Aishwary Sharma

2022CH11437

Group No. – 1

Tut-6

Bairstow Method:

#include<bits/stdc++.h>

using namespace std;

void lstf(vector<int> v){

for(int i = 0;i<v.size();i++){

cout << v[i]<< " ";

}

}

int root = 1;

void bairstow(int degree,double &r, double &s, vector<double>aCoeff){

int n = degree ;

if(aCoeff.size()!= degree+1){

cout << "coefficient error" ;

return ;

}

double delS, delR ;

double errR =1 ; double errS = 1;

vector<double> bCoeff(degree+1);

while(errR>1e-05 or errS>1e-05){

// creating b(1 to n) vector

bCoeff[n]= aCoeff[n];

bCoeff[n-1]= aCoeff[n-1]+bCoeff[n]\*r;

for(int i = n-2;i>=0;i--){

bCoeff[i]=aCoeff[i] + bCoeff[i+1]\*r + bCoeff[i+2]\*s ;

}

// creating c(1 to n) vector

vector<double> cCoeff(degree+1);

cCoeff[n]= bCoeff[n];

cCoeff[n-1]= bCoeff[n-1]+cCoeff[n]\*r;

for(int i = n-2;i>=1;i--){

cCoeff[i]=bCoeff[i] + cCoeff[i+1]\*r + cCoeff[i+2]\*s ;

}

//Calculating del s and del r

delS = (-bCoeff[0]\*cCoeff[2]+cCoeff[1]\*bCoeff[1])/(cCoeff[2]\*cCoeff[2]-cCoeff[1]\*cCoeff[3]);

delR = (-bCoeff[1]-cCoeff[3]\*delS)/cCoeff[2];

r = r+delR ; s = s+delS ;

errR = abs(delR/(r));

errS = abs(delS/(s));

cout << "r: " << r << " s: "<< s << endl;

for(int i = 0; i<bCoeff.size();i++){

cout<< "b"<<i<<" = " << bCoeff[i]<< " ";

}

cout << endl;

for(int i = 0; i<cCoeff.size();i++){

cout<< "c"<<i<<" = " << cCoeff[i]<< " ";

}

cout << endl ;

cout << "delR = "<< delR << " delS = " << delS << endl ;

cout << "errR = "<< errR << " errS = " << errS << endl;

}

degree = degree -2 ;

if ((r\*r+4\*s) >= 0){

cout<< "root"<< root << " = " << (r + sqrt(r\*r+4\*s))/2 << endl ; root++;

cout<< "root"<< root << " = " << (r - sqrt(r\*r+4\*s))/2 << endl ; root ++ ;

}

else{

cout<< "root"<< root << " = " << (r/2)<<"+"<<sqrt(-(r\*r+4\*s))/2<<"i"<< endl ; root++;

cout<< "root"<< root << " = " << (r/2)<<"-"<<sqrt(-(r\*r+4\*s))/2<<"i"<< endl ; root++;

}

if(degree == 2){

double b2 = bCoeff[2]; double b3 = bCoeff[3]; double b4 = bCoeff[4] ;

if((b3\*b3-4\*b2\*b4)>=0){

cout<< "root"<< root << " = " << (-b3 + sqrt(b3\*b3-4\*b2\*b4))/(2\*b4) << endl ; root++;

cout<< "root"<< root << " = " << (-b3 - sqrt(b3\*b3-4\*b2\*b4))/(2\*b4) << endl ; root++;

}

else{

cout<< "root"<< root << " = " << (-b3/2\*b4)<<"+"<<sqrt(4\*b4\*b2-b3\*b3)/(2\*b4)<<"i"<< endl ; root++;

cout<< "root"<< root << " = " << (-b3/2\*b4)<<"-"<<sqrt(4\*b4\*b2-b3\*b3)/(2\*b4)<<"i"<< endl ; root++;

}

}

else if(degree == 1){

double b2 = bCoeff[2]; double b3 = bCoeff[3];

cout<< "root"<< root << " = " << -b2/b3 << endl ; root++ ;

}

else{

vector<double> v(degree+1) ;

for(int i = 0 ; i< degree+1; i++){

v[i]= bCoeff[i+2];

}

bairstow(degree , r , s , v);

}

}

int main() {

int degree ;

cout << "Degree of polynomial: " ;

cin >> degree ;

cout << "Coefficients of the polynomial (an to a0 ): ";

vector<double>a1(degree+1) ;

for(int i = degree ; i >= 0 ; i--){

cin >> a1[i];

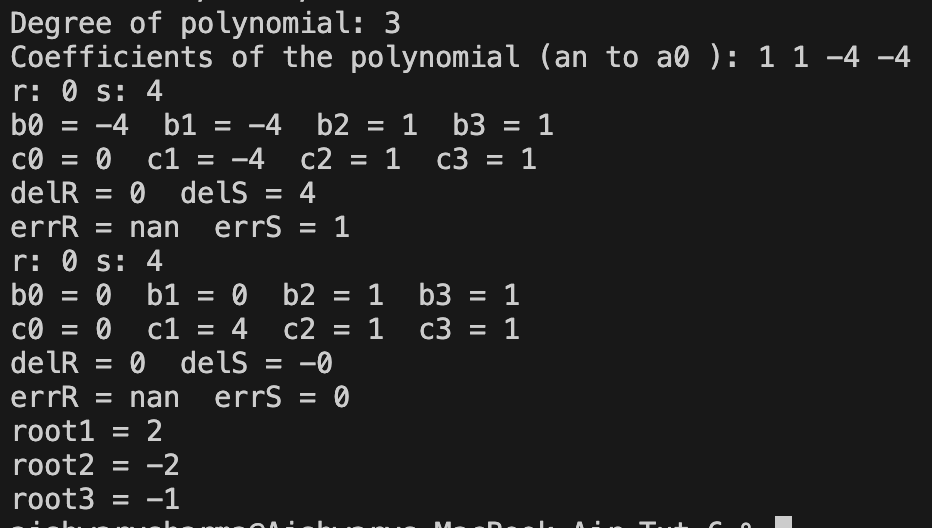
}

double r =0; double s = 0;

bairstow(degree,r,s,a1);

}

(a) f (x) = x3 + x2 − 4x − 4



(b) f (x) = x3 − 0.5x2 + 4x – 2

