**נספח**

**להלן תוכנית המבקשת משוואה מדרגה 3 ומוצאת את השורשים הממשיים שלה:**

/\* cubic\_eq1.c - find real roots of a cubic equation \*/

#include <math.h>

#include <stdio.h>

#define EPS 0.0000000000001

#define PI 3.14159265

int solve\_quadratic(double a, double b,

double c,

double \*r1, double \*i1,

double \*r2, double \*i2)

{

double delta, delta\_root, xv, qv;

if ( a == 0)

if ( b == 0)

return 0;

else

{

\*r1 = -c/b;

\*i1 = 0;

return 1;

} /\* else \*/

xv = -b/(2\*a); /\* Take care of square quad \*/

qv = a\*xv\*xv + b\*xv + c;

if(fabs(qv) < EPS)

{

\*r1 = \*r2 = xv;

\*i1 = \*i2 = 0.0;

return 2;

} /\* if \*/

delta = b\*b - 4\*a\*c;

if ( delta < 0.0 )

{

\*r1 = \*r2 = -b/(2\*a); /\* real part only \*/

delta\_root = sqrt(-delta);

\*i1 = - delta\_root/(2.0\*a);

\*i2 = delta\_root/(2.0\*a);

return 0;

} /\* if \*/

else

{

delta\_root = sqrt(delta);

\*r1 = (-b - delta\_root)/(2.0\*a);

\*r2 = (-b + delta\_root)/(2.0\*a);

\*i1 = 0;

\*i2 = 0;

return 2;

} /\* else \*/

} /\* solve\_quadratic \*/

void divide\_cubic(double a, double \*b,

double \*c, double x)

{

\*b = \*b + a\*x;

\*c = \*c + (\*b)\*x;

} /\* divide\_cubic \*/

void cubic\_root(double x, double y, double \*r, double \*i )

{

double radius, theta, temp;

if ( x == 0.0 && y == 0.0)

{

\*r = \*i = 0.0;

return;

} /\* if \*/

radius = sqrt(x\*x + y\*y);

theta = (asin(fabs(y/radius)));

temp = exp(log(radius)/3.0);

if ((x < 0)&&(y>=0))

theta = 3\*PI - theta;

else

if ((x<0)&&(y<=0))

theta = 3\*PI + theta;

else

if ((x>=0)&&(y<0))

theta=4\*PI-theta;

\*r = temp\*cos(theta/3.0);

\*i = temp\*sin(theta/3.0);

} /\* cubic\_root \*/

int solve\_cubic(double a1, double a2,

double a3, double a4,

double \*x1, double \*x2, double \*x3)

{

double p, q;

double a, b, c;

double u1, u2, wr,wi;

double x, t, ur, ui, vr, vi;

double b2, c2;

double i1, i2;

int n;

if ( a1 == 0.0)

{

n = solve\_quadratic(a2, a3, a4, x1, &i1, x2, &i2);

return n;

} /\* if \*/

a = a2/a1;

b = a3/a1;

c = a4/a1;

p = b - a\*a/3.0;

q = c + (2.0\*a\*a\*a - 9.0\*a\*b)/27.0 ;

if ( ( p==0.0 ) && (q == 0.0))

{

\*x1 = \*x2 = \*x3 = -a/3;

return 3;

} /\* if \*/

if (p == 0.0) /\* q != 0 \*/

{

cubic\_root(q,0, &ur, &ui);

x = ur - (a/3.0);

\*x1 = x;

b2 = a;

c2 = b;

divide\_cubic(1.0, &b2, &c2, x);

n = solve\_quadratic(1.0, b2, c2, x2,&i1, x3, &i2);

return n+1;

} /\* if \*/

solve\_quadratic(1.0, -q, -((p\*p\*p)/27.0), &u1, &i1, &u2, &i2);

if (u1 > u2)

{

wr = u1;

wi = i1;

} /\* if \*/

else

{

wr = u2;

wi = i2;

} /\* else \*/

cubic\_root(wr, wi, &ur, &ui);

cubic\_root(-q + wr, wi, &vr, &vi);

t = vr - ur;

x = t - (a/3.0);

\*x1 = x;

b2 = a;

c2 = b;

divide\_cubic(1.0, &b2, &c2, x);

n = solve\_quadratic(1.0, b2, c2, x2,&i1, x3, &i2);

return n+1;

} /\* solve\_cubic \*/

int main()

{

double r,i;

double a, b, c, d;

double x1, x2, x3;

int no\_of\_solutions;

printf("Enter 4 coefs, first must be =/= 0:\n");

scanf("%lf %lf %lf %lf", &a, &b, &c, &d);

no\_of\_solutions = solve\_cubic(a, b, c, d, &x1, &x2, &x3);

if (no\_of\_solutions == 1)

printf("Single real solution is: %lf\n", x1);

else /\* 3 real solutions \*/

printf("Solutions : %lf %lf %lf\n", x1, x2, x3);

return 0;

} /\* main \*/

דוגמאות ריצה:

D:\>cubic\_eq1.exe

Enter 4 coefs, first must be =/= 0:

1 -6 11 -6

Solutions : 2.000000 1.000000 3.000000

D:\>cubic\_eq1.exe

Enter 4 coefs, first must be =/= 0:

1 2 -1 -2

Solutions : 1.000000 -2.000000 -1.000000

D:\>cubic\_eq1.exe

Enter 4 coefs, first must be =/= 0:

2 -30 162 -350

Single real solution is: 7.000000

D:\>cubic\_eq1.exe

Enter 4 coefs, first must be =/= 0:

1 -1 -1 1

Solutions : -1.000000 1.000000 1.000000

D:\>cubic\_eq1.exe

Enter 4 coefs, first must be =/= 0:

1 -5 1 -5

Single real solution is: 5.000000

D:\>cubic\_eq1.exe

Enter 4 coefs, first must be =/= 0:

1 -6 -1 30

Solutions : -2.000000 3.000000 5.000000

D:\> cubic\_eq1.exe

Enter 4 coefs, first must be =/= 0:

1 1 1 1

Single real solution is: -1.000000

D:\> cubic\_eq1.exe

Enter 4 coefs, first must be =/= 0:

1 0 0 8

Single real solution is: 2.000000

D:\> cubic\_eq1.exe

Enter 4 coefs, first must be =/= 0:

1 0 0 -8

Single real solution is: -2.000000

D:\> cubic\_eq1.exe

Enter 4 coefs, first must be =/= 0:

1 -6 -67 360

Solutions : -8.000000 5.000000 9.000000

D:\> cubic\_eq1.exe

Enter 4 coefs, first must be =/= 0:

1 3 3 1

Solutions : -1.000000 -1.000000 -1.000000

D:\> cubic\_eq1.exe

Enter 4 coefs, first must be =/= 0:

1 -3 3 -1

Solutions : 1.000000 1.000000 1.000000

D:\> cubic\_eq1.exe

Enter 4 coefs, first must be =/= 0:

1 0 1 0

Single real solution is: 0.000000

D:\> cubic\_eq1.exe

Enter 4 coefs, first must be =/= 0:

1 -2 1 0

Solutions : 0.000000 1.000000 1.000000

D:\> cubic\_eq1.exe

Enter 4 coefs, first must be =/= 0:

1 -5 6 0

Solutions : 0.000000 2.000000 3.000000

D:\>

**התוכנית הבאה מדפיסה את שלושת הפתרונות של המשוואה, גם אם הם קומפלקסיים**

/\* cubic\_eq2.c - find real roots of a cubic equation \*/

#include <math.h>

#include <stdio.h>

#define EPS 0.0000000000001

#define PI 3.14159265

int solve\_quadratic(double a, double b,

double c,

double \*r1, double \*i1,

double \*r2, double \*i2)

{

double delta, delta\_root, xv, qv;

if ( a == 0)

if ( b == 0)

return 0;

else

{

\*r1 = -c/b;

\*i1 = 0;

return 1;

} /\* else \*/

xv = -b/(2\*a); /\* Take care of square quad \*/

qv = a\*xv\*xv + b\*xv + c;

if(fabs(qv) < EPS)

{

\*r1 = \*r2 = xv;

\*i1 = \*i2 = 0.0;

return 2;

} /\* if \*/

delta = b\*b - 4\*a\*c;

if ( delta < 0.0 )

{

\*r1 = \*r2 = -b/(2\*a); /\* real part only \*/

delta\_root = sqrt(-delta);

\*i1 = - delta\_root/(2.0\*a);

\*i2 = delta\_root/(2.0\*a);

return 0;

} /\* if \*/

else

{

delta\_root = sqrt(delta);

\*r1 = (-b - delta\_root)/(2.0\*a);

\*r2 = (-b + delta\_root)/(2.0\*a);

\*i1 = 0;

\*i2 = 0;

return 2;

} /\* else \*/

} /\* solve\_quadratic \*/

void divide\_cubic(double a, double \*b,

double \*c, double x)

{

\*b = \*b + a\*x;

\*c = \*c + (\*b)\*x;

} /\* divide\_cubic \*/

void cubic\_root(double x, double y, double \*r, double \*i )

{

double radius, theta, temp;

if ( x == 0.0 && y == 0.0)

{

\*r = \*i = 0.0;

return;

} /\* if \*/

radius = sqrt(x\*x + y\*y);

theta = (asin(fabs(y/radius)));

temp = exp(log(radius)/3.0);

if ((x < 0)&&(y>=0))

theta = 3\*PI - theta;

else

if ((x<0)&&(y<=0))

theta = 3\*PI + theta;

else

if ((x>=0)&&(y<0))

theta=4\*PI-theta;

\*r = temp\*cos(theta/3.0);

\*i = temp\*sin(theta/3.0);

} /\* cubic\_root \*/

int solve\_cubic(double a1, double a2,

double a3, double a4,

double \*x1, double \*y1,

double \*x2, double \*y2,

double \*x3, double \*y3)

{

double p, q;

double a, b, c;

double u1, u2, wr,wi;

double x, t, ur, ui, vr, vi;

double b2, c2;

double i1, i2;

int n;

if ( a1 == 0.0)

{

n = solve\_quadratic(a2, a3, a4, x1, &i1, x2, &i2);

return n;

} /\* if \*/

a = a2/a1;

b = a3/a1;

c = a4/a1;

p = b - a\*a/3.0;

q = c + (2.0\*a\*a\*a - 9.0\*a\*b)/27.0 ;

if ( ( p==0.0 ) && (q == 0.0))

{

\*x1 = \*x2 = \*x3 = -a/3;

\*y1 = \*y2 = \*y3 = 0;

return 3;

} /\* if \*/

if (p == 0.0) /\* q != 0 \*/

{

cubic\_root(q,0, &ur, &ui);

x = ur - (a/3.0);

\*x1 = x;

\*y1 = 0;

b2 = a;

c2 = b;

divide\_cubic(1.0, &b2, &c2, x);

n = solve\_quadratic(1.0, b2, c2, x2,y2, x3, y3);

return n+1;

} /\* if \*/

solve\_quadratic(1.0, -q, -((p\*p\*p)/27.0), &u1, &i1, &u2, &i2);

if (u1 > u2)

{

wr = u1;

wi = i1;

} /\* if \*/

else

{

wr = u2;

wi = i2;

} /\* else \*/

cubic\_root(wr, wi, &ur, &ui);

cubic\_root(-q + wr, wi, &vr, &vi);

t = vr - ur;

x = t - (a/3.0);

\*x1 = x;

\*y1 = 0;

b2 = a;

c2 = b;

divide\_cubic(1.0, &b2, &c2, x);

n = solve\_quadratic(1.0, b2, c2, x2,y2, x3, y3);

return n+1;

} /\* solve\_cubic \*/

int main()

{

double r,i;

double a, b, c, d;

double x1, x2, x3;

double y1, y2, y3;

int no\_of\_solutions;

printf("Enter 4 coefs, first must be =/= 0:\n");

scanf("%lf %lf %lf %lf", &a, &b, &c, &d);

no\_of\_solutions = solve\_cubic(a, b, c, d, &x1, &y1, &x2, &y2, &x3, &y3);

if (no\_of\_solutions == 1)

{

printf("Single real solution is: %lf\n", x1);

printf("Complex solution: %lf", x2);

if ( y2 < 0)

printf(" %lfi\n", y2);

else

printf("+ %lfi\n", y2);

printf("Complex solution: %lf", x2);

if ( y3 < 0)

printf(" %lfi\n", y3);

else

printf("+ %lfi\n", y3);

} /\* if \*/

else /\* 3 real solutions \*/

printf("Real Solutions : %lf %lf %lf\n", x1, x2, x3);

return 0;

} /\* main \*/

**פלט רוצה**

D:\>cubic\_eq2.exe

Enter 4 coefs, first must be =/= 0:

1 0 0 8

Single real solution is: 2.000000

Complex solution: -1.000000 -1.732051i

Complex solution: -1.000000+ 1.732051i

D:\>