Practical Exercise 8

Ομάδα : 3

Συμμετέχοντες : Μιχάλης Μιχαήλ

Σώτος Βασιλείου

Πασιουρτίδης Κώστας

A person holding a card

Description automatically generated with low confidence

Άσκηση 1

Τόσο στο bisection method όσο και στο False position method οι κώδικες είναι ίδιοι με διαφορά στην τιμή του χ ( βλέπε συνάρτηση solve )

1. Bisection method x=(a+b)/2;
2. False position method x=((a\*fb)-(b\*fa))/(fb-fa);

Ο κώδικας διαβάζει από το αρχείο το διάστημα που θέλουμε να εξερευνήσουμε , τις σταθερές της συνάρτησης και το μέγεθος που πολυωνύμου . Το πρόβλημα βρίσκει λύση της δοσμένης συνάρτησης (δλδ ποιο χ για f(χ) =0 ) εξερευνώντας το διάστημα με την μέγιστη ακρίβεια που του δίνουμε . Το πρόγραμμα επιστρέφει την τιμή το x (f(χ) =0) και τον αριθμό των επαναλήψεων που χρειάστηκε .Παρατηρήστε στον πιο κάτω ψευδοκώδικα ότι η συνάρτηση solve καλείτε αναδρομικά και τερματίζει στις περιπτώσεις όπου είτε βρήκε την λύση της συνάρτησης είτε φτάσαμε την μέγιστη ακρίβεια είτε ξεπεράσαμε τις μέγιστες επιτρεπτές επαναλήψεις .

Void(main){

Fscanf() // διαβασμα από αρχείο

// διάστημα , μέγεθος πολυωνύμου , σταθερές συνάρτησης

Result = Solve (διάστημα , ακρίβεια , επαναλήψεις ) ;

}

Float Solve (διάστημα , ακρίβεια , επαναλήψεις ) {

X= (συναρτήσεις εκφώνησης ανάλογα με μέθοδο )

If (fx= 0 || b-a)<e || iter>N)

Return x ;

if(fx\*f(a,n,c)>0){//an i timi tis sinartisis me to neo x

//epi

// tin timi tis sinartisis me to pio arisera simio

return solve(x,b,e,iter,N,n,c); //deksia tis mesis

}

return solve(a,x,e,iter,N,n,c);// aristera tis mesis 2

}

Float Function(){

for(p=1;p<n+1;p++){

sum=sum + c[p]\*(pow(x,p));

}

}

Graphical user interface, text, application

Description automatically generated Graphical user interface, text, application

Description automatically generated

Άσκηση 2

Αρχικα στη main ανοιγω το αρχειο το διαβαζω και παιρνω τις τιμες αποθηκευωντας τις τιμες αυτές σε ένα δυσδιαστατο πινακα.

ifstream fin

fin.open("ECE325\_system.txt")

int N,i,j

fin>>N

double mat[N][N+1]

for(i=0; i<N; i++){

for ( j = 0; j < N+1; j++)

fin>>mat[i][j]

}

fin.close()

Μετα ξεκινα ο κυριος κωδικας για την ευρεση των λυσεων του συστηματος

**1ο section:**

Εξηγω ακριβως τι γινετε στα σχολια

for (int k=0; k<N; k++)

{

//arxikopioisi gia na vrw megaliteres times

int i\_max = k;

int v\_max = mat[i\_max][k];

/\* vrisko megaliteri timi gia to pivot an iparxei \*/

for ( i = k+1; i < N; i++)

if (abs(mat[i][k]) > v\_max){

v\_max = mat[i][k]

i\_max = i

}

/\* an to stoixio mias kirias diagwniou einai miden

\* tote o pinakas einai monadikos

\* kai tha paei gia diairesi me miden \*/

if (!mat[k][i\_max]){

singular\_flag=k; // Matrix is singular

goto jump

}

/\* swap tin grammi tis megaliteris timis me trexon grammi \*/

**2ο section:**

if (i\_max != k){

for (int l=0; l<=N; l++){

double temp = mat[k][l];

mat[k][l] = mat[i\_max][l];

mat[i\_max][l] = temp;

}

}

for ( i=k+1; i<N; i++)

{

/\* o sintelestis f kani tin trexon grammi kth stixiou isi me 0

\* kai apomenoun kth stiles me 0 \*/

double f = mat[i][k]/mat[k][k];

/\* afairo to fth pollaplasio tou sigekrimenou kth stixiou tis grammis \*/

for (int j=k+1; j<=N; j++)

mat[i][j] -= mat[k][j]\*f;

/\*gemizoto katw trigwniko me midenika\*/

mat[i][k] = 0;

}

}

singular\_flag=-1;

**3ο section:**

Εξηγω ακριβως τι γινετε στα σχολια

double x[N] // pinakas gia na apothikevsw tis lisis dldi ta x1,x2,x3..

/\* ksekino na ipologizo apo tin teleftaia grammi pros tin prwti grammi gia na vrw ta x1,x2..\*/

for ( i = N-1; i >= 0; i--)

{

/\* start with the RHS of the equation \*/

x[i] = mat[i][N]

/\*arxikopoio to j=i+1 afou o matrix einai anw trigonikos\*/

for ( j=i+1; j<N; j++)

{

//afairo ola ta lth stoixia

x[i] -= mat[i][j]\*x[j]

}

x[i] = x[i]/mat[i][i] }

Και τελος τυπώνω τις λυσεις που εχω αποθηκευσει στον πινακα double x[N] ως Χ1, Χ2,Χ3 κλπ.

Άσκηση 2.2 Lu decomposition

Graphical user interface, text, application, email

Description automatically generated

Αρχικά ορίζω τις μεταβλητές όπου θα χρησιμοποιήσω για να κάνω το decomposition και ακολούθως διαβάζω το Ν για να ξέρω το μέγεθος του πίνακα και στην συνέχει διαβάζω Ν στοιχεία και το επόμενο όπου είναι η ορίζουσα μου.

Text

Description automatically generated

Στην συνέχεια δημιουργώ τους 2 πίνακες L και U: όπου Α=LU

A picture containing text, clock

Description automatically generated

Όπου έχουν αυτή την μορφή και βγαίνουν λύνοντας την γραμμική εξίσωση χρησιμοποιώντας των ποιο πάνω αλγόριθμο.

Text

Description automatically generated

Ακολούθως τυπώνονται οι 2 πίνακες L και U.

Graphical user interface, text, application, email

Description automatically generated

Εφόσον έχω τους 2 μου πίνακες με την μέθοδο forward substitution βρίσκω την λύση του συστήματος.

C(n) ∼ n2. k + n = n(n − 1) + n = n2.

Text

Description automatically generated

Τέλος παρουσιάζονται τα αποτελέσματα (x1,x2…..xn)

Πηγές:

<http://blog.gtiwari333.com/2009/12/c-c-code-lu-decomposition-for-solving.html?fbclid=IwAR1sBfpYa038hVCMeHTj48-p6BAVZl1Pvu7JYtOw4KR4uaKbt9DRoGE4u0Q>

https://en.wikipedia.org/wiki/LU\_decomposition