Apraksts

Šajā laboratorijas darbā tiks apskatīts funkcijas sakņu meklēšana ar dihotomijas metodes palīdzību, tas ir, dalot funkcijas doto intervālu vairākas reizes un vērtejot, kādi ir funkcijas intervāla galapunktu reizinājumi.

Algoritma apraksts

Lai noteiktu saknes funkcijai konkrētajā definīcijas apgabalā, tiek izmantota dihotomijas metode: ja konkrētā definīcijas apgabala galapunktis $f(x_o)$ un $f(x_1)$ vērtību reizinājums ir pozitīvs, tad tas nozīmē, ka funkcija nav krustojusi *Ox* asi, bet ja reizinājums ir negatīvs, tad funkcija ir krustojusi *Ox* asi un ir saknes.

Konkrēta funkcija un kods

Kā par piemēru, tiks izmantota man uzdota funkcija $f(x)=cos(x/2)$. Vispirms tiek prasītas funkcijas definīcijas apgabala galapunkti, kā arī precizitātes vērtība, lai zinātu, cik precīzi vēlamies iegūt sakni.

```
int main(){
        float a, b, x, delta_x, funkca, funkcb, funkcx;
        int k=0;
        printf("Cien.lietotāj, ievadiet a vērtību:\n");
        scanf("%f",&a);
        printf("Cien.lietotāj, ievadiet b vērtību:\n");
        scanf("%f",&b);
        printf("Cien.lietotāj, ievadiet precizitātes vērtību:\n");
        scanf("%f",&delta_x);
        funkca = cos(a/2);
        funkcb = cos(b/2);
        if(funkca*funkcb>0){
                printf("Intervālā [%.2f;%.2f] cos(x/2) funkcijai",a,b);
                printf(" sakņu nav (vai tajā ir pāru sakņu skaits)\n");
                return 1;
        }
```

Tālāk tiek noteiktas funkcijas vērtības doto definīcijas apgabala galapunktiem. Ja šo vērtību reizinājums ir pozitīvs, tad tajā definīcijas apgabalā nav sakņu vai arī ir pāra skaits sakņu (par to pēc tam). Savukārt, ja tiek konstatēs, ka reizinājums ir negatīvs, tas nozīmē, ka kaut kur ir sakne šajā apgabalā. Tāpēc šo definīcijas apgabalu dala uz pusēm tik daudz reižu, kamēr tas ir lielāks par doto precizitāti.

```
while((b-a)>delta_x){
        k++;
        x=(a+b)/2;
        if(funkca*cos(x/2)>0){
                a = x;
        }else{
                b = x;
        }
```

Pēc koda var secināt, ka šis process norisināsies tik daudz kamēr tiks atrasta aptuvena saknes vērtība (atkarībā no prasītās precizitātes) un procesa norišu reizies, tas ir, iterācijas tiek uzskaitītas ar *k* koeficienta palīdzību.

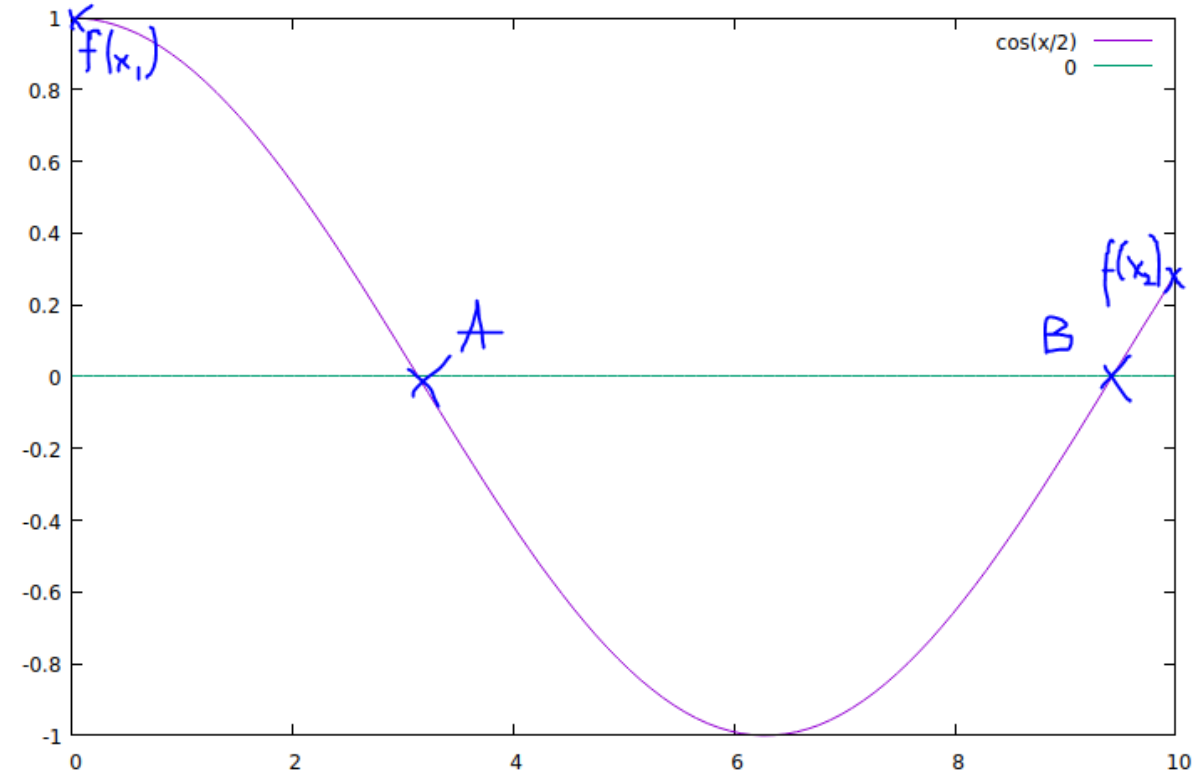
Kā uzdevuma 2.daļa tiek prasīts, ka lietotājam jāievada funkcijas vērtība, lai noskaidrotu šai funkcijas vērtībai atbilstošo argumenta vertību, tādejādi, atvieglojot funkcijas saknes meklēšanu dotajā definīcijas apgabalā.

```
rezultats = 2*acos(c);
y = cos(rezultats/2);
printf("Ievadītā c vērtības argumenta vērtība: %.4f\n",rezultats);
printf("Aprēķinātā funkcijas vērtība iegūtajam argumentam: %.4f\n",y);
return 0;
```

Kā var redzēt, tiek izmantota inversā funkcija dotajai funkcijai, tas ir, *f(c)=2*arccos(c)*. Tādejādi tiks iegūta un izvadīta argumenta vērtība pie dotās funkcijas vērtības, kā arī pēc tam veikta pretējā operācija pārbaudes dēļ.

Pāra saknes

Kā jau tika minēts, šī programma var noteikt divas lietas: ir vai nav saknes dotajā definīcijas apgabalā. Ja ir viena sakne, tad tā tiks izvadīta, taču ja nav sakņu vai arī ir vairākas, tad tas izvadīs citu rezultātu. Problēma, kāpēc ar šo metodi nevar tikt izvadītas pāra skaita saknes, ir tāda, ja ir pāra sakņu skaits, tad funkcija vērtība būs ar tādu pašu zīmi kā sākotnējā vērtība un rezultātā abu funkcijas vērtību būs pozitīva, ko mūsu programma uztvers, ka "sakņu nav". Tāpēc papildus jāliek brīdinājums, ka, ja ir vairākas pāra skaita saknes, tad rezultāts nebūs "nav sakņu", bet gan "ir pāra skaits sakņu".



Ilustrācija 1. Pāra sakņu problēma: ja ir pāra skaits sakņu, vienādas f(x) vērtību zīmes, līdz ar to reizinājums būs pozitīvs.

Papildus materiāli

```
Cien.lietotāj, ievadiet a vērtību:
0
Cien.lietotāj, ievadiet b vērtību:
5
Cien.lietotāj, ievadiet c vērtību:
0
Cien.lietotāj, ievadiet precizitātes vērtību:
0.001
```

Ilustrācija 2. Ievadītās lietotāja vērtības.

```
cos(  0.000/2)=  1.000          cos(  5.000/2)= -0.801
 1.iterācija: cos(  2.500/2)=  0.315    cos(  2.500/2)=  0.598  cos(  5.000/2)=
 -0.801
 2.iterācija: cos(  2.500/2)=  0.315    cos(  3.750/2)= -0.572  cos(  3.750/2)=
 -0.300
 3.iterācija: cos(  3.125/2)=  0.008    cos(  3.125/2)=  0.017  cos(  3.750/2)=
 -0.300
 4.iterācija: cos(  3.125/2)=  0.008    cos(  3.438/2)= -0.292  cos(  3.438/2)=
 -0.147
 5.iterācija: cos(  3.125/2)=  0.008    cos(  3.281/2)= -0.139  cos(  3.281/2)=
 -0.070
 6.iterācija: cos(  3.125/2)=  0.008    cos(  3.203/2)= -0.061  cos(  3.203/2)=
 -0.031
 7.iterācija: cos(  3.125/2)=  0.008    cos(  3.164/2)= -0.022  cos(  3.164/2)=
 -0.011
 8.iterācija: cos(  3.125/2)=  0.008    cos(  3.145/2)= -0.003  cos(  3.145/2)=
 -0.001
 9.iterācija: cos(  3.135/2)=  0.003    cos(  3.135/2)=  0.007  cos(  3.145/2)=
 -0.001
10.iterācija: cos(  3.140/2)=  0.001    cos(  3.140/2)=  0.002  cos(  3.145/2)=
 -0.001
11.iterācija: cos(  3.140/2)=  0.001    cos(  3.142/2)= -0.000  cos(  3.142/2)=
 -0.000
12.iterācija: cos(  3.141/2)=  0.000    cos(  3.141/2)=  0.001  cos(  3.142/2)=
 -0.000
13.iterācija: cos(  3.141/2)=  0.000    cos(  3.141/2)=  0.000  cos(  3.142/2)=
 -0.000
Sakne atrodas pie x=3.141,jo cos(3.141/2) ir 0.000
```

Ilustrācija 3. Iterāciju skaits sakņu vērtības aptuvenai noteikšanai.

```
Sakne atrodas pie x=3.141,jo cos(3.141/2) ir 0.000
Ievadītā c vērtības argumenta vērtība: 3.1416
Aprēķinātā funkcijas vērtība iegūtajam argumentam: -0.0000
```

Ilustrācija 4. Rezultāti ar dihotomijas algoritmu un inverso funkciju.