# Apraksts

Šajā laboratorijas darbā tiks pētīts, kā ar dažādām metodēm var aprēķināt laukumu zem funkcijas grafika, tas ir, aprēķināt noteikto integrāli.

## Taisnstūra metodes kods un rezultāts

Šajā metodē, lai aprēķinātu laukumu zem funkcijas grafika, izmantosim mazos taisnstūrīšus. Algoritms ir sekojošs: sākumā aprēķinam lielajam taisnstūrim, kura garums ir *b-a* un platums ir *f((a+b)/2)*. Tad mēs salīdzinam iegūto laukumu ar iepriekš iegūto laukumu, kas šajā gadījumā ir 0. Kamēr izmaiņa starp laukumiem ir lielāka par ievadīto precizitāti, šis apgabals tiek dalīts arvien mazāko intervālos, no kā iegūti arvien mazāki taisnstūrīši un precīzāks ir rezultāts.

```
printf("Cien.lietotaj, ievadiet a:\n");
scanf("%f",&a);
printf("Cien.lietotaj, ievadiet b:\n");
scanf("%f",&b);
printf("Cien.lietotaj, ievadiet precizitati:\n");
scanf("%f",&eps);
```

*Ilustrācija 1. Vispirms tiek prasīti lietotājam parametri.*

```
int taisnsturis(){
	integr2=(b-a)*(cos(((a+b)/n)/2));
	while(fabs(integr2-integr1)>eps){
		n*=2;
		h=(b-a)/n;
		integr1=integr2;
		integr2=0;
		for(k=0;k<n;k++){
			integr2+=h*cos((a+(k+0.5)*h)/2);
		}
	}
	printf("Integrala vertiba: %.5f\n",integr2);
}
```

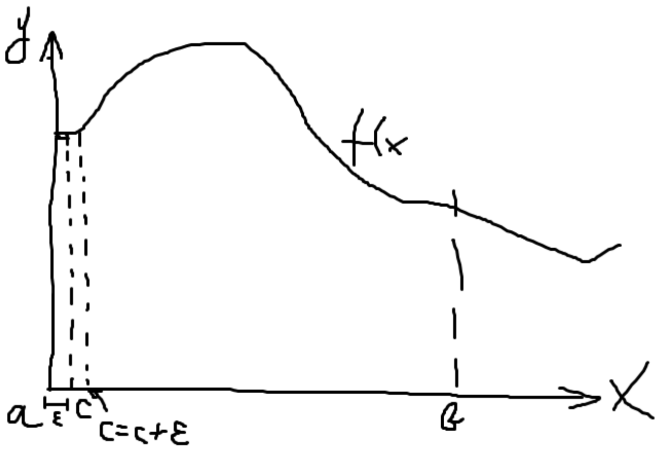
*Ilustrācija 2. Tālāk seko funkcija aprakstītajam algoritmam. Dotā funkcija ir f(x)=cos(x/2)*

```
Cien.lietotaj, ievadiet a:
0
Cien.lietotaj, ievadiet b:
3.1415
Cien.lietotaj, ievadiet precizitati:
0.001
Integrala vertiba: 2.00020
```

*Ilustrācija 3. Izvadītais rezultāts funkcijai cos(x/2) intervālā no 0 līdz π.*

## Trapeces kods un rezultāts

Šajā metodē algoritms būs mazliet citādāks. Šeit mēs pieņemsim jaunu mainīgo $c$, kas būs $c = a + \varepsilon$.




*Ilustrācija 4. Aptuvenā reprezentācija trapeces algoritmam.*

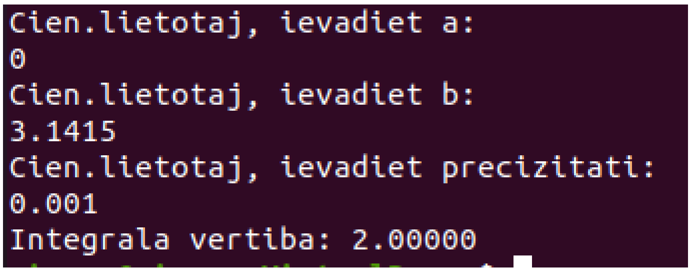
Kā var redzēt 4.attēlā, pašai pirmajai trapecei tiks noteiktas robežas $a$ un $c=a+\varepsilon$, kas atbilst uzdotajai precizitātei. Pēc tam tiks noteiktas funkcijas vērtības $f(a)$ un $f(c)$, no kurām tiks noteikts trapeces augstums un trapeces laukums. Tad trapeces laukums tiek pieskaitīts mainīgajam $S$, kas pirms pirmās trapeces ir 0.

Tad mainīgajiem $a$ un $c$ pieskaita vērtību $\varepsilon$, lai varētu aprēķināt laukumu nākamajai trapecei, kuras platums atbilst dotajai precizitātei ($a=a+\varepsilon$ un $c=c+\varepsilon$).

Tā process tiek atkārtots, kamēr mainīgais $c$ nav vienāds ar $b+\varepsilon$ (ja raksta ciklu līdz $c<b$, tad tiek izlaists viena trapece, tāpēc vajag rakstīt līdz $c<b+\varepsilon$).

```
int trapece(){
        float c=a+eps,S,s1;
        while(c<b+eps){
                float s1 =(c-a)*((cos(a/2)+cos(c/2))/2);
                S+=s1;
                a=a+eps;
                c=c+eps;
        }
        printf("Integrala vertiba: %.5f\n",integr2);
}
```

*Ilustrācija 5. Kods trapeces metodes aprakstītajam algoritmam.*



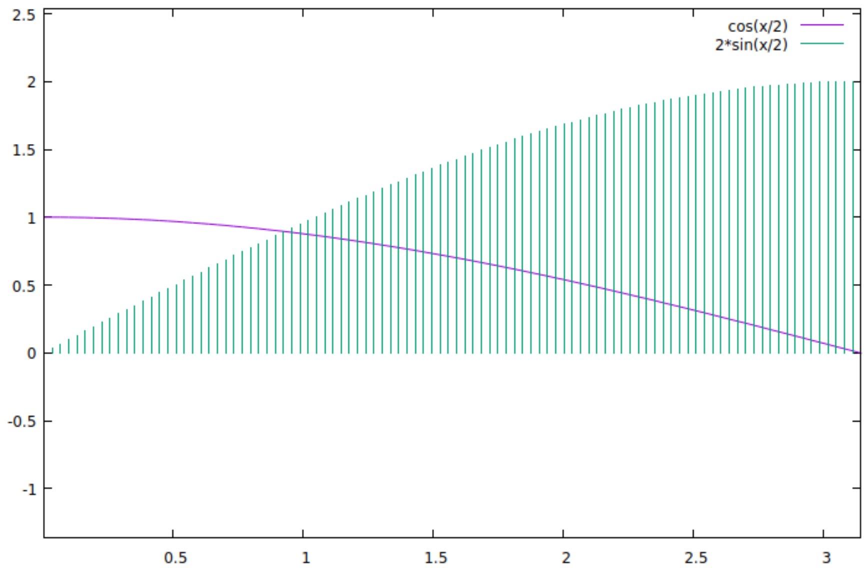
```
Cien.lietotaj, ievadiet a:
0
Cien.lietotaj, ievadiet b:
3.1415
Cien.lietotaj, ievadiet precizitati:
0.001
Integrala vertiba: 2.00000
```

*Ilustrācija 6. Izvadītais rezultāts ar trapeces algoritmu.*

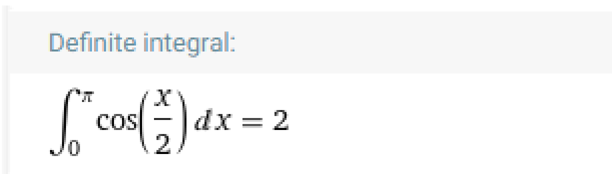
## Secinājumi

Abi algoritmi ļoti labi strādā noteiktā integrāļa vērtību aprēķināšanai ar lietotāja sniegto precizitāti. Abos gadījumos tika izmantota funkcija *f(x)=cos(x/2)*, kā arī robežas no 0 līdz π, taču, ievadot arī citas vērtības, var konstatēt, ka abi algoritmi strādā ar diezgan lielu precizitāti. Protams, rezultātos var redzēt sīkas atšķirības, taču tās ir mazākas par ε vērtību, tāpēc būtiskas atšķirības šajiem rezultātiem nav.

## Pielikums




*Ilustrācija 7. Doti 2 grafiki: cos(x/2)(oriģinālfunkcija) un 2*sin(x/2)(integrētā funkcija oriģinālfunkcijai). Zaļās strīpas reprezentē laukumu, kas tika aprēķināts robežās [0;π]*



Definite integral:

$$\int_0^{\pi} \cos\left(\frac{x}{2}\right) dx = 2$$

*Ilustrācija 8. Integrāļa rezultāts ar Wolfram Alpha palīdzību.*