Άσκηση 3

Χρήστος Αλέξανδρος Τσιγγιρόπουλος

29 December 2021

1 Άσκηση 3.1

Το πρόγραμμα ζητάει απο τον χρήστη το μέγεθος του πίνακα A, και περιμένει μετά σαν είσοδο τον επαυξήμενο πίνακα. Δλδ αν εισαχθεί στην αρχή το 3 θα περιμένει 4 στοιχεία (3 του πίνακα A και 1 του πίνακα B) +(αλλαγή γραμμής), 4 στοιχεία +(αλλαγή γραμμής) και άλλα 4 στοιχεία +(αλλαγή γραμμής). Άν όλα πάνε καλά το πρόγραμμα καλεί την συνάρτηση pa_lu(A,b) που επιστρέφει 5 συναρτήσεις (P,A,L,U,Z) όπου PA=LU , Ux=y με Ax=b και Ly=Pb=z . Στην συνέχεια καλώ την συνάρτηση ly_z(L,Z) που επιστρέφει τον πίνακα y. Τέλος, καλείται η συνάρτηση ux_y(U, y) που επιστρέφει τον πίνακα x δλδ τον πίνακα με τις ρίζες που τις τυπώνει.

Ειδικότερα για τις 3 συναρτήσεις:

- pa_lu(A,b)

  Αρχικά αρχικοποιούμε τους πίνακες p,u,l στον p (0 παντού και 1 στην διαγώνιο), u=A ,στον l (0 παντού). Στην συνέχεια κάνω οδήγηση στους πίνακες u,b,l,p με βάση τον u για το [k,k] στοιχείο του u. Έπειτα βάζω στον l τον σωστό συντελεστή και στον u κάνω gauss για να δημιουργήσω τον άνω τριγωνικό της τριγωνιοποίησης. Τέλος βάζω 1 στην κύρια διαγώνιο του πίνακα l για να πάρω τον τελικό πίνακα. Στο τέλος ο πίνακας Z είναι ίσος με τον αλλαγμένο b. Επιστρέφω τους 5 πίνακες (p,a,l,u,b) με a τον πίνακα που μπήκε σαν όρισμα.

- ly_z(L,Z)

  Η συνάρτηση αυτή λύνει το Ly=Z ως προς y και επιστρέφει τον πίνακα y. Ειδικότερα, αρχικοποιεί τον y με 0 και για κάθε i τοποθετεί σε αυτό την λύση. Η λύση για κάθε i είναι y[i] = (z[i]-sum)/L[i][i] με

$$sum = \sum_{j=1}^{i-1} L[i][j] \cdot y[j] + \sum_{j=i+1}^{n} L[i][j] \cdot y[j]$$

- ux_y(U,y)

  Η συνάρτηση αυτή λύνει το Ux=y ώς προς x δλδ επιστρέφει τον πίνακα x, που περιέχει τις ρίζες του αρχικού προβλήματος Ax=b. Κάνει το ίδιο με την πάνω συνάρτηση με την διαφορά ότι το i ξεκινάει απο το τέλος προς στην αρχή γιατί ο πίνακας U είναι άνω τριγωνικός. Διαφορετικά δεν θα μπορούσε να βγεί το αποτέλεσμα. Με αυτόν τον τρόπο η πρώτη ρίζα που βρίσκουμε είναι η xn μετά η xn-1 και στο τέλος την x1. Η λύση για κάθε i σε αυτή την περίπτωση είναι x[i] = (y[i]-sum)/U[i][i] με

$$sum = \sum_{j=n}^{i+1} U[i][j] \cdot x[j] + \sum_{j=i-1}^{0} U[i][j] \cdot x[j]$$

Παράδειγμα: για είσοδο: 3

2 1 5 5

4 4 -4 0

1 3 1 6

Το πρόγραμμα τυπώνει :

x1 = -1.000 x2 = 2.000 x3 = 1.000

2 Άσκηση 3.2

Το πρόγραμμα έχει ορισμένο ένα πίνακα A και τυπώνει τον κάτω τριγωνικό πίνακα απο την αποσύνθεση Cholesky. Ειδικότερα, αρχικοποιεί ένα πίνακα L (nxn) με 0 με n = μέγεθος(A). Στην συνέχεια αποθηκεύεται :

$$L[k][i] = \frac{A[k][i] - \sum_{j=1}^{i-1} L[i][j] \cdot L[k][j]}{L[i][i]}, k \neq i$$

$$L[k][k] = \frac{A[k][k] - \sum_{j=1}^{k-1} L^2[k][j]}{L[i][i]}, k = i$$

και στο τέλος τυπώνει τον πίνακα L.

3 Άσκηση 3.3

Στό πρόγραμμα αυτό, αρχικοποιούμε με τις συναρτήσεις initialize_a(n) και initialize_b(n) τέσσερις πίνακες, τους a10,b10 με n=10 και a10000,b10000 με n=10000. Στην συνέχεια, καλούμε την συνάρτηση gauss_seidel(a,b), αρχικά για τους πίνακες a10,b10. Η συνάρτηση αυτή επιστρέφει έναν πίνακα x που περιέχει τις προσεγγίσεις των ριζών.

Πιο συγκεκριμένα η συνάρτηση αυτή , αρχικοποιεί δύο νέους πίνακες x0,xn (nxn) με 0 . Έπειτα για κάθε διαφορετικό i τοποθετεί στο xn[i] :

$$xn[i] = \frac{b[i] - \sum_{j=1}^{i-1} a[i][j] \cdot \mathbf{x0[i]} - \sum_{j=i+1}^{n} a[i][j] \cdot \mathbf{xn[i]}}{a[i][i]} \tag{1}$$

και αποθηκεύουμε τον νέο πίνακα xn στον πίνακα x0.

Αφού γίνουν αυτά καλούμε την συνάρτηση find(a,b,xn) που επιστρέφει True/False δλδ χρησιμοποιείτε για τον έλεγχο τερματισμού. Σε αυτή, ελέγχουμε αν η άπειρη νόρμα στο σφάλμα της λύσης, είναι μικρότερη απο το 0.5e-4 δλδ :

$$\|A\|_\infty \leq 0.5 \cdot 10^{-4}$$

Αν είναι μεγαλύτερη τότε γυρνάει True και επαναλαμβάνουμε από την (1). Αν είναι μικρότερη τότε γυρνάει False και τελειώνει η συνάρτηση, τυπώνοντας τις επαναλήψεις που χρειάστηκαν και επιστρέφοντας την λίστα με τις προσεγγίσεις.

Για τους πίνακες a10,b10 εμφανίζονται και οι προσεγγίσεις, ενώ για τους πίνακες a10000,b10000 εμφανίζονται μόνο οι επαναλήψεις.