Navadna iteracijska metoda

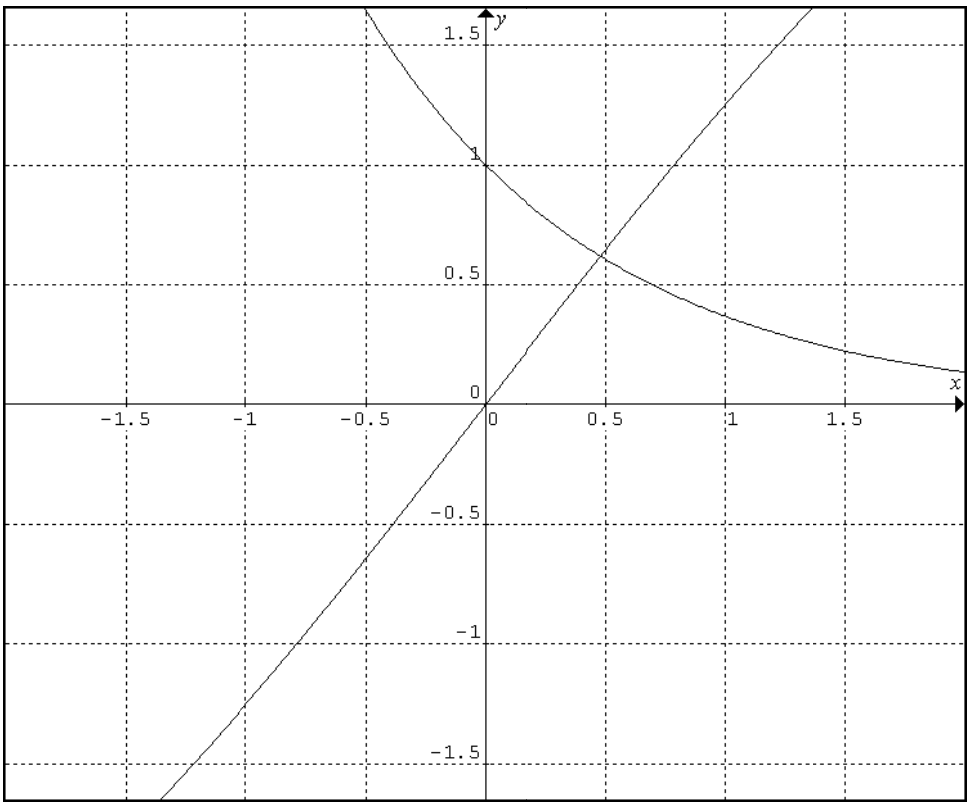
Pri navadni iteracijski metodi nelinearno enačbo f(x)=0 prevedemo v ekvivalentno obliko:

x=g(x)

Poiskati moramo tako vrednost α, da je α =g(α) in je f(α)=0.
Enačba x=g(x) pove, da je rešitev enačbe f(x)=0 tista vrednost x=α, pri kateri se funkciji y=x in y=g(x) sekata.

Algoritem navadne iteracijske metode:

1. postavimo indeks i=0, izberemo začetni približek x_0, izberemo dopustno relativno napako ε, določimo maksimalno število iteracij (ponovitev).
2. izračunamo nov približek x_{i+1} po formuli $x_{i+1}=g(x_i)$.
3. če je izvedemo določeno maksimalno število iteracij se program ustavi, računanje prekinemo. Pomeni, da x_{i+1} ni primeren približek rešitve enačbe.
4. če je $\left|\frac{x_{i+1}-x_i}{x_{i+1}}\right| > \varepsilon$, povečamo indeks i in se vrnemo na drugi korak. Sicer pomeni, da je x_{i+1},dober približek rešitve enačbe α. Ponavljanje zaključimo.



Graf prikazuje funkciji y=exp(-x) in y=x+0.3*sin(x). Presečišče je približno pri 0.48 .

Zaporedje približkov x_1, x_2, x_3 konverigira(se približuje) h korenu enačbe f(x)=0, če je blizu korena $|g'(x)| < 1$.

```
!  resujemo  enacbo  x=exp(-x)-0.3*sin(x)
G(X)=EXP(-X)-0.3*SIN(X)
WRITE(*,*)'Podaj   zacetni priblizek:'
READ*,X0
WRITE(*,*)'Podaj zahtevano natancnost:'
READ*,ES
WRITE(*,*)'Podaj maximalno stevilo iteracij:'
READ*,MAXIT
ITER=0
EP=1.1*ES
DO WHILE (EP.GT.ES.AND.ITER.LT.MAXIT)
WRITE(*,10)ITER,X0
ITER=ITER+1
X1=G(X0)
IF(X1.NE.0.0)THEN
EP=ABS((X1-X0)/X1)
ENDIF
X0=X1
ENDDO
WRITE(*,*)'Koren =',X0
WRITE(*,*)'Dosezena natancnost=',EP
WRITE(*,*)' Stevilo iteracij=',ITER
10 FORMAT(I5,F14.6)
END
```

Newtnova metoda

Pri newtnovi metodi izberemo začetni približek x_0, nadomestimo nelinearno funkcijo f(x) s tangento v točki $(x_0,f(x_0))$ in poiščemo kje tangenta seka os x. To je novi približek korena x_1. Postavimo tangento v točki $(x_1,f(x_1))$ in postopek nadaljujemo.

$$f'(x_0) = \tan\varphi = \frac{f(x_0)}{x_0 - x_1}$$

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

$$v \quad splošnem \quad zapišemo:$$

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

Algoritem Newtnove metode:

1. izračunamo f'(x).
2. postavimo indeks i=0, izberemo začetni približek $x_{0,}$ izberemo dopustno relativno napako ε, določimo maksimalno število iteracij (ponovitev).
3. izračunamo nov približek x_{i+1} po formuli $x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$
4. če je izvedemo predpisano maksimalno število iteracij se program ustavi, računanje prekinemo. Pomeni, da x_{i+1} ni primeren približek rešitve enačbe.
5. če je $\left|\frac{x_{i+1} - x_i}{x_{i+1}}\right| > \varepsilon$, povečamo indeks i in se vrnemo na drugi korak. Sicer pomeni, da je x_{i+1},dober približek rešitve enačbe α. Ponavljanje zaključimo.

```
FUNCTION F(X)
F=X-EXP(-X)+0.3*SIN(X)
RETURN
END

FUNCTION FD(X)
FD=1+EXP(-X)+0.3*COS(X)
RETURN
END

WRITE(*,*)'Podaj zacetni priblizek:'
READ(*,*)X0
WRITE(*,*)'Podaj zahtevano natancnost:'
READ(*,*)ES
WRITE(*,*)'Podaj maksimalno stevilo iteracij:'
READ(*,*)MAXIT
ITER=0
EP=1.1*ES

DO WHILE (EP.GT.ES.AND.ITER.LT.MAXIT)
WRITE (*,10)ITER,X0
ITER=ITER+1
X1=X0-F(X0)/FD(X0)
IF(X1.NE.0.0)THEN
EP=ABS((X1-X0)/X1)
ENDIF
X0=X1
ENDDO

WRITE(*,*)' Koren=',X0
WRITE(*,*)' Dosezena natancnost=',EP
WRITE(*,*)' Stevilo iteracij=',ITER
10 FORMAT(I5,F14.6)
END
```

Sekantna metoda

Kadar je računanje odvoda funkcije dolgotrajno, uporabimo sekantno metodo.

Za določitev ničle nelinearne funkcije f(x) najprej izberemo dva približka x_0 in x_1. Nadomestimo nelinearno funkcijo s sekanto skozi točki (x_0,f(x_0)) in (x_1,f(x_1)) ter izračunamo kje sekanta seka os x. To presečišče je novi približek x_2. Nato postavimo novo sekanto skozi točki (x_1,f(x_1)) in (x_2,f(x_2)) in nadaljujemo postopek.

$$\tan\varphi = \frac{f(x_0) - f(x_1)}{x_0 - x_1}$$

$$\tan\varphi = \frac{f(x_1)}{x_1 - x_2}$$

zgornji enčbi izenačimo in dobimo:

$$x_2 = x_1 - \frac{f(x_1)(x_0 - x_1)}{f(x_0) - f(x_1)}$$

Nadaljne približke izračunamo po formuli:

$$x_{i+1} = x_i - \frac{f(x_i)(x_{i-1} - x_i)}{f(x_{i-1}) - f(x_i)}$$

1. postavimo indeks i=0, izberemo začetna približka x_0,x_1 izberemo dopustno relativno napako ε, določimo maksimalno število iteracij (ponovitev).
2. izračunamo nov približek x_{i+1} po formuli $x_{i+1} = x_i - \frac{f(x_i)(x_{i-1} - x_i)}{f(x_{i-1}) - f(x_i)}$
3. če je izvedemo predpisano maksimalno število iteracij se program ustavi, računanje prekinemo. Pomeni, da x_{i+1} ni primeren približek rešitve enačbe.
4. če je $\left|\frac{x_{i+1} - x_i}{x_{i+1}}\right| > \varepsilon$, povečamo indeks i in se vrnemo na drugi korak. Sicer pomeni, da je x_{i+1},dober približek rešitve enačbe α. Ponavljanje zaključimo.

```
FUNCTION F(X)
F=X-EXP(-X)+0.3*SIN(X)
RETURN
END


WRITE(*,*)'Podaj zacetna priblizka:'
READ(*,*)X0,X1
WRITE(*,*)'Podaj zahtevano natancnost:'
READ(*,*)ES
WRITE(*,*)'Podaj maksimalno stevilo iteracij:'
READ(*,*)MAXIT
ITER=0
EP=1.1*ES

DO WHILE (EP.GT.ES.AND.ITER.LT.MAXIT)
WRITE (*,10)ITER,X0
ITER=ITER+1
X2=X1-F(X1)*(X0-X1)/(F(X0)-F(X1))
IF(X1.NE.0.0)THEN
EP=ABS((X1-X0)/X1)
ENDIF
X0=X1
X1=X2
ENDDO

WRITE(*,*)' Koren=',X0
WRITE(*,*)' Dosezena natancnost=',EP
WRITE(*,*)' Stevilo iteracij=',ITER
10 FORMAT(I5,F14.6)
END
```

Bisekcijska metoda

Pri bisekcijski metodi moramo najprej poiskati takšen interval (x_l,x_d), da funkcija f(x) na tem intervalu natanko enkrat zamenja predznak. Če velja $f(x_l)*f(x_d)<0$, je koren zaprt v interval $x_l < \alpha < x_d$.
Pri bisekcijski metodi interval, ki vsebuje koren, na vsakem koraku iteracije razpolovimo.
Pri nadaljnem računanju obdržimo tisto polovico intervala, v kateri ima funkcija na krajiščih nasprotni predznak. Z zmanjševanjem intervala se z mejama poljubno približamo korenu enačbe.

Algoritem bisekcijske metode

1. postavimo indeks i=1, izberemo začetni meji x_l in x_d intervala, da velja $f(x_l)*f(x_d)<0$, določimo dopustno relativno napako ε, določimo maksimalno število iteracij (ponovitev).
2. izračunamo razpolovišče $x_i = \frac{(x_l + x_d)}{2}$ in funkcijski vrednosti $f(x_l)$ in $f(x_i)$. Če je $f(x_i)=0$, je x_i koren α in zaključimo ponavljanje.
3. če imata funkcijski vrednosti $f(x_l)$ in $f(x_i)$ nasproten predznak je $f(x_l)*f(x_i)<0$ se koren (ničla) nahaja znotraj intervala (x_l,x_i) in postavimo $x_d=x_i$.
   V nasprotnem primeru se koren nahaja znotraj intervala (x_i,x_d) zato postavimo $x_l=x_i$. Če je i=1 povečamo indeks (števec) in se vrnemo na drugi korak.
4. če je izvedemo predpisano maksimalno število iteracij se program ustavi, računanje prekinemo. Pomeni, da x_i ni primeren približek rešitve enačbe.
5. če je $\left|\frac{x_i - x_{i-1}}{x_i}\right| > \varepsilon$, povečamo indeks i in se vrnemo na drugi korak. Sicer pomeni, da je x_i,dober približek rešitve enačbe α. Ponavljanje zaključimo.

Ker je $x_i - x_{i-1} = \frac{x_d - x_l}{2} \quad in \quad x_i = \frac{x_d + x_l}{2}$ sledi, da je ocena relativne napake korena $\varepsilon = \frac{x_d - x_l}{x_d + x_l}$.

```
FUNCTION f(x)
F=X-EXP(-X)+0.3*SIN(X)
RETURN
END

PRINT*,'podaj levo in desno mejo ter stevilo ponovitev:'
READ*,xl,xd,maxit
IF(f(xl)*f(xd)<0) THEN
DO i=1,maxit
x=(xl+xd)/2
IF(f(xl)*f(x)<0) xd=x
IF(f(xl)*f(x)>0) xl=x
IF(f(xl)*f(x)==0 GOTO 10
PRINT*,'x novi=',x,'korak=',i
IF(ABS((xd-xl)/(xd+xl)).LT.0.0005) STOP
ENDDO
ELSE
STOP'pogoj za bisekcijo ni izpolnjen'
ENDIF
STOP
10 PRINT*,'nicla funkcije=',x
END
```

Numerično integriranje

Če je f(x) taka funkcija, da je mogoče izračunati njen nedoločeni integral $\int f(x)dx = F(x) + C$, potem za določeni integral te funkcije na intervalu [a,b], na katerem je obravnavana
funkcija zvezna, velja obrazec:

$$\int_a^b f(x)dx = F(b) - F(a)$$

Ustreznega nedoločenega integrala se ne da vedno izračunati, ali pa je tako kompliciran, da je ta postopek preveč zamuden. V tem primeru ne računamo natančne vrednosti ampak izračunamo
približek. Postopek računanja približne vrednosti integrala imenujemo numerično integriranje.

Pravokotniška metoda

Interval [a,b] razdelimo na n enakih delov. Širina vsakega podintervala je (b-a)/n. Vrednost določenega integrala lahko aproksimiramo z vsoto ploščin pravokotnikov, ki imajo širino
(b-a)/n, višina pa je vsakokrat funkcijska vrednost v začetni ali končni točki podintervala.
Tako dobimo formulo:

Ploščine pravokotnikov v začetnih točkah podintervalov:

$$\int_a^b f(x)dx = \frac{b-a}{n}\left[f(x_0) + f(x_1) + f(x_2) + .. + f(x_{n-1})\right]$$

ali

Ploščine pravokotnikov v končnih točkah podintervalov:

$$\int_a^b f(x)dx = \frac{b-a}{n}\left[f(x_1) + f(x_2) + f(x_3) + .. + f(x_n)\right]$$

V obeh primerih je seveda približek tem boljši, čim večje je število podintervalov n.

```
FUNCTION F(X)
F=x*x*log(x)
RETURN
END

WRITE(*,*)'Podaj meji:'
READ (*, *) A, B
WRITE(*,*)'Podaj stevilo delitvenih tock:'
READ (*, *) N
H=(B-A)/(N-1)
RI=F(A)+F(B)
DO X=A+H,B-0.9*H,H
RI=RI+F(X)
ENDDO
RI=RI*H
WRITE(*,*)'INTEGRAL=',RI
END
```

Metoda trapezov

Pri tej metodi namesto ploščin pravokotnikov na posameznih podintervalih vzamemo ploščine trapezov, ki nastanejo, če loke krivulje f(x) v vsakem podintervalu nadomestimo z ustreznimi tetivami krivulje.

Približno vrednost integrala lahko izrazimo z vsoto:

$$\int_a^b f(x)dx = \frac{b-a}{n}\left[\frac{f(x_0)+f(x_1)}{2}+\frac{f(x_1)+f(x_2)}{2}+..+\frac{f(x_{n-1})+f(x_n)}{2}\right]=$$

$$=\frac{b-a}{2n}\left[f(x_0)+2\left(f(x_1)+f(x_2)+..+f(x_{n-1})\right)+f(x_n)\right]$$

Ta metoda daje boljše približke od metode pravokotnikov, seveda je tudi pri tej metodi stopnja natančnosti odvisna od števila podintervalov n.

```
FUNCTION F(X)
F=x*x*log(x)
RETURN
END

WRITE(*,*)' Podaj meji:'
READ (*, *) A, B
WRITE(*,*)' Podaj stevilo delitvenih tock:'
READ (*, *) N
H=(B-A)/(N-1)
TI=F(A)+F(B)
DO X=A+H,B-0.9*H,H
TI=TI+2*F(X)
ENDDO
TI=TI*H/2
WRITE(*,*)' INTEGRAL=',TI
END
```

Simpsonov obrazec

Naj bodo v ravnini dane tri točke tako, da je abscisa srednje točke enaka aritmetični sredini abscis ostalih dveh točk. Njihove koordinate naj bodo:

$T_1(0,y_1)$
$T_2(h/2,y_2)$
$T_3(h,y_3)$

Skozi tri točke poteka enolično določena parabola druge stopnje, ki jo zapišemo z enačbo $y=ax^2+bx+c$. V enačbo $y=ax^2+bx+c$ vstavimo koordinate točk T_1, T_2 in T_3 ter dobimo parametre a,b in c. Parametri a,b in c so določeni z naslednjim sistemom enačb:

$Y_1=c$
$Y_2=ah^2/4+bh/2+c$
$Y_3=ah^2+bh+c$

Ploščina lika, ki ga oklepa parabola z abscisno osjo med ordinatama v točkah x=0 in x=h je dana z določenim integralom:

$$p=\int_0^h (ax^2+bx+c)\,dx=\left[a\frac{x^3}{3}+b\frac{x^2}{2}+cx\right]_0^h=$$

$$p=a\frac{h^3}{3}+b\frac{h^2}{2}+ch=\frac{h}{6}(2ah^2+3bh+6c)$$

Če sedaj enačbo $Y_2=ah^2/4+bh/2+c$ pomnožimo s 4 dobimo:

$4Y_2=ah^2+2bh+4c$

in seštejemo $Y_1+4Y_2+Y_3=c+ah^2+2bh+4c+ah^2+bh+c=2ah^2+3bh+6c$.

$$p=\frac{h}{6}(y_1+4y_2+y_3)$$

Ploščino omenjenega lika izračunamo torej naravnost iz koordinat danih točk, ne da bi morali najprej določiti parametre ustrezne parabole druge stopnje.

V tistih točkah na abscisni osi, ki imajo sode indekse, imamo postavljene ordinate. Lik med krivuljo f(x) in abscisno osjo ter ordinatama v točkah a in b razdelimo na n pasov širine
$h=\frac{b-a}{n}$, kjer je vedno po ena točka z lihim indeksom ravno razpolovišče ustreznega podintervala.

$$p = \int_a^b f(x)dx \approx \frac{h}{3}\left(y_0 + 4y_1 + 2y_2 + 4y_3 + .. + 2y_{n-2} + 4y_{n-1} + y_n\right)$$

Zgornjo enačbo za izračun približne vrednosti določenega integrala imenujemo Simpsonovo formulo. Ta metoda, da boljše približke kot metoda pravokotnikov ali trapezna metoda. Število delitvenih točk pri simpsonovi metodi mora biti liho. Približek računanja je tem boljši čim večji je n.

```
FUNCTION F(X)
F=X*X*ALOG(X)
RETURN
END

WRITE(*,*)' Podaj meji:'
READ(*,*)A,B
WRITE(*,*)' Podaj stevilo delitvenih tock:'
READ (*,*)N
IF(MOD(N,2).EQ.0) STOP 'Stev. del. tock mora biti liho'
SI=F(A)+F(B)
H=(B-A)/(N-1)
DO X=A+H, B-0.9*H,2*H
SI=SI+4*F(X)
ENDDO
DO X=A+2*H, B-1.9*H,2*H
SI=SI+2*F(X)
ENDDO
SI=SI*H/3
WRITE(*,*)'INTEGRAL=',SI
END
```