, με τη χρήση της sym_condrand_3873, είναι ο ακόλουθος:

```
function [W] = st_condrand_3873(n,kappa)

% Calling condrand_3873 to create an appropriate matrix.
[A] = sym_condrand_3873(n,kappa);
% Calculating tridiagonal Hessenberg form of matrix A.
W = hess(A);
% Printing the final condition number.
c_n_w = cond(W);
fprintf('Final condition number: %d\n',c_n_w);

end
```

Όπως και πριν, σαν επιβεβαίωση της σωστής λειτουργίας του κώδικα, απλά βάλαμε 2 τιμές στα n και kappa, 8 και 2 αντίστοιχα. Και πήραμε την εξής εκτύπωση:

```
Command Window
>> B = st_condrand_3873(8,2)
Given condition number: 2
Generated condition number: 2.000000e+00
Final condition number: 2.000000e+00

B =

    0.8319   -0.1370    0.0000   -0.0000    0.0000    0.0000   -0.0000   -0.0000
   -0.1370    0.6264    0.0958   -0.0000   -0.0000   -0.0000   -0.0000    0.0000
         0    0.0958    0.7151    0.1222    0.0000   -0.0000   -0.0000   -0.0000
         0         0    0.1222    0.8412    0.1682    0.0000   -0.0000    0.0000
         0         0         0    0.1682    0.7038   -0.0606   -0.0000   -0.0000
         0         0         0         0   -0.0606    0.6350   -0.0394   -0.0000
         0         0         0         0         0   -0.0394    0.6299   -0.0034
         0         0         0         0         0         0   -0.0034    0.8203

fx >>
```

Παρατηρούμε αρχικά ότι οι δείκτες κατάστασής μας ταυτίζονται όπως θέλουμε, αλλά ότι εκεί που θα περιμέναμε στο πάνω τριγωνικό κομμάτι του μητρώου(πέρα από την 1^{η} υπερ-διαγώνιο), να έχουμε μηδενικά, υπάρχει ένα μικρό σφάλμα λόγω αριθμών κινητής υποδιαστολής. Οπότε ο τρόπος να έχουμε παραγωγή αυστηρών τριδιαγώνιων συμμετρικών μητρώων, είναι να μηδενίζουμε manually αυτές τις τιμές, οπότε και θα προσθέσουμε ένα μικρό κομμάτι κώδικα στην st_condrand_3873 που να το κάνει αυτό, και θα έχουμε έτσι το τελικό πρόγραμμα:

```
function [W] = st_condrand_3873(n,kappa)

% Calling condrand_3873 to create an appropriate matrix.
[A] = sym_condrand_3873(n,kappa);
% Calculating tridiagonal Hessenberg form of matrix A.
F = hess(A);
% Creating a zero matrix W and saving the values of the 3 main diagonal of
matrix F.
W = zeros(n);
W = diag(diag(F)) + diag(diag(F,-1),-1) + diag(diag(F,1),1);
% Printing the final condition number.
c_n_w = cond(W);
fprintf('Final condition number: %d\n',c_n_w);

end
```

Με αυτή την προσθήκη έχουμε το τελικό μας τριδιαγώνιο συμμετρικό μητρώο, όπως φαίνεται στη συνέχεια:

```
Command Window
>> B = st_condrand_3873(8,2)
Given condition number: 2
Generated condition number: 2.000000e+00
Final condition number: 2

B =

    0.7398   -0.1411         0         0         0         0         0         0
   -0.1411    0.7851    0.1644         0         0         0         0         0
         0    0.1644    0.7671    0.1041         0         0         0         0
         0         0    0.1041    0.6605    0.1135         0         0         0
         0         0         0    0.1135    0.7197   -0.0806         0         0
         0         0         0         0   -0.0806    0.7053   -0.1521         0
         0         0         0         0         0   -0.1521    0.7532   -0.0026
         0         0         0         0         0         0   -0.0026    0.6729

>>
```

(Όπως και σε προηγούμενες εικόνες, βλέπουμε το μικρό αυτό σφάλμα που παρατηρήθηκε στις τιμές των μητρώων, να υπάρχει και στην τιμή του δείκτη κατάστασης(2.000000e+00), που μιας και είναι ελάχιστο, δε μας επηρρεάζει στους υπολογισμούς μας.)