Εργασία 1
Επιστημονικοί Υπολογισμοί
Ελένη Λιάρου
1115201900100

Για κάθε μία από τις 4 περιπτώσεις που είχαμε να επιλύσουμε υπάρχει ξεχωριστό αρχείο κώδικα(γραμμένο σε C). Πάνω από κάθε πίνακα αναφέρω ποιο αρχείο πρέπει να εκτελεστεί για να δούμε τα αποτελέσματα.

Για να τρέξουμε ένα πρόγραμμα αρκεί να το μεταγλωττίσουμε (gcc -o <onoma ektelesimou> <onoma_arxeio>.c) και μετά να γράψουμε στην γραμμή εντολών ./<onoma_ektelesimou> k, το k μπορεί να είναι κ=1,2,3,4,5 για κάθε τιμή του k παρακάτω εξηγώ ποιο παράδειγμα τρέχει ο αλγόριθμος.

Σε όλες τις περιπτώσεις αν δώσει ο χρήστης κ=1 τότε θα του ζητηθεί να εισάγει ένα-ένα τα στοιχεία του πίνακα, αν δώσει κ=2 τότε θα δημιουργηθεί τυχαίος πίνακας με στοιχεία από το 0 έως το 100 και το διάνυσμα x είναι το x=(1,1,..,1)^T, ενώ αν δώσει κ=3 θα πρέπει να δώσει ως εντολή ./<onoma_ektelesimou> k m, όπου κ=3 και m το όνομα του αρχείου που υπάρχουν τα στοιχεία του πίνακα. Στην πρώτη γραμμή αυτού του αρχείου θα πρέπει να βρίσκεται η διάσταση n του πίνακα, και μετά σε κάθε γραμμή και από ένα στοιχείο του πίνακα A, μετά είτε τα στοιχεία του πίνακα b και της λύσης x είτε τα στοιχεία του αντίστροφου του A.

Σε όλα τα αρχεία υπάρχει σχολιασμός για την λειτουργικότητα και χρησιμότητα των συναρτήσεων που υλοποίησα. Έχω βασιστεί στους αλγορίθμους που αναλύσαμε στο μάθημα. Επιπλέον συμπεριλαμβάνω αρχεία .txt που υλοποιούν κάποιες από τις ενδεικτικές εφαρμογές που αναφέρονται στην εκφώνηση.
num.txt: ενδεικτική εφαρμογή 1.a.1
num1.txt: ενδεικτική εφαρμογή 1.b.1
num2.txt: ενδεικτική εφαρμογή 2.α.1
num3.txt: ενδεικτική εφαρμογή 2.a.2

Τον αριθμό συνθήκης του πίνακα A τον υπολογίζω μόνο όταν χρησιμοποιώ κάποια από τις 2 μεθόδους για να βρω τον αντίστροφο αφού για την επίλυση του συστήματος Ax=b δεν βρίσκω τον A^-1(σε σχετικό email μου είχατε πει πως δεν υπάρχει πρόβλημα).

Στα αρχεία που εφαρμόζουν την μέθοδο Cholesky υπάρχει η επιλογή για δημιουργία τυχαίου πίνακα A παρόλα αυτά είναι πολύ πιθανό να μην εμφανίσει σωστά αποτελέσματα αφού με την rand δεν μπορούμε να εξασφαλίσουμε την συμμετρία ενός πίνακα. Δυστυχώς δεν υπάρχει η επιλογή που αναφέρετε στην εφαρμογή 2.a.7 για την MATLAB και για την γλώσσα C.

Η επιλογή k=3(δημιουργία πίνακα με rand) ενώ υπάρχει για τα αρχεία που βρίσκουν τον αντίστροφο ενός πίνακα δεν λειτουργεί σωστά, διότι αν φτιάξουμε έναν πίνακα με rand δεν μπορούμε να γνωρίζουμε από πριν τον πραγματικό αντίστροφο για να βρούμε τον αριθμό συνθήκης, τα σφάλματα κλπ.

Για την επίλυση των γραμμικών συστημάτων αποθηκεύω στον ίδιο πίνακα τα στοιχεία του A και του b, δηλαδή χρησιμοποιώ πίνακα διάσταση nx(n+1), ενώ για το x πίνακα nx1.
Για την εύρεση του αντίστροφου:

a) Με την μέθοδο LU

Βρίσκω τους πίνακες P,L και U, έτσι ώστε PA=LU, υπολογίζω τους πίνακες L^-1, U^-1, με προς τα εμπρός και προς τα πίσω αντικατάσταση με τον μοναδιαίο πίνακα αντίστοιχα. Τέλος κάνω τον πολλαπλασιασμό L^-1*U^-1 *P και έτσι βρίσκω τον A^-1.

b) Με την μέθοδο Cholesky

Βρίσκω τον πίνακα L, έτσι ώστε L*L^T = A, υπολογίζω τον L^-1, με προς τα εμπρός αντικατάσταση με τον μοναδιαίο, και έχουμε A^-1 = (L^T)^-1*L-1 = (L^-1)^T*L^-1.

1. Μέθοδος LU με οδήγηση
    - Επίλυση γραμμικού συστήματος Ax=b. (lu.c)

Αν δώσουμε k=4 ο πίνακας A θα είναι ο πίνακας Hilbert της εφαρμογής 1.a.3.
Αν δώσουμε k=5 ο πίνακας A θα είναι αυτός της εφαρμογής 1.a.4 και ο χρήστης θα μπορεί να επιλέξει για την διάσταση n ανάμεσα σε 100, 500, 1000.

Το πινακάκι έχει συμπληρωθεί με βάση την ενδεικτική εφαρμογή 1.a.4 (gcc -o lu lu.c, ./lu 5)

Διάσταση πίνακα n	Απόλυτο σχ. σφάλμα δx/x	Απόλυτο σχ. υπόλοιπο r/b	Αριθμός συνθήκης του A	Χρόνος
100	0.0(1.110223e-15)	0.0(1.091177e-15)	-	0.010826
500	0.0(2.664535e-15)	0.0(2.584850e-15)	-	0.300739
1000	0.0(3.108624e-15)	0.0(3.049765e-15)	-	1.479917

Όταν έτρεξα ./lu 4 για να δω τα αποτελέσματα του αλγορίθμου για τον πίνακα Hilbert με n=10 παρατήρησα ότι υπάρχει πολύ μεγάλη απόκλιση μεταξύ της πραγματικής λύσης και αυτής που υπολόγισε ο αλγόριθμος, το απόλυτο σχετικό σφάλμα ήταν ίσο με ||dx||/||x|| = 8311831201020.536133(8.311831e+12), ενώ το απόλυτο υπόλοιπο ήταν ίσο με ||r||/||b|| = 0.000064(6.442629e-05). Αυτό οφείλεται στο ότι ο πίνακας Hilbert είναι ill-conditioned(έχει μεγάλο αριθμό συνθήκης).

    - Υπολογισμός του αντίστροφου A^-1. (lu_inv.c)

Αν δώσουμε k=4 ο πίνακας A θα είναι ο πίνακας Hilbert της εφαρμογής 1.b.3. και ο χρήστης θα μπορεί να επιλέξει για την διάσταση n.
Αν δώσουμε k=5 ο πίνακας A θα είναι αυτός της εφαρμογής 1.b.2 και ο χρήστης θα μπορεί να επιλέξει για την διάσταση n.

Το πινακάκι έχει συμπληρωθεί με βάση την ενδεικτική εφαρμογή 1.b.2 (gcc -o lu lu_inv.c, ./lu 5)
Όταν έβαλα διάσταση n>200 ο αλγόριθμος αργούσε αρκετά να εμφανίσει το αποτέλεσμα για αυτό τα πειράματα έγιναν για n=100, 150, 200.

Διάσταση πίνακα n	Απόλυτο σχ. σφάλμα	Απόλυτο σχ. υπόλοιπο	Αριθμός συνθήκης του Α	Χρόνος
100	0.535739(5.357389e-01)	2048.366472(2.048366e+03)	3	0.431535
200	0.538169(5.381685e-01)	8224.899701(8.224900e+03)	3	8.925562
150	0.537356(5.373562e-01)	4620.341017(4.620341e+03)	3	2.067496

Παρατηρώ ότι όσο αυξάνεται το n ο χρόνος αυξάνεται, το απόλυτο σχετικό σφάλμα παρουσιάζει μία ελαφριά αύξηση ενώ το απόλυτο σχετικό υπόλοιπο παρουσιάζει αρκετά μεγάλη αύξηση. Προφανώς αφού οι πίνακες ακολουθούν τις ίδιες πράξεις για την δημιουργία των στοιχείων τους έχουν ίδιο αριθμό συνθήκης ανεξαρτήτως του n. Ο πίνακας της 1.b.2 έχει μικρό αριθμό συνθήκης για αυτό ο υπολογιζόμενος αντίστροφος είναι πολύ κοντά στον πραγματικό. Έκανα μία προσπάθεια να τρέξω και τον πίνακα Hilbert για n=10, έγινε όμως overflow.

2. Μέθοδος Cholesky

- Επίλυση γραμμικού συστήματος Ax=b (cholesky.c)

Αν δώσουμε k=4 ο πίνακας Α θα είναι ο πίνακας της εφαρμογής 2.a.6 και ο χρήστης θα μπορεί να επιλέξει για την διάσταση n ανάμεσα σε 100, 500, 1000.

Το πινακάκι συμπληρώθηκε με βάση την εφαρμογή 2.a.6(gcc -o ch cholesky.c, ./ch 4)

Διάσταση πίνακα n	Απόλυτο σχ. σφάλμα δx/x	Απόλυτο σχ. υπόλοιπο r/b	Αριθμός συνθήκης του Α	Χρόνος
100	0.0(6.661338e-16)	0.0(6.819854e-16)	-	0.010063
500	0.0(1.554312e-15)	0.0(1.573387e-15)	-	0.233398
1000	0.0(1.998401e-15)	0.0(2.033177e-15)	-	1.068932

- Υπολογισμός του αντίστροφου A^-1 (cholesky_inv.c)

Αν δώσουμε k=4 ο πίνακας Α θα είναι ο πίνακας Hilbert της εφαρμογής 1.b.3. και ο χρήστης θα μπορεί να επιλέξει για την διάσταση n.
Αν δώσουμε k=5 ο πίνακας Α θα είναι αυτός της εφαρμογής 1.b.2 και ο χρήστης θα μπορεί να επιλέξει για την διάσταση n.

Το πινακάκι έχει συμπληρωθεί με βάση την ενδεικτική εφαρμογή 1.b.2 (gcc -o ch cholesky_inv.c, ./ch 5)
Όταν έβαλα διάσταση n>200 ο αλγόριθμος αργούσε αρκετά να εμφανίσει το αποτέλεσμα για αυτό τα πειράματα έγιναν για n=100, 150, 200.

Διάσταση πίνακα n	Απόλυτο σχ. σφάλμα	Απόλυτο σχ. υπόλοιπο	Αριθμός συνθήκης του A	Χρόνος
100	0.0(1.468041e-15)	0.0(1.282461e-10)	3	0.023636
150	0.0(3.572695e-16)	0.0(2.207744e-09)	3	0.064876
200	0.0(6.263639e-16)	0.0(2.183494e-09)	3	0.115099

- Παρατήρησα ότι η μέθοδος Cholesky είναι πιο γρήγορη από την LU τόσο για την επίλυση γραμμικού συστήματος όσο και για την εύρεση του αντίστροφου.
- Χρησιμοποίησα και τις 2 μεθόδους για την εύρεση αντίστροφου του πίνακα της εφαρμογής 1.b.1, και η μέθοδος Cholesky βρήκε με μεγαλύτερη ακρίβεια τον αντίστροφο αφού και τα 2 της σφάλματα ήταν 0, ενώ της LU ήταν 0.005079 και 0.152387(απόλυτο σχετικό σφάλμα και απόλυτο σχετικό υπόλοιπο αντίστοιχα). Ο συγκεκριμένος πίνακας έχει condition number = 4488.