

4. uzdevums

Sastādīt programmu, kura aprēķina laukumu zem funkcijas $y=ax^3+bx^2+cx+d$ grafika intervālā $[u,w]$ ar 1) taisnstūru metodi 2) trapeču metodi. a, b, c, d, u, w un precizitāti ievada lietotājs. Salīdzināt abas metodes pēc veikto iterāciju skaita pie vienas un tās pašas precizitātes.

Kods:

```
# Programmas nosaukums: 4. uzd MPR12
```

```
# 4. uzdevums MPR12
```

```
# Uzdevuma formulējums: Sastādīt programmu, kura aprēķina laukumu zem funkcijas
y=ax^3+bx^2+cx+d grafija intervālā [u,w] ar 1) taisnstūru metodi 2) trapeču metodi. a, b, c, d, u, w un
precizitāti ievada lietotājs. Salīdzināt abas metodes pēc veikto iterāciju skaita pie vienas un tās pašas
precizitātes.
```

```
# Versija 1.0
```

```
print("Programma noteic laukumu zem funkcijas y = ax^3 + bx^2 + cx + d intervāla [u,w].\nJa
laukums ir zem X ass, tad laukums ir negatīvs.\n")
```

```
q = float(input("Ievadi a ==> "))
```

```
w = float(input("Ievadi b ==> "))
```

```
e = float(input("Ievadi c ==> "))
```

```
r = float(input("Ievadi d ==> "))
```

```
a = float(input("Ievadi sākumpunktu u ==> "))
```

```
b = float(input("Ievadi galapunktu w ==> "))
```

```
pr = float(input("Ievadi precizitāti ==> "))
```

```
#----- ar taisnsturiem
```

```
s2 = 0
```

```
n=2
```

```
paz = True
```

```

while paz :

    s1 = s2

    x = (b - a) / n

    s2 = 0

    for i in range (n) :

        y=a+i*x

        s2 = s2 + (y*y*y*q+y*y*w+y*e+r) * x

    n = n * 2

    if abs(s1 - s2) < pr :

        paz = False

print("\nS ar taisnturiem = " + str(s2))
print("Iterāciju skaits: " + str(n) + "\n")

```

#----- ar trapecem

```

n = 2

s2 = 0

paz = True

while paz :

    s1 = s2

    x = (b - a) / n

    s2 = 0

    c = a*a*a*q+a*a*w+a*e+r

    for i in range (1,n+1) :

        y=a+x*i

        d = y*y*y*q+y*y*w+y*e+r

        s2 = s2 + x * (c + d) / 2

        c = d

```

```

n = n * 2

if abs(s1 - s2) < pr :

    paz = False

print("S ar trapecem = " + str(s2))

print("Iterāciju skaits: " + str(n))

```

Testa piemēri:

1)

```

Programma noteic laukumu zem funkcijas  $y = ax^3 + bx^2 + cx + d$  intervāla  $[u,w]$ .
Ja laukums ir zem X ass, tad laukums ir negatīvs.

Ievadi a ==> 1
Ievadi b ==> 0
Ievadi c ==> 0
Ievadi d ==> 0
Ievadi sākumpunktu u ==> -2
Ievadi galapunktu w ==> 2
Ievadi precizitāti ==> 0.1

S ar taisnturiem = -0.0625
Iterāciju skaits: 1024

S ar trapecem = 0.0
Iterāciju skaits: 4

```

2)

```

Programma noteic laukumu zem funkcijas  $y = ax^3 + bx^2 + cx + d$  intervāla  $[u,w]$ .
Ja laukums ir zem X ass, tad laukums ir negatīvs.

Ievadi a ==> 1
Ievadi b ==> 0
Ievadi c ==> -4
Ievadi d ==> 0
Ievadi sākumpunktu u ==> -2
Ievadi galapunktu w ==> 2
Ievadi precizitāti ==> 0.001

S ar taisnturiem = 0.0
Iterāciju skaits: 4

S ar trapecem = 0.0
Iterāciju skaits: 4

```

3)

```
Programma noteic laukumu zem funkcijas  $y = ax^3 + bx^2 + cx + d$  intervāla  $[u,w]$ .  
Ja laukums ir zem X ass, tad laukums ir negatīvs.
```

```
Ievadi a ==> 1  
Ievadi b ==> 0  
Ievadi c ==> -4  
Ievadi d ==> 4  
Ievadi sākumpunktu u ==> -2  
Ievadi galapunktu w ==> 2  
Ievadi precizitāti ==> 0.001
```

```
S ar taisnturiem = 16.0  
Iterāciju skaits: 8
```

```
S ar trapecem = 16.0  
Iterāciju skaits: 8
```

4)

```
Programma noteic laukumu zem funkcijas  $y = ax^3 + bx^2 + cx + d$  intervāla  $[u,w]$ .  
Ja laukums ir zem X ass, tad laukums ir negatīvs.
```

```
Ievadi a ==> 1  
Ievadi b ==> 3  
Ievadi c ==> 5  
Ievadi d ==> 2  
Ievadi sākumpunktu u ==> 0  
Ievadi galapunktu w ==> 4  
Ievadi precizitāti ==> 0.01
```

```
S ar taisnturiem = 175.9919434487815  
Iterāciju skaits: 65536
```

```
S ar trapecem = 176.00146484375  
Iterāciju skaits: 512
```

5)

```
Programma noteic laukumu zem funkcijas  $y = ax^3 + bx^2 + cx + d$  intervāla  $[u,w]$ .  
Ja laukums ir zem X ass, tad laukums ir negatīvs.
```

```
Ievadi a ==> -2  
Ievadi b ==> 3  
Ievadi c ==> 6  
Ievadi d ==> 1  
Ievadi sākumpunktu u ==> 0  
Ievadi galapunktu w ==> 2  
Ievadi precizitāti ==> 0.1
```

```
S ar taisnturiem = 13.937255859375  
Iterāciju skaits: 256
```

```
S ar trapecem = 13.984375  
Iterāciju skaits: 32
```

Vairumā gadījumu iterāciju skaits pie vienas un tās pašas precizitātes ir mazāks, izmantojot trapeču metodi nekā izmantojot taisnstūru metodi. Dažos gadījumos iterāciju skaits ir vienāds. Varam secināt, ka, izmantojot trapeču metodi, programmai ir jāveic mazāks vai vienāds iterāciju skaits, lai tiktu pie rezultāta nekā izmantojot taisnstūru metodi.