BENDRASIS PROGRAMAVIMO DOKUMENTAS	EUROPOS SĄJUNGA Europos socialinis fondas	ŠVIETIMO IR MOKSLO MINISTERIJA	
KURKIME ATEITĮ DRAUGE!			

2004-2006 m. Bendrojo programavimo dokumento 2 prioriteto „Žmogiškųjų išteklių plėtra“ 4 priemonė „Mokymosi visą gyvenimą sąlygų plėtra“

Projekto sutarties numeris: **ESF/2004/2.4.0-K01-160/SUT-261**

Projekto pavadinimas: **Inovatyvūs mokymosi metodai ir naujausios technologijos gamtos mokslų bakalaurų rengimui**

FIZ 313 KOMPIUTERINĖ FIZIKA

Laboratorinis darbas

FORTRAN FUNKCIJŲ TAIKYMAS NEIŠREIKŠTINŲ LYGČIŲ SPRENDIMUI

1. Darbo tikslas:

1. Pagerinti sudėtingų funkcijų išreiškimo ir skaičiavimo įgūdžius FORTRAN programavimo kalba sprendžiant lygtį dalinimų metodu.

2. Darbo užduotis:

2.1. Susipažinti su fortran matematinėmis funkcijomis bei jų naudojimu.

2.2. Parašyti programą, kuri rastų funkcijos $f(x)$ šaknį $f(x) = 0$ intervale $x \in [-1, 3]$ ne mažiau kaip 8 skaitmenų tikslumu.

3. Bendroji teorija

FORTAN programos struktūra

Tokios programos, kai visas uždavinyje naudojamų duomenų apdorojimo procesas yra tiesiogiai aprašomas FORTRAN operatoriais, tinka tiktai paprastiems uždaviniams. Didesnį uždavinį lengviau programuoti dalimis. Kuo savarankiškesnės dalys ir kuo mažiau ryšiu tarp jų, tuo lengviau sudaryti ir nagrinėti programą.

Tokia programa paprastai suskirstoma į atskiras mažesnes programas, viena iš kurių yra *pagrindinė* programa, o kitos - *procedūros*. Jose naudojami duomenys ir atliekami veiksmai yra aprašomi atskirai ir jos turi savo vardus. Tose programos vietose, kuriose reikia atlikti procedūros veiksmus, pakanka parašyti kreipimąsi į ją. Į vieną kartą aprašytą procedūrą galima kreiptis daug kartų iš įvairių programos vietų ir atlikti jų veiksmus. Sudėtingesniais atvejais paprogramės gali iškviesti kitas, žemesnio lygio, paprogrames.

FORTRANo kalba parašytos programos programinius vienetus galima suskirstyti į tokias grupes:

pagrindinė programa;

modulis;

išorinės procedūros;

vidinės procedūros;

duomenų blokas.

Procedūros paprastai esti dviejų tipų: *funkcijos* ir *paprogramės*. ***Paprogramė*** – tai savarankiškas, savo vardą turintis programinis vienetas, prie kurio pavadinimo stovi operatorius **SUBROUTINE**. Esminis paprogramės bruožas tai, kad pagrindinėje programoje ji iškviečiama operatoriumi **CALL**.

Funkcija – tai savarankiškas, savo vardą turintis programinis vienetas, prie kurio pavadinimo stovi operatorius **FUNCTION**. Esminis funkcijos bruožas tai, kad pagrindinėje programoje iškvietus funkciją, ji sugrąžina į iškvietimo vietą konkrečią vertę, kuri gaunama atlikus funkcijoje užprogramuotų veiksmų seką.

Naudojimo požiūriu programiniai vienetai skirstomi į

įdiegtąsias procedūras;

prijungiamas procedūras ir modulius;

projektuojamas procedūras ir modulius.

Įdiegtos procedūros – tai visos funkcijos ir paprogramės, kurios būna įtrauktos į standartines **FORTRAN** bibliotekas, pavyzdžiui, matematinės funkcijos **sin(x)**, **cos(x)**, **exp(x)** arba išvesties/įvesties funkcijos **print** arba **read**. ***Procedūrų biblioteka*** – tai naudojimui paruoštas mašininis kodas. Kompliatorius, programos tekste suradęs kreipimąsi į tokią funkciją, automatiškai įtraukia į ruošiamą vykdomąjį failą.

Su įdiegtosiomis procedūromis jau susidūrėme nagrinėdami fortran pavyzdžius. Jų yra keturių tipų:

1) elementariosios procedūros;

2) informacinės funkcijos;

3) pertvarkomosios funkcijos;

4) neelementariosios paprogramės.

Paprastai visos vidinės paprogramės ir funkcijos būna aprašytos pagalbiniuose žinynuose. Labai išsamų žinyną turi ir FPS 4.1 sistema. Todėl čia aprašysime, remdamiesi svarbiausių funkcijų pavyzdžiais, tik jų vartojimo principus.

Elementariosios procedūros

Kaip jau buvo minėta, įdiegtosios procedūros būna keturių tipų. ***Elementariosios procedūros*** beveik visos yra funkcijos. Jas galima suskirstyti į tokias rūšis:

a) kintamojo tipo keitikliai (pvz., INT(a) paverčia kintamąjį sveikuoju tipu;

b) skaitinės funkcijos (pvz., FLOOR(a) grąžina didžiausią sveiką skaičių, mažesnį už a);

c) minimalios ir maksimalios reikšmės radimo funkcijos (pvz., MIN(a1, a2, a3, ...));

d) matematinės funkcijos labai plati grupė, skirtos skaičiavimams: eksponentės, logaritmo, šaknies traukimo, kėlimo laipsniu, trigonometrinės funkcijos.

Informacinės procedūros

Informacinės funkcijos taip pat gana plati klasė funkcijų, tačiau jų plačiau neaptarinėsime. Gal tik panagrinėkime vieną kitą tokios funkcijos pavyzdį. Funkcija

HUGE(x) grąžina maksimalią galima x reikšmę.

PRECISION(x) grąžina sveiką skaičių, kuris nurodo, kokio tikslumo x (kiek skaitmenų po kablelio).

Funkcija EPSILON(x) grąžina dydį 2^{1-P}, kur P – bitų skaičius, atvaizduojant realų skaičių dvejetainėje sistemoje. Dydį 2^{1-P} dar vadina ***mašininiu tikslumu***.

```
program infoPVZ
      real a; real(8) b; ; integer c
      print *, ' huge: ', huge(a), huge (b), , huge (c)
      print *, ' precision: ', precision(a), precision(b), precision(c)
      print *, ' epsilon: ', epsilon(a), epsilon(b)
end program
```

1 pav. Skaitinių informacinių funkcijų vartojimo pavyzdys.

Remiantis šiuo pavyzdžiu ištyrinėkite savo kompiuterio skaičiavimo galimybes ir gautus rezultatus palyginkite tarpusavyje.

Pertvarkomosios funkcijos

Pertvarkomosios funkcijos yra gana specializuotos funkcijos, dauguma jų skirtos darbui su bitais. Jų čia neaptarinėsime.

Neelementariosios procedūros

Neelementariosios paprogramės – sprendžia taip pat gana specifines užduotis, tačiau tarp jų galima rasti gana įdomių pritaikymų.

a) paprogramės darbui su operatyviąja atmintimi (pvz., funkcija MALLOC(i) – išskiria i baitų atminties);

b) funkcijos darbui su tekstu;

c) failo pabaigos funkcija EOF() - susipažinsime, kai nagrinėsime įvesties ir išvesties procedūras;

d) datos ir laiko paprogramės;

e) statistinės funkcijos skirtos atsitiktinių skaičių generacijai.

Elementariosios funkcijos

Šiame skyrelyje trumpai apžvelgsime funkcijas, reikalingas fizikinių uždavinių sprendimui kompiuteriu. Tai daugiausia elementariosios matematinės funkcijos.

Skaitinės funkcijos

Dažniausiai pasitaikančios skaitinės funkcijos pateiktos 1 lentelėje. Šioje grupėje nėra daug funkcijų, jų sąrašus, aprašymą bei naudojimo pavyzdžius galima rasti fortran žinynuose.

1 lentelė. Dažniausiai pasitaikančios skaitinės funkcijos.

Funkcija	Aprašymas	Pavyzdys	
CEILING(a)	Realiam a suranda mažiausią sveikąjį didesnį už a	ceiling(5.3) ceiling(-2.1)	6 -2
FLOOR(a)	Realiam a suranda mažiausią sveikąjį mažesnį už a	floor(5.3 floor(-2.1)	5 -3
NINT(a)	Skaičiaus apvalinimas	nint(5.3) nint(-2.8)	5 -3

Maksimumo ir minimumo funkcijos

Dažniausiai pasitaikančios ,aksimumo ir minimumo radimo funkcijos pateiktos 2 lentelėje. Šioje grupėje nėra daug funkcijų, tik keletas gimininingų funkcijų toms, kurios nurodytos 4.2 lentelėje.

2 lentelė. Dažniausiai pasitaikančios maksimumo ir minimumo funkcijos.

Funkcija	Aprašymas	Pavyzdys	
MAX(a1,a2,...)	Iš dviejų ir daugiau skaičių randa didžiausią	max(5.3, 5.1)	5.3
MIN(a1, a2,...)	Iš dviejų ir daugiau skaičių randa mažiausią	min(5.3, 5.1)	5.1

Laipsnio kėlimo funkcijos

Yra kelios funkcijos, kurios susijusios su skaičiaus laipsnio kėlimu, t.y., eksponentė, logaritmai ir kvadratinė šaknis. Jos aprašytos 3 lentelėje. Reikia neužmiršti, kad praktiškai visos funkcijos turi sau gimininingą funkciją, kuri atlieka dvigubo tikslumo skaičiavimus. Tokių funkcijų pirkeyje rašoma raidė D, pavyzdžiui, exp(x) grąžina rezultatą e*x kaip paprastą realų skaičių, o funkcija **dexp(x_8)** – realų dvigubo tikslumo skaičių. Šiuo atveju, argumentas taip pat turi būti dvigubo tikslumo skaičius).

3 lentelė. Funkcijos susijusios su skaičiaus kėlimu laipsniu.

Funkcija	Aprašymas	Pavyzdys	
EXP(x)	Suskaičiuoja e^x=e***x*. Jei kompleksinis x=(a,b), tai e**x·a(cos((b)+isin(b))	exp(3.4)	29.96
DEXP(x)	Tas pat, tik dvigubo tikslumo	dexp(3.4)	29.96

LOG(x)	Suskaičiuoja natūrinį logaritmą	log(3.4)	1.22
LOG10(x)	Suskaičiuoja dešimtainį logaritmą	log10(3.4)	0.53
SQRT(x)	Ištraukia kvadratinę šaknį iš x	sqrt(3.4)	1.84

Žemiau pateiktas pavyzdys, kuriame parodytas laipsnio kėlimo funkcijų taikymas. Norisi atkreipti dėmesį, kad sqrt(x) gali būti parašytas ir x**0.5, tačiau antruoju atveju kompiuterinis kėlimas laipsniu skaičiavimai užtrunka ilgiau negu kadratinės šaknies traukimas, todėl derėtų vartoti sqrt(x).

```
real :: x1=3.4
real(8) :: x2 = 3.4, r2
real r1, r3, r4, r5, r6

    r1 = exp(x1), r2 = dexp(x2)
    print *, r1,r2
    r3 = log(x1), r4 = log10(x1)
    print *, r3,r4
    r5 = sqrt(x2), r6 = x2**0.5
    print *, r5,r6
end
```

2 pav. Laipsnio kėlimo funkcijų taikymas.

2 pavyzdyje pateikta programa atspausdina ekrane tokius rezultatus:

```
   29.964100                    29.964102904996410
     1.223776   5.314789E-01
     1.843909   1.843909
```

Trigonometrinės funkcijos

Didžiulę funkcijų klasė sudaro trigonometrinės funkcijos (4 lentelė). Kiekvienai trigonometrinei funkcijai yra jos dvigubo tikslumo atitikmuo – priekyje rašoma raidė D (pvz., DSIN(x)). Be to, kiekvienos trigonometrinės funkcijos argumentas gali būti išreikštas ir radianais, ir laipsniais. Įprastai fortranas supranta argumentą kaip radianus (SIN(x)), o prirašius raidę D iš dešinės (degree) – kaip laipsnius: SIND(x).

4 lentelėje vienai funkcijai (SIN) pateikiami visi galimi variantai (dvigubo tikslumo ir argumentai laipsniais ir radianais), o visoms kitoms – tik po vieną pavyzdį.

4 lentelė. Trigonometrinės funkcijos.

Funkcija	Aprašymas
SIN(x)	Sinusas skaičiaus x, x - radianais
DSIN(x)	Sinusas skaičiaus x, dvigubo tikslumo, x radianais
SIND(x)	Sinusas skaičiaus x, x laipsniais
DSIND(x)	Sinusas skaičiaus x, dvigubo tikslumo, x laipsniais
ASIN(x)	arcsin(x), $\|x\| \leq 1$; rezultatas intervale $[-\pi/2, \pi/2]$
ASIND(x)	arcsin(x), $\|x\| \leq 1$; rezultatas intervale [-90, 90]

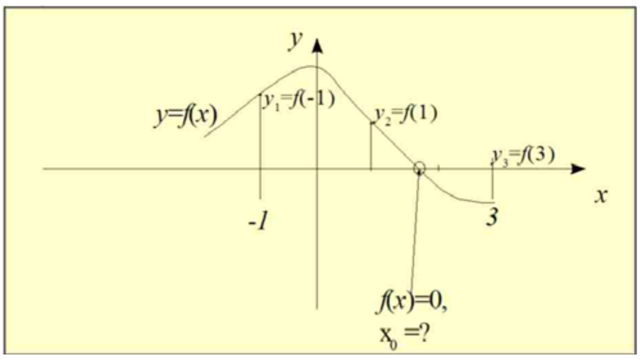
COS(x)	x - radianais
ACOS(x)	arccos(x) , $|x| \leq 1$; rezultatas $[0, \pi]$
TAN(x)	tan(x)
COTAN(x)	Ctan(x)
ATAN(X)	arctan(x), rezultatas $[-\pi/2, \pi/2]$
SINH(x)	Hiperbolinis sinusas
COSH(x)	Hiperbolinis kosinusas
TANH(x)	Hiperbolinis tangentas

4. Tyrimo metodika

Sprendžiant įvairaus tipo fizikos uždavinius tenka susidurti tiek su paprastomis, tiek su sudėtingomis funkcijomis. Matematiniu požiūriu galimybės išspręsti polinomo tipo lygtis gerai žinomos: pirmojo laipsnio ($ax + b = 0$) ir antrojo laipsnio (kvadratinė $ax^2 + bx + c = 0$) lygtis galima išspręsti visada, trečiojo laipsnio lygtis analitiškai išsprendžiama ne visada. Vienas iš charakteringiausių trečiojo laipsnio lygčių pavyzdžių fizikoje – tai Van der Valso lygtis. Aukštesnio laipsnio lygtis išspręsti analitiškai įmanoma tik labai retais atvejais. Be to, fizikoje neretos ir neišreikštinio pavidalo lygtys, kada ieškomojo kintamojo neįmanoma išsireikšti. Atsiradus kompiuteriams tokias lygtis patogu spręsti skaitiniais metodais. Tokių metodų pavyzdžiai yra Niutono ir dalinimų (*bisection*) metodai, kurie aprašyti ir FPS 4.1 informacinėje sistemoje: *Fortran 90 for Scientists and Engineers 16 skyriuje Intruduction to Numerical methods*.

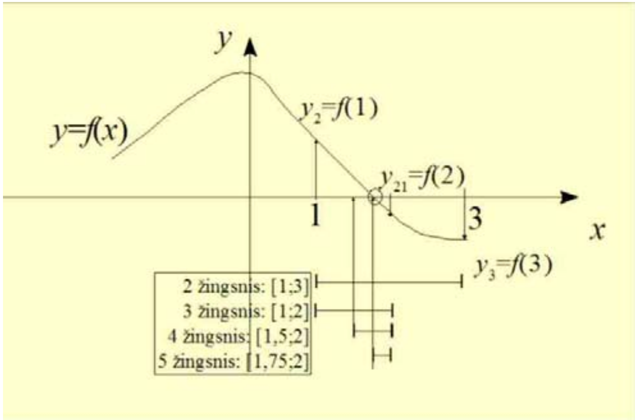
Dalinimo metodas

3 paveiksle pavaizduota kreivės $y = f(x)$ dalis. Tarkime, kad kreivė kerta x ašį vieną kartą. Išspręsti lygtį $f(x) = 0$ geometriniu požiūriu reiškia rasti kreivės sankirtos su abscisių ašimi koordinatę x_0. Dalinimo metodas remiasi tuo, jog pirmame artėjime daliname nagrinėjamą intervalą į dvi lygias dalis ir apskaičiuojame funkcijos $y = f(x)$ reikšmes trijuose taškuose – intervalo galuose ir vidurio taške: $y_1 = f(-1)$, $y_2 = f(1)$, $y_3 = f(3)$. Dabar tikriname kuriame intervalo dalyje $y = f(x)$ keičia ženklą. Iš pateikto grafiko matome, kad $y_1 > 0$ ir $y_2 > 0$, vadinasi, šiame intervale $f(x)$ nekerta x ašies. Kitame intervale matome, kad $y_2 > 0$, $y_3 < 0$. Tai reiškia, kad intervale tarp 1 ir 3 $f(x)$ kirto x ašį.



3 pav. Lygties *f*(*x*) = 0 sprendimas dalinimų metodu: pirmas žingsnis.

Tada atliekame antrą žingsnį: išmetame intervalą, kuriame funkcija ašies nekerta (t.y., išmetame [-1,1]), ir pakartojame dalinimo procedūrą sumažintame intervale [1;3] (4 pav.). Patikriname funkcijos vertes trijuose taškuose, t.y., intervalo kraštuose 1 ir 3 bei viduriniame taške x = 2. Kaip matome, $y_2 > 0$, o $y_{21} < 0$, todėl funkcija kerta x ašį intervale [1, 2].



4 pav. Lygties *f*(*x*) = 0 sprendimas dalinimų metodu: 2 – 5 žingsniai.

Trečias žingsnis: daliname intervalą per pusę, skaičiuojame y vertę trijuose taškuose ir gauname, kad $f(x)$ kerta x ašį intervale [1.5; 2].

Geometriškai atvaizduoti dalinimo algoritmą darosi sudėtinga, nes intervalo plotis mažėja, tačiau skaitiškai šią procedūrą galima tęsti iki norimo tikslumo.

Veikiančios programos pavyzdys pateiktas žemiau. Aprašytu metodu ištirta funkcija $f(x) = 3x - 2$. Pasitikrinti lygties $f(x) = 0$ sprendinį labai lengva: $x_0 = 2/3$.

```
 ******************** bisection1.f90  **********
! dalinimų metodas - pavyzdys
! funkcija f(x) = 3x - 2
program bisection1
     implicit none
     real y1, y2, y3                     !skaičiuosim tris f(x) vertes
     real xmaz     ! kairysis intervalo rėžis
     real xdid     ! keičiamas  dešinysis intervalo rėžis
     real dx       ! dx nurodys intervalo vidurį
     real :: a = -1., b = 3.                             ! pradiniai intervalo rėžiai
     real :: tikslumas = 0.00001    ! skaičiavimo tikslumas
     xmaz = a                       ! nustatom mažesnįjį pradinį rėžį
     xdid = b                       ! nustatom didesnįjį rėžį
     dx = (xdid - xmaz) /2.         !  kur intervalo vidurys

     do while (dx > tikslumas)
          y1 = 3 * xmaz - (2.)
          y2 = 3 * (xmaz + dx) - (2.)
          y3 = 3 * xdid - (2.)
! intervalui kairėje tikrinam
          if (((y1 > 0) .and. (y2<0)) .or. ((y1 < 0) .and. (y2>0))) then
               xmaz = xmaz
               xdid = xdid - dx
               dx = (xdid - xmaz) /2.
          else
! intervalui dešinėje tikrinam
          if (((y2 > 0) .and. (y3<0)) .or. ((y2 < 0) .and. (y3>0))) then
```

```
            xmaz = xmaz + dx
            xdid = xdid
            dx = (xdid - xmaz) /2.
         else
            print *, "nekeicia zenklo ne viename intervale"
            stop
         endif
      endif
      print *, "xmaz = ", xmaz, "   xdid = ", xdid, "   dx = ", dx
   enddo
   print *, " Galutinis atsakymas: f(x) sprendinys yra intervale"
   print *, " [ ", xmaz, ";", xdid, "]"
end
```

<u>10 dalinimų metodas</u>

Intervalą, kuriame ieškomas lygties $f(x) = 0$ sprendinys, galima dalinti ir į daugiau dalių. Vartotojui patogu, kad jo rezultato tikslumas didėtų ne 2 kartais, kaip aprašytame metode, o 10 kartų, kaip mums įprasta dešimtainėje skaičiavimo sistemoje. Tokiu atveju uždaviniu kompiuterinio sprendimo algoritmą sudarys tokie žingsniai:

1) skaičiavimams reikalingų kintamųjų parinkimas ir pradinių reikšmių priskyrimas, t.y., mažesniojo ir didesnio intervalo rėžio bei pradinio žingsnio parinkimas;

2) nustatymas intervalo, kuriame funkcija keičia ženklą (t.y., $f(x)$ kerta x abscisių ašį);

3) mažesniojo ir didesniojo intervalo rėžių perskaičiavimas bei žingsnio perskaičiavimas (galima aiškumo dėlei išvesti pasikeitimus į ekraną);

4) žingsnio palyginimas su reikalaujamu tikslumu:

   4.1) jei gautas žingsnis didesnis už pasirinktą tikslumą, toliau vykdyti 2 punktą;

   4.2) jei gautas žingsnis mažesnis už pasirinktą tikslumą, toliau vykdyti 5 punktą;

5) išvesti gautus rezultatus ir baigti darbą.

<u>5. Tyrimo eiga</u>

1. Laboratorinis darbas atliekamas kompiuterių klasėje.

2. Išanalizuokite formulę, kurią parenka dėstytojas iš priedo Nr.1. Susipažinkite su Fortan matematinėmis funkcijomis, reikalingomis apskaičiuoti reikšmę pagal duotą formulę.

3. Parašykite procedūrą, kuri leistų suskaičiuoti funkcijos reikšmę duotam argumentui.

4. Pritaikykite dešimties dalinimų algoritmą ir parašykite programą, kuri nustato koordinatę, kur $f(x)$ kerta x abscisių ašį intervale [-1,3] 8 skaitmenų po kablelio tikslumu.

5. Rašant programą laikytis tvarkingo rašymo taisyklių, konstrukcijų blokai turi išsiskirti vizualiai pagal „reljefinį“ kodo užrašymą.

6. Atsiskaitomų darbų projektai turi būti saugomi tvarkingoje asmeninių failų struktūroje. Atsiskaitymo metu mokėti paaiškinti su fortran programos kompliavimu ir vykdymu susijusius klausimus FPS 4.1 sistemoje.

7. Atsiskaitymas vyksta prie kompiuterio klasėje, būtina mokėti paaiškinti pagrindinius programos struktūros elementus ir jos veikimą.

6. Kontroliniai kiausimai

1. Loginė programos struktūra, procedūrų reikalingumo pagrįstumas.
2. Fortan matematinių procedūrų bendrosios savybės ir naudojimo ypatybės.
3. Funkcijų apibrėžtis, viengubo ir dvigubo tikslumo funkcijos.
4. Dalinimų metodas.
5. Procedūrų formalieji ir faktiniai parametrai.
6. Dalinimų metodo tikslumas.
7. Pateiktosios programos struktūra, veikimo principas.

7. Literatūra

1. A. Kanapickas. Paskaitų konspektas. 4 skyrius: vidinės funkcijos.

2. MS FPS 4.1 pagalbos sistema: *Microsoft Developer Studio User's Guide -> Chapter 16.1 -> introduction to numerical methods – equations..*

3. R. Belevičius, R. Kutas. FORTRANAS. Vadovėlis. V.: 1998. 241 p., 7 skyrius (paprogramiai)..

Priedas Nr. 1. Neišreikštinų funkcijų pavyzdžiai.

1.	$\frac{1}{3}e^{-\cos^2 x} + \sqrt{x+\frac{9}{7}} / (1+\ln(x+3)) - \frac{4}{5}$
2.	$\frac{15}{9} - (\sqrt{\frac{1}{3}+\frac{x}{5}} + \sqrt[3]{\lvert x\rvert}) / e^{-\frac{x}{3}}$
3.	$\cos\frac{x}{3} - e^{-(x+1)^2} - \frac{4}{9}$
4.	$\cos(4\ln(x+\frac{7}{3})) - \ln(4-\frac{x}{7})/3 - \frac{4}{11}$
5.	$e^{-\sin\frac{2x}{5}} - e^{\cos\frac{x}{2}} + \sqrt{\frac{4}{3}+x}$
6.	$20{,}7 + (\sin^2(1{,}2x) - \arccos\frac{x}{8}) \cdot e^{1{,}5x}$
7.	$\sqrt{e^{-\frac{x}{2}} - 0{,}1} - x\cos(3x-1{,}5)$
8.	$1{,}3x - 2{,}5\sin(5\cdot\sqrt{\frac{4}{3}+arctgx} - 0{,}7)$
9.	$1{,}3 + tg\cos\frac{\pi x}{2{,}3} - \sqrt[3]{(\ln(\frac{5}{3}+x)+1)^2}$
10.	$\pi + \ln\left\lvert\frac{4}{7} - \frac{\sin(arctgx)}{2}\right\rvert$
11.	$e^{\frac{4x}{5}} + 2\sin(7\ln(x+\frac{5}{3})) - \pi$
12.	$\sin(\ln(x+2)) - \cos(\pi\ln(x+\frac{5}{3})) + \frac{x}{5}$
13.	$\cos 1{,}5x - e^{\sin(x+\frac{4}{3})} + \sqrt{x+\frac{7}{6}}$
14.	$e^{-\cos\sqrt{x+\frac{5}{3}}} - 1{,}7arctg(\frac{x}{5}-\frac{3}{4})\cdot\sin 1{,}7x$
15.	$(x+\frac{7}{6})^{\frac{4}{3}} + \sin e^x + \arcsin(\cos\pi x)$
16.	$\ln(\frac{5}{3}+x) - \sin\sqrt{\frac{7}{3}+x} + \frac{1}{7}$
17.	$8x/(\frac{70}{3} + 7\sqrt{\frac{7}{6}+x}) - e^{-\sin^2(\frac{2x}{3})}$
18.	$\arcsin\sqrt{\frac{1}{3}+\frac{x}{7}} - 1{,}5\sin\ln(x+\frac{4}{3})$
19.	$2\sin\frac{4{,}5x+0{,}2}{4} + \sqrt[5]{(x+\frac{7}{4})^2}\cdot e^{-\frac{x}{4}}$