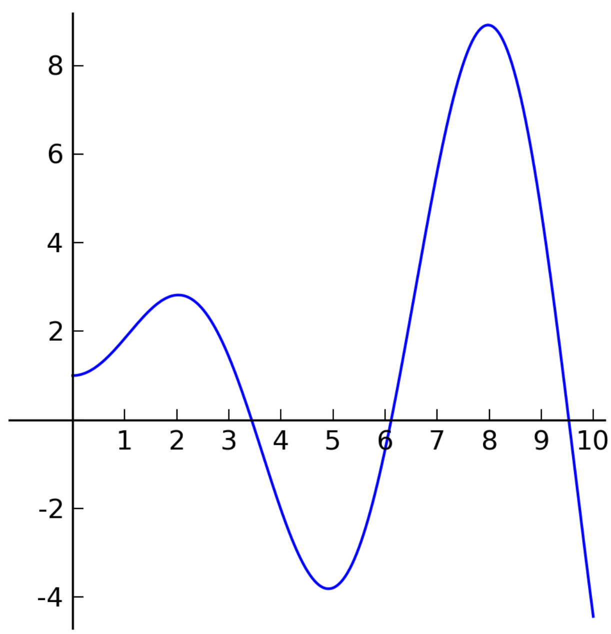
Približno traženje nultočki

– kratke informacije –

Općenito je nemoguće pronaći egzaktno nultočke neke neprekidne funkcije. Stoga se u praksi koriste razne iterativne metode koje numerički pronalaze nultočke, tj. pronalaze nultočke s točnošću na određeni broj decimala. Postoji mnogo metoda koje numerički pronalaze nultočke nelinearnih funkcija na zadanu točnost. Te metode se razlikuju po brzini pronalaženja nultočke, ali isto tako i po garanciji da li će uspjeti pronaći nultočku. Uobičajeni slučaj je da brže metode nemaju sigurnu garanciju da će uspjeti pronaći nultočku, dok sporije metode imaju tu garanciju da će sigurno pronaći nultočku.

Ovdje nećemo ulaziti u strogu matematičku teoriju svih tih metoda, nego je ideja da samo ukratko objasnimo naredbu `find_root` u `maximi` pomoću koje u velikoj većini slučajeva možemo približno pronaći nultočku funkcije unutar nekog segmenta. Ova metoda zahtijeva da se zada segment $[a, b]$ za kojeg smo sigurni da unutar njega neprekidna funkcija ima bar jednu nultočku. Tu sigurnost osiguravamo zahtjevom $f(a) \cdot f(b) < 0$, tj. segment mora biti zadan tako da u njegovim rubnim točkama funkcija poprima vrijednosti različite po predznaku.

Na primjer, na donjoj slici vidimo da funkcija $f(x) = x \sin x + 1$ ima tri nultočke na segmentu $[0, 10]$.



Slika 1: Funkcija $f(x) = x \sin x + 1$

Stoga, ako u `maximi` napišemo

```
find_root(x*sin(x)+1,x,0,10);
```

`maxima` će izbaciti rezultat

```
3.436828912326677.
```

Maxima je dala samo jednu od nultočki na segmentu $[0, 10]$. Želimo li dobiti određenu nultočku, treba navesti neki segment oko te nultočke koji sadrži samo tu nultočku. Na primjer, sa slike uočavamo da funkcija f ima jednu nultočku na segmentu $[9, 10]$. Želimo li dobiti tu nultočku, trebamo napisati

```
find_root(x*sin(x)+1,x,9,10);
```

U tom slučaju maxima daje rezultat

```
9.529904336449029.
```

Isto tako, želimo li dobiti približnu nultočku koja je blizu broja 6, napišemo

```
find_root(x*sin(x)+1,x,6,7);
```

pa dobivamo rezultat

```
6.119024225042324.
```

Naravno, mogli smo napisati i neki drugi segment oko te nultočke, npr.

```
find_root(x*sin(x)+1,x,5.5,6.5);
```

i ponovo bismo dobili

```
6.119024225042324.
```

Ukoliko želimo raditi u većoj preciznosti, možemo koristiti naredbu `bf_find_root` i varijablu `fpprec` za broj decimala. Na primjer, ako želimo pronaći nultočku funkcije f na segmentu $[6, 7]$ s preciznošću na 50 znamenki, tada napišemo

```
bf_find_root(x*sin(x)+1,x,6,7), fpprec:50;
```

pa maxima vraća vrijednost

```
6.1190242250423236286846582759149564174857737088648b0
```

Pritom `b0` označava 10^0. Na primjer, `2.3b-14` označava broj $2.3 \cdot 10^{-14}$.

Na segmentu $[3, 7]$ funkcija f ima dvije nultočke. Međutim, u ovom slučaju nije dobro reći naredbi `find_root` da traži nultočku na segmentu $[3, 7]$. Zašto? Naime, $f(3)$ i $f(7)$ su istog predznaka pa nije zadovoljen uvjet $f(3) \cdot f(7) < 0$ koji nam osigurava sigurno postojanje nultočke na segmentu. Dakle, ako u maximi napišemo

```
find_root(x*sin(x)+1,x,3,7);
```

maxima javlja poruku

```
find_root: function has same sign at endpoints: mequal(f(3.0),1.423360024179602),
mequal(f(7.0),5.598906191031523)
 -- an error. To debug this try: debugmode(true);
```

da funkcija f u rubovima intervala ima vrijednosti s istim predznacima.

Možemo najprije u maximi definirati funkciju f pa tek onda koristiti naredbu `find_root`. Drugim riječima, umjesto da npr. napišemo

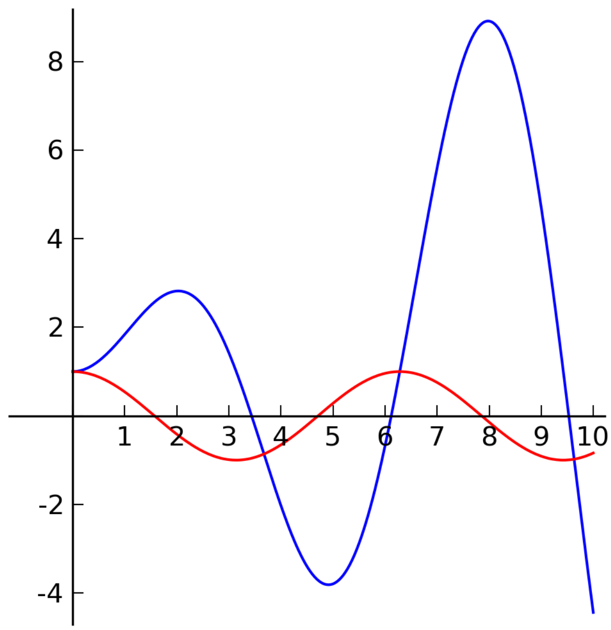
```
find_root(x*sin(x)+1,x,9,10);
```

možemo napisati

```
f(x):=x*sin(x)+1;
find_root(f,9,10);
```

što je korisno ukoliko funkciju f puno puta koristimo.

Pomoću naredbe `find_root` možemo približno pronaći točke presjeka grafova dviju funkcija f i g tako da nađemo približna rješenja jednadžbe $f(x) = g(x)$. Pogledajmo primjer funkcija $f(x) = x \sin x + 1$ i $g(x) = \cos x$. Ukoliko promatramo te funkcije na segmentu $[0, 10]$, na donjoj slici vidimo da se njihovi grafovi sijeku u četiri točke. Jedna od tih točaka je $(0, 1)$ koja se lako očita sa slike jer je očito $x = 0$ jedno rješenje jednadžbe $x \sin x + 1 = \cos x$. Ostala rješenja te jednadžbe na segmentu $[0, 10]$ nije moguće egzaktno odrediti, ali se mogu odrediti približno pomoću naredbe `find_root`. Gledajući sliku, vidimo da je jedan od presjeka na segmentu $[3, 4]$. Taj presjek približno



Slika 2: Funkcije $f(x) = x \sin x + 1$ i $g(x) = \cos x$

možemo pronaći na sljedeći način:

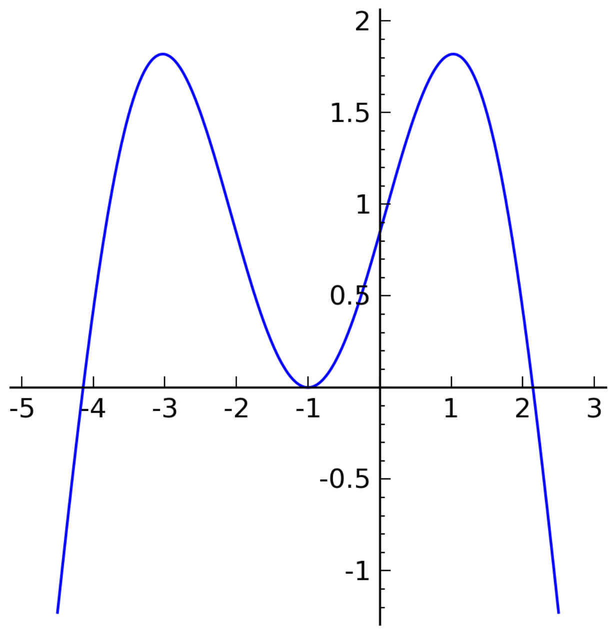
```
f(x):=x*sin(x)+1;
g(x):=cos(x);
find_root(f(x)=g(x),x,3,4);
```

Na taj način dobivamo približnu x-koordinatu presjeka. Želimo li i y-koordinatu, tada najprije približnu x-koordinatu spremimo u varijablu i izračunamo vrijednost funkcije f ili g u toj spremljenoj vrijednosti. Na taj način dobivamo približnu y-koordinatu presječne točke.

```
x1:find_root(f(x)=g(x),x,3,4);
y1:f(x1);
```

Na taj način dobivamo da se grafovi funkcija f i g na segmentu $[3, 4]$ sijeku u točki s koordinatama $(3.673194406304252, -0.86199622464574)$. Na analogni način možemo odrediti sve preostale presjeke grafova funkcija f i g na segmentu $[0, 10]$.

Objasnimo još ukratko u kojem slučaju naredba `find_root` neće uspjeti pronaći nultočku, iako nultočka možda postoji. Intuitivno rečeno, to će se dogoditi kada graf funkcije "dira" x-os na način kako je to prikazano na slici 3. na kojoj je prikazan graf funkcije $f(x) = (x+1)\sin(x+1)$. Sve nultočke ove funkcije možemo čak i egzaktno odrediti rješavanjem jednadžbe $(x+1)\sin(x+1) = 0$. Isto tako bismo mogli nultočke te funkcije odrediti približno pomoću naredbe `find_root`, osim nultočke $x_0 = -1$. U toj točki graf funkcije dira x-os. Još nije toliki problem u tom "diranju", već je problem što ne postoji segment na kojemu bi -1 bila jedina nultočka funkcije f tako da funkcija f u rubnim točkama tog segmenta poprima vrijednosti različite po predznaku. Svaki segment $[a, b]$ na kojemu je -1 jedina nultočka ne zadovoljava uvjet $f(a) \cdot f(b) < 0$. Ukoliko pak neki segment $[a, b]$ koji sadrži nultočku -1 zadovoljava uvjet $f(a) \cdot f(b) < 0$, tada -1 nije jedina nultočka na tom segmentu pa ukoliko i primijenimo naredbu `find_root` za taj segment, ona neće vratiti približno rješenje za nultočku -1, već će vratiti neku drugu nultočku na tom segmentu.



Slika 3: Funkcija $f(x) = (x+1)\sin(x+1)$

U ovom slučaju, ako ipak želimo pronaći nultočku -1, možemo u `maximi` koristiti naredbu `newton` koja ne zahtijeva da se na početku navede segment, nego samo početna točka koja je dovoljno blizu nultočke koju želimo pronaći. Dakle, ako napišemo

```
f(x):=(x+1)*sin(x+1);
newton(f(x),x,0,10^(-15));
```

tada tražimo nultočku funkcije f počevši od točke 0 i tražimo tako dugo dok vrijednost funkcije nije manja od 10^{-15}. `Maxima` u ovom slučaju vraća vrijednost

```
-0.99999997749231
```

kao aproksimaciju nultočke -1. Pogledajte u `wxmaxima` datoteci još neke komentare za ovaj primjer.