Apraksts

Šajā laboratorijas darbā tika pētīts, kā ar Teilora rindu palīdzību ir iespējams ar ļoti lielu precizitāti atdarināt oriģinālas funkcijas. Manā konkrētajā piemērā tika izmantota funkcija *f(x)=cos(x/2)*.

Oriģinālā funkcija

Kods ir uzrakstīts tā, ka lietotājs var ievadīt jebkādu reāla skaitļa vērtību funkcijas vērtības noteikšanai. Par piemēru izmantošu x vērtību 2,05. Izmantojot oriģinālo kosinusu no pusargumenta funkciju, tiek iegūts, ka

```
Cien.lietotāj, lūdzu, ievadi vērtību:
2.05.
cos(x/2) aprēķināšana:
Ievadi argumentu x/2: 1.025
cos(1.025)=0.519
```

Iemesls, kapēc atsevišķi pirms kosinusa funkcijas tika sadalīts arguments, bija, ka, ja tieši kosinusā ieliek dalīšanas funkciju, tad sistēma var veikt aritmētiskās darbības nepareizajā secībā, tāpēc es nodrošinājos ar to, ka vispirms tiek izmantots pusarguments un tad tiek izmantota kosinusa funkcija. Kā var redzēt, *f(2,05)=cos(2,05/2)=0,519*.

Teilora rinda

Teilora rinda funkcijai *cos(x/2)* ir sekojoša:

$$\cos(x/2) \simeq \sum_{k=0}^{500} \frac{(-1)^k \cdot x^{2\cdot k}}{(2\cdot k)! \cdot 2^{2\cdot k}}$$

No kā seko, ka kosinuss no pusargumenta būs līdzīgs pie dotās funkcijas vairākkārtējas saskaitīšanas (ap 500 reizes). Kods dotajai rindai ir sekojošs:

```
a = pow(-1,k)*pow(x,2*k)/(1*pow(2,2*k));
S = a;

printf("a0 = %.5Lf\t S0 = %.5Lf\n",a,S);
for(k=1;k<500;k++){
        a = a * (-1)*x*x/((2*k)*(2*k-1));
        S = S + a;
        //printf("%.100f\t%8.100f\t%8.10f\n",x,a,S);
}
printf("a499 = %.300Lf\n S499 = %.300Lf\n",a,S);
k = 500;
a = pow(-1,k)*pow(x,2*k)/(1*pow(2,2*k));
S = S + a;
printf("a500 = %.300Lf\n S500 = %.10Lf\n",a,S);
```

Princips ir sekojošs: vispirms tiek aprēķināts pirmais loceklis, kad *k=0*. Par cik summa sākumā ir 0, tad tai tiek pieskaitīts pirmais loceklis. Tālāk *a* saskaitāmā aprēķināšanā tiek pielietots *rekurences reizinātājs.* Tas ir :

$$r = \frac{a(n)}{a(n-1)}$$

kur a_n ir rindas loceklis, savukārt, a_{n-1} ir rindas iepriekšējas loceklis pirms a_n.

Rekurences reizinātājs manai funkcijai ir:

$$r = -\frac{x\hat{}2}{2k*(2k-1)}$$

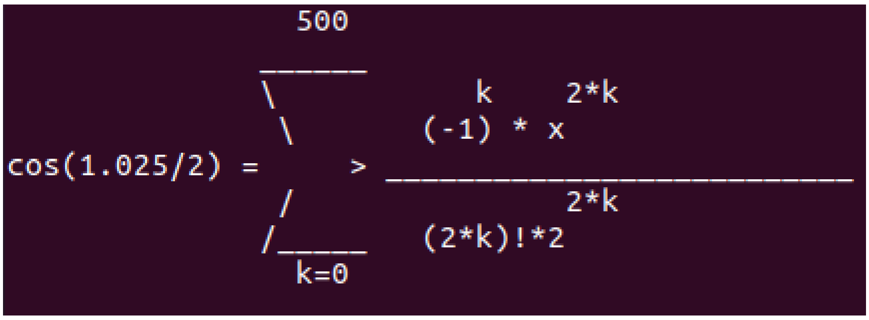
Tālāk iegūtais rekurences reizinātājs tiks izmantots, katru iegūto *a* locekli sareizinot ar *r*, iegūstot nākamo locekli a_{n+1}. Pēc tam katrs loceklis tiks pieskaitīts kopējai summai. Kā var redzēt kodā, pēc 0.kārtas locekļa seko cikls, kurā ar katru *k* tiek aprēķināts *r, r* tiek piereizināts *a*, iegūstot jaunu a_{n+1} un tas tiek pieskaitīts kopējai summai *S*. Iegūtais rezultāts ir:

```
a499 = -0.0000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000
 S499 = 0.5190988868333696869294437148401044623824418522417545318603515625000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000
a500 =  0.0000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000004941624344
 S500 = 0.5190988868
```

Kā var spriest, pie ļoti lielām *k* vērtībām *a* lockelis paliek arvien mazāks un mazāks, tas nekad nebūs 0, bet bezgalīgi mazs skaitlis. Līdz ar to summētā rinda tiek konverģēta, kas nozīmē, ka *S* tuvosies uz konkrētu skaitli. Šajā gadījumā šis skaitlis būs ļoti līdzīgs funkcijas vērtībai *f(x)=cos(x/2)*.

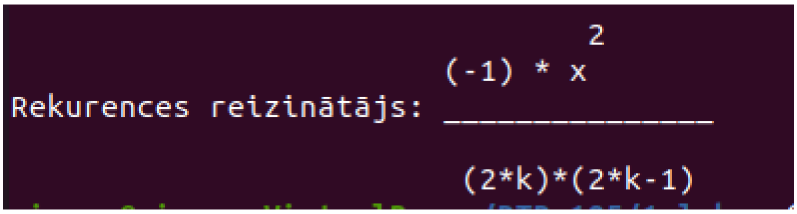
Secinājumi: Teilora rindas metode ir ļoti efektīva dažādu funkcijas vērtību atrašanai ar ļoti lielu precizitāti. Oriģinālfunkcijas vērtības viennozīmīgi nebūs identiski vienādas ar Teilora rindas summēšanas rezultātu, taču summētais rezultāts būs pietiekami precīzs, lai varētu izmantot kā aizstājējfunkciju oriģinālfunkcijai.

Papildus materiāli



```
                  500
                ______
                \          k     2*k
                 \     (-1) * x
cos(1.025/2) =    >   __________________________
                 /               2*k
                /_____   (2*k)!*2
                  k=0
```

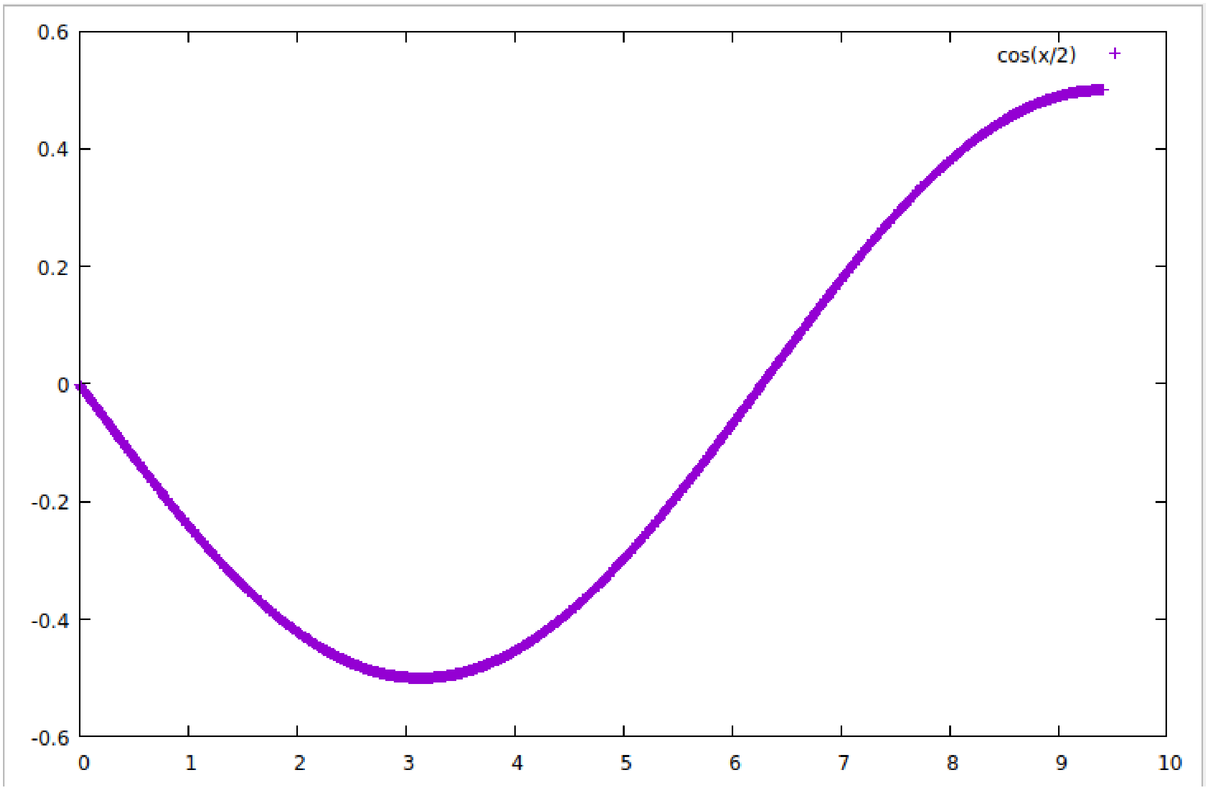
Ilustrācija 1. ASCII zīmējums dotajai Teilora rindai



```
                                          2
                              (-1) * x
Rekurences reizinātājs: _______________

                               (2*k)*(2*k-1)
```

Ilustrācija 2. ASCII zīmējums iegūtajam rekurences reizinātājam.



Ilustrācija 3. Ar GNUPLOT iegūtā f(x)=cos(x/2) funkcija.