להלן תוכנית הפותרת משוואה בשיטת דקר dekker:

/\* dekker3.c \*/

#include <stdio.h>

#include <math.h>

long double pi = 3.1415926535897932;

long double f(long double x)

{

return sinl(x\*pi/180.0) - cosl(2.0\*x\*pi/180.0);

} /\* f \*/

long double dekker( long double (\*fun)(long double), long double a,

long double b, long double eps)

{

long double x, f, s, m, bk, bk1, funa, funb, funbk, funbk1;

int flag;

funa = (\*fun)(a);

funb = (\*fun)(b);

do {

if ( fabsl(funa) < fabsl(funb))

{

bk1 = b;

funbk1 = funb;

bk = a;

funbk = funa;

flag = 1;

}// if

else

{

bk1 = a;

funbk1 = funa;

bk = b;

funbk = funb;

flag = 2;

}// else

m = (a+b)/2.0;

s = bk - ((bk - bk1)/((\*fun)(bk) -(\*fun)(bk1)))\* (\*fun)(bk);

if (flag == 1)

if( (s >= a) && (s <=m))

x = s;

else

x = m;

else

if( (s <= b) && (s >=m))

x = s;

else

x = m;

f = (\*fun)(x);

if (fabsl(f) < eps)

return x;

if ( funa\*f < 0.0)

{

b = x;

funb = f;

}// if

else

{

a = x;

funa = f;

}// else

} while( fabs(b-a) > eps);

return x;

} /\* dekker \*/

int main()

{

long double xstar;

int i;

xstar = dekker(f, 0.0, 70.0, 0.0000001);

printf(" sin(%Lf) = cos(%Lf) \n", xstar, 2\*xstar);

} /\* main \*/

פלט ריצה:

sin(30.000000) = cos(60.000000)

התוכנית הבאה פותרת משוואה בעזרת השיטה של ברנט brent:

/\* brent.c - brent method \*/

#include <stdio.h>

#include <math.h>

void swap(long double \*x, long double \*y)

{

long double temp;

temp = \*y;

\*y = \*x;

\*x = temp;

}// swap

long double brent(long double (\*fun)(long double),

long double a, long double b, long double eps)

{

long double fa, fb, fc, fs, c, c0, c1, c2,temp, mtflag, d, s;

int i, mflag;

c = a;

d = c;

fa = (\*fun)(a);

fb = (\*fun)(b);

fc = (\*fun)(c);

if ( fa\*fb >= 0)

return 0.0;

if ( fabsl(fa) < fabsl(fb))

{

swap(&a, &b);

swap(&fa, &fb);

} // if

mflag = 1;

while ( (fabsl(fb) > eps) && ( fabsl(b-a) > eps))

{

if ( (fa != fc) && fb != fc)

{

c0 = a\*fb\*fc/((fa-fb)\*(fa-fc));

c1 = b\*fa\*fc/((fb-fa)\*(fb-fc));

c2 = c\*fa\*fb/((fc-fa)\*(fc-fb));

s = c0 + c1 + c2;

} // if

else

s = b - fb\*(b-a)/(fb - fa);

if ( ( s < (3\*(a+b)/4) || s > b) || ( (mflag == 1) &&

fabsl(s-b) >= (fabsl(b-c)/2) ) ||

( (mflag == 0) && fabsl(s-b) >= (fabsl(c-d)/2) ) )

{

s = (a+b)/2;

mflag = 1;

} //if

else

mflag = 0;

fs = (\*fun)(s);

d = c;

c = b;

fc = fb;

if ( (fa\*fs)< 0)

b = s;

else

a = s;

if ( fabsl(fa) < fabsl(fb))

{

swap(&a, &b);

swap(&fa, &fb);

} // if

} // while

return b;

} /\*brent \*/

long double mypoly(long double x)

{

return (x\*x\*x - 100.0);

} /\* mypoly \*/

long double mypolyd(long double x)

{

return (3\*x\*x);

}

int main ()

{

long double x;

x = brent(mypoly, 3.0, 5.0, 0.0000001);

printf("\nSolution to mypoly(x) = 0, x = %Lf, f(%Lf) = %Lf\n",

x, x, mypoly(x));

return 0;

} /\* main \*/

פלט ריצה:

Solution to mypoly(x) = 0, x = 4.641589, f(4.641589) = 0.000002