

Εξήγηση κώδικα για την μέθοδο LU:

Έχουμε τα παρακάτω αρχεία:

- ask1_LU_a.cpp: περιέχει τις συναρτήσεις για την επίλυση ενός γραμμικού συστήματος με την μέθοδο LU. Η συνάρτηση «LU_decomposition» κάνει την τριγωνική διαχώριση του πίνακα με LU. Η συνάρτηση «solveLinearSystem» λύνει το γραμμικό σύστημα μετά την LU παραγοντοποίηση καλώντας τις συναρτήσεις που κάνουν προς τα πίσω και προς τα μπροστά αντικατάσταση.
- ask1_LU_b.cpp: βρίσκει τον αντίστροφο πίνακα του A χρησιμοποιώντας την μέθοδο παραγοντοποίησης LU.
- error.cpp: περιέχει την υλοποίηση των συναρτήσεων για τον υπολογισμό των νορμών διανυσμάτων, για τον πολλαπλασιασμό δύο πινάκων, για τον υπολογισμό των σφαλμάτων και του condition number. Έχουμε και μία συνάρτηση που βρίσκει τον πραγματικό αντίστροφο του A (Pei Matrix).
- Process_data.cpp: περιέχει τις συναρτήσεις που διαβάζουν και επεξεργάζονται τα δεδομένα που δίνει ο χρήστης. Η συνάρτηση «ReadFromFile» διαβάζει από ένα αρχείο κειμένου το μέγεθος n του πίνακα, τον πίνακα A και τον πίνακα b. Η συνάρτηση «random_number» φτιάχνει τυχαίους αριθμούς σε διάστημα [a,b]. Η συνάρτηση «readingData» η οποία διαβάζει τα δεδομένα. Υπάρχουν τρεις τρόποι για να γίνει αυτό:
 - Μπορεί ο χρήστης να δώσει ένα προς ένα τα στοιχεία του A και του b από την γραμμή εντολών.
 - Μπορεί το πρόγραμμα να διαβάσει τους πίνακες από ένα αρχείο κειμένου.
 - Μπορεί το πρόγραμμα να ζητήσει από τον χρήστη απλά το μέγεθος του πίνακα και να φτιάξει έναν τυχαίο πίνακα ή να χρησιμοποιήσει έναν ήδη έτοιμο (Pei matrix – εφαρμογή 1-β-2).
- main.cpp: περιέχει την main του προγράμματος η οποία καλεί τις συναρτήσεις για την επίλυση ενός γραμμικού συστήματος και την εύρεση του αντιστρόφου.
- Process_data.h: περιέχει τους ορισμούς των συναρτήσεων που βρίσκονται στο αρχείο Process_data.cpp.
- Functions.h: περιέχει τους ορισμούς των συναρτήσεων που βρίσκονται στο αρχείο ask1_LU_a.cpp, στο αρχείο ask1_LU_b.cpp και στο error.cpp.

Το αρχείο κειμένου από το οποίο το πρόγραμμα μπορεί να διαβάζει τα δεδομένα πρέπει να έχει την παρακάτω μορφή. Στην πρώτη γραμμή θα έχει την τιμή του n. Για τις επόμενες n γραμμές θα είναι ο πίνακας A στη συνέχεια θα υπάρχει μία κενή γραμμή και στο τέλος ο πίνακας b. π.χ. αν $n = 3$ το αρχείο κειμένου θα πρέπει να είναι:

3

$a_{11} \ a_{12} \ a_{13}$

$a_{21} \ a_{22} \ a_{23}$

$a_{31} \ a_{32} \ a_{33}$

$b_1 \ b_2 \ b_3$

Εξήγηση κώδικα για την μέθοδο Cholesky:

Έχουμε τα παρακάτω αρχεία:

- ask1_LL_a.cpp: περιέχει τις συναρτήσεις για την επίλυση ενός γραμμικού συστήματος με την μέθοδο LL. Η συνάρτηση «choleskyDecomposition» κάνει την τριγωνική διαχώριση του πίνακα με την LL μέθοδο. Η συνάρτηση «solveUsingCholesky» λύνει το γραμμικό σύστημα μετά την LL παραγοντοποίηση με προς τα πίσω και προς τα μπροστά αντικατάσταση.
- ask1_LL_b.cpp: βρίσκει τον αντίστροφο πίνακα του A χρησιμοποιώντας την μέθοδο παραγοντοποίησης Cholesky.
- error.cpp: περιέχει την υλοποίηση των συναρτήσεων για τον υπολογισμό των νορμών διανυσμάτων, τον πολλαπλασιασμό δύο πινάκων, τον υπολογισμό των σφαλμάτων και του condition number, έχει παρόμοια υλοποίηση με το αντίστοιχο αρχείο της μεθόδου LU. Έχουμε και μία συνάρτηση που βρίσκει τον πραγματικό αντίστροφο του A (Pei matrix).
- Process_data.cpp: έχει την ίδια υλοποίηση με αυτή για την μέθοδο LU. Με την μόνη διαφορά ότι στην περίπτωση που ο χρήστης θέλει να φτιάξει έναν τυχαίο πίνακα A, φροντίζουμε αυτός να είναι συμμετρικός και θετικά ορισμένος.
- main.cpp: περιέχει την main του προγράμματος η οποία καλεί τις συναρτήσεις για την επίλυση ενός γραμμικού συστήματος, την εύρεση του αντιστρόφου, τον υπολογισμό των σφαλμάτων, του condition number και υπολογίζει το CPU time.
- Process_data.h: περιέχει τους ορισμούς των συναρτήσεων που βρίσκονται στο αρχείο Process_data.cpp.
- Functions.h: περιέχει τους ορισμούς των συναρτήσεων που βρίσκονται στο αρχείο ask1_LL_a.cpp, στο αρχείο ask1_LL_b.cpp και στο error.cpp.

Το αρχείο κειμένου από το οποίο το πρόγραμμα μπορεί να διαβάζει τα δεδομένα πρέπει να έχει την ίδια μορφή όπως και το αντίστοιχο αρχείο της μεθόδου LU που αναφέρεται παραπάνω.

Οδηγίες μεταγλώττισης προγραμμάτων:

- Για την LU μέθοδο: πάμε στον φάκελο «LU_method» κάνουμε make και δημιουργείται το εκτελέσιμο αρχείο με όνομα «lu». Εκτελούμε το lu “./lu” το πρόγραμμα ρωτάει τον χρήστη αρχικά αν θέλει να δώσει το διάνυσμα b, αν δεν το δώσει υπολογίζεται το b με δεδομένη λύση $x^T = (1, 1, \dots, 1)$. Μετά ρωτάει με ποιον τρόπο ο χρήστης θέλει να δώσει τα δεδομένα. Υπάρχουν οι εξής τρόποι:
 - i. Να δώσει ο χρήστης τα στοιχεία των πινάκων ένα προς ένα
 - ii. Να δημιουργηθούν τυχαία δεδομένα ή να χρησιμοποιηθεί ένας ήδη υπάρχον πίνακας (στην συγκεκριμένη περίπτωση ο Pei)
 - iii. Να διαβαστούν τα δεδομένα από ένα αρχείο κειμένου.

Στη συνέχεια λύνει το γραμμικό σύστημα $Ax = b$, βρίσκει τον αντίστροφο πίνακα του A, υπολογίζει το CPU time για την επίλυση του συστήματος και το CPU time για τον υπολογισμό του αντιστρόφου. Τέλος, αν ο χρήστης δεν έχει δώσει το b (και έχει διαλέξει ως A να παρθεί ένας έτοιμος πίνακας – Pei) τότε το πρόγραμμα τυπώνει το condition number, το absolute relative error για την επίλυση του συστήματος, το absolute relative residual για την επίλυση του συστήματος και τα αντίστοιχα για την εύρεση του αντιστρόφου πίνακα του A.

- Για την μέθοδο Cholesky: πάμε στον φάκελο «Cholesky_method» κάνουμε make και δημιουργείται το εκτελέσιμο αρχείο «ll». Εκτελούμε το ll “./ll” το πρόγραμμα ρωτάει αρχικά αν ο χρήστης θέλει να δώσει το b αλλιώς θα υπολογιστεί με δεδομένη λύση την $x^T = (1, 1, \dots, 1)$. Μετά το πρόγραμμα ρωτάει τον χρήστη με ποιον τρόπο θέλει να δώσει τα δεδομένα. Υπάρχουν οι εξής τρόποι:
 - i. Να δώσει ο χρήστης τα στοιχεία των πινάκων ένα προς ένα
 - ii. Να δημιουργηθούν τυχαία δεδομένα ή να χρησιμοποιηθεί ένας ήδη υπάρχον πίνακας (Pei matrix)
 - iii. Να διαβαστούν τα δεδομένα από ένα αρχείο κειμένου.

Στη συνέχεια λύνει το γραμμικό σύστημα $Ax = b$, βρίσκει τον αντίστροφο πίνακα του A, υπολογίζει το CPU time για την επίλυση του συστήματος και το CPU time για τον υπολογισμό του αντιστρόφου. Τέλος, αν ο χρήστης δεν έχει δώσει το b και έχει διαλέξει να χρησιμοποιηθεί σαν A ένας έτοιμος πίνακας – Pei matrix, τότε το πρόγραμμα τυπώνει το condition number, το absolute relative error για την επίλυση του συστήματος, το absolute relative residual για την επίλυση του συστήματος και τα αντίστοιχα για την εύρεση του αντιστρόφου πίνακα του A.

Σημείωση: και για τις δύο μεθόδους για να υπολογίζει το πρόγραμμα τον αριθμό συνθήκης και τα σφάλματα θα πρέπει ο χρήστης να ζητάει από το πρόγραμμα την περίπτωση που χρησιμοποιείται ένας έτοιμος πίνακας (στην συγκεκριμένη περίπτωση θα είναι ο πίνακας Pei – εφαρμογή 1.β.2) και να μην δίνει τον b έτσι ώστε να θεωρούμε σαν έτοιμη λύση την $x^T = (1, \dots, 1)$. Σε κάθε άλλη περίπτωση το condition number και τα σφάλματα δεν θα υπολογίζονται.

Σχολιασμός των αποτελεσμάτων για την μέθοδο LU:

1. Μέθοδος Παραγοντοποίησης LU με μερική οδήγηση

1.α Επίλυση του Γραμμικού Συστήματος $Ax = b$				
Διάσταση Πίνακα A	Απ. σχ. Σφάλμα $\frac{\ \delta x\ _\infty}{\ x\ _\infty}$	Απ. σχ. Υπόλοιπο $\frac{\ r\ _\infty}{\ b\ _\infty}$	Αριθμός Συνθήκης ¹ $cond(A)$	Χρόνος
100	9.86e-19	4.88e-19	3	0.00435s
500	6.18e-18	3.1e-18	3	0.543s
1000	1.32e-17	6.59e-18	3	8.25s

1.β Υπολογισμός του αντιστρόφου A^{-1}				
Διάσταση Πίνακα A	Απ. σχ. Σφάλμα $\ A^{-1} - \hat{A}^{-1}\ _\infty / \ A^{-1}\ _\infty$	Απ. σχ. Υπόλοιπο $\ A\hat{A}^{-1} - I\ _\infty / \ A^{-1}\ _\infty$	Αριθμός Συνθήκης	Χρόνος
100	3.95e-18	6.83e-16	3	0.454s
500	2.08e-17	1.66e-14	3	276s
1000	4.23e-17	6.87e-14	3	8.53e+03s

- Αρχικά παρατηρούμε και για την επίλυση του γραμμικού συστήματος και για τον υπολογισμό του αντιστρόφου όσο αυξάνετε το μέγεθος του πίνακα αυξάνετε και ο χρόνος. Ο υπολογισμός του αντιστρόφου χρειάζεται περισσότερο χρόνο από την επίλυση του γραμμικού συστήματος $Ax = b$, μιας και χρησιμοποιεί την επίλυση συστήματος με την μέθοδο LU η φορές.
- Το απόλυτο σχετικό σφάλμα τόσο για την επίλυση του γραμμικού συστήματος όσο και για τον υπολογισμό του αντιστρόφου μεγαλώνει όσο μεγαλώνει και το μέγεθος του A. Το γεγονός αυτό σημαίνει ότι όσο μεγαλώνει η διάσταση του πίνακα χάνεται όλο και περισσότερη πληροφορία και συμβαίνουν περισσότερα σφάλματα στις πράξεις.
- Το απόλυτο σχετικό υπόλοιπο επίσης αυξάνεται όσο αυξάνεται και το μέγεθος του πίνακα τόσο για την επίλυση του γραμμικού συστήματος όσο και για τον υπολογισμό του αντιστρόφου. Το απόλυτο σχετικό υπόλοιπο μεγαλώνει όταν αυξάνουμε τη διάσταση του πίνακα A επειδή οι πράξεις που απαιτούνται για να βρούμε το L, U και τον αντίστροφο του A γίνονται πιο περίπλοκες και έτσι αυξάνουν τον αριθμό των αριθμητικών σφαλμάτων. Αυτό σημαίνει ότι, όσο αυξάνουμε τη διάσταση του πίνακα, τα αριθμητικά σφάλματα μπορεί να επηρεάσουν την ακρίβεια των υπολοίπων που υπολογίζονται και, κατά συνέπεια, το απόλυτο σχετικό υπόλοιπο μεγαλώνει.
- Ο αριθμός συνθήκης παραμένει σταθερός σε όλες τις περιπτώσεις αυτό υποδεικνύει ότι η ευαισθησία του συστήματος στις μικρές αλλαγές δεν επηρεάζεται από το μέγεθος του πίνακα. Το γεγονός αυτό σημαίνει ότι η αριθμητική σταθερότητα του συστήματος παραμένει σχετικά σταθερή ανεξάρτητα από το μέγεθος των δεδομένων.

Σχολιασμός των αποτελεσμάτων για την μέθοδο LL:

2. Μέθοδος Παραγοντοποίησης Cholesky LL^T

2.α Επίλυση του Γραμμικού Συστήματος $Ax = b$				
Διάσταση Πίνακα A	Απ. σχ. Σφάλμα $\frac{\ \delta x\ _\infty}{\ x\ _\infty}$	Απ. σχ. Υπόλοιπο $\frac{\ r\ _\infty}{\ b\ _\infty}$	Αριθμός Συνθήκης $cond(A)$	Χρόνος
100	5e-12	5e-12	3	0.0029s
500	1e-12	1e-12	3	0.4s
1000	5e-13	5e-13	3	3.5s

2.β Υπολογισμός του αντιστρόφου A^{-1}				
Διάσταση Πίνακα A	Απ. σχ. Σφάλμα $\ A^{-1} - \hat{A}^{-1}\ _\infty / \ A^{-1}\ _\infty$	Απ. σχ. Υπόλοιπο $\ A\hat{A}^{-1} - I\ _\infty / \ A^{-1}\ _\infty$	Αριθμός Συνθήκης	Χρόνος
100	1.2e-11	1e-09	3	0.34s
500	2.3e-12	1e-09	3	200s
1000	1.2e-12	1e-09	3	2.8e+03s

- Για τον αριθμό συνθήκης παρατηρούμε ότι όσο και να μεγαλώνει η διάσταση του πίνακα, παραμένει σταθερός που σημαίνει ότι η ευστάθεια της λύσης του συστήματος δεν επηρεάζεται σημαντικά από το μέγεθος του πίνακα. Αυτό υποδεικνύει το γεγονός ότι ακόμα και όταν ο πίνακας γίνεται μεγάλος, η ακρίβεια της λύσης δεν υποβαθμίζεται σημαντικά.
- Για τον χρόνο παρατηρούμε (και στον υπολογισμό του αντιστρόφου και στην επίλυση του γραμμικού συστήματος) ότι αυξάνεται όσο αυξάνεται το μέγεθος των δεδομένων. Επιπλέον, ο υπολογισμός του αντιστρόφου χρειάζεται περισσότερο χρόνο από την επίλυση του γραμμικού συστήματος.
- Για το απόλυτο σχετικό σφάλμα και υπόλοιπο παρατηρούμε (και στην επίλυση του γραμμικού συστήματος και στον υπολογισμό του αντιστρόφου) ότι παραμένει σχετικά σταθερό χωρίς μεγάλες αλλαγές. Το γεγονός αυτό υποδεικνύει ότι παρόλο που μεγαλώνει η διάσταση του πίνακα τα σφάλματα που συμβαίνουν στις πράξεις δεν αυξάνονται. Άρα η αύξηση των δεδομένων δεν συνεπάγεται και αύξηση των σφαλμάτων στις πράξεις και απώλεια δεδομένων.

Σύγκριση δύο μεθόδων:

Παρατηρούμε ότι η μέθοδος LU χρειάζεται περισσότερο χρόνο από την LL για την επίλυση ενός γραμμικού συστήματος και την εύρεση του αντιστρόφου μιας και εκτελεί περισσότερες πράξεις. Η μέθοδος LU έχει μικρότερο απόλυτο σχετικό υπόλοιπο και σφάλμα σε σχέση με την μέθοδο LL που σημαίνει ότι η μέθοδος LL όταν εκτελεί τις πράξεις έχει μεγαλύτερη απώλεια δεδομένων σε σχέση με την μέθοδο LU.

Σημείωση: για τους παραπάνω υπολογισμούς στα πινακάκια έχει χρησιμοποιηθεί σαν A ο Pex matrix για $n = 100, 500, 1000$.