# Konvexita a konkávnost funkcí

Viliam Holub

## O co jde, definice

Konvexita  a konkávnost jsou vlastnosti funkce, kterou zkoumá matematická analýza.

Nějdříve se podíváme na definici:

Definice:
Řekneme, že funkce je konvexní na intervalu I, jestliže pro všechna  a  z intervalu <0, 1> platí

$$f(\lambda\, x_1 + (1-\lambda)\, x_2) \le \lambda\, f(x_1) + (1-\lambda)\, f(x_2)$$

Řekneme, že funkce je konkávní na intervalu I, jestliže pro všechna  a  z intervalu <0, 1> platí

$$\lambda\, f(x_1) + (1-\lambda)\, f(x_2) \le f(\lambda\, x_1 + (1-\lambda)\, x_2)$$

Pokud pro funkci na daném intervalu platí některá z výše uvedených definic s ostrou nerovností, říkáme, že funkce je ryze konvexní resp. konkávní.

Pokud je tato definice pro někoho příliš temná, snad mu pomůžou následující příklady, nebo ústní vysvětlení:

Funkce je na intervalu I konvení, pokud platí, že vybereme-li libovolné dva body $x_1, x_2$ z tohoto intervalu, bez újmy na obecnosti nechť $x_1 \le x_2$ a proložíme-li na grafu funkce body $f(x_1)$ a $f(x_2)$ přímkou, pak hodnoty funkce na intervalu $\langle x_1, x_2 \rangle$ leží pod touto přímkou. Obdobně funkce je na intervalu I konkávní, pokud hodnoty funkce leží nad přímkou.

Také se říká "Do konkávy kávu nenaleješ."

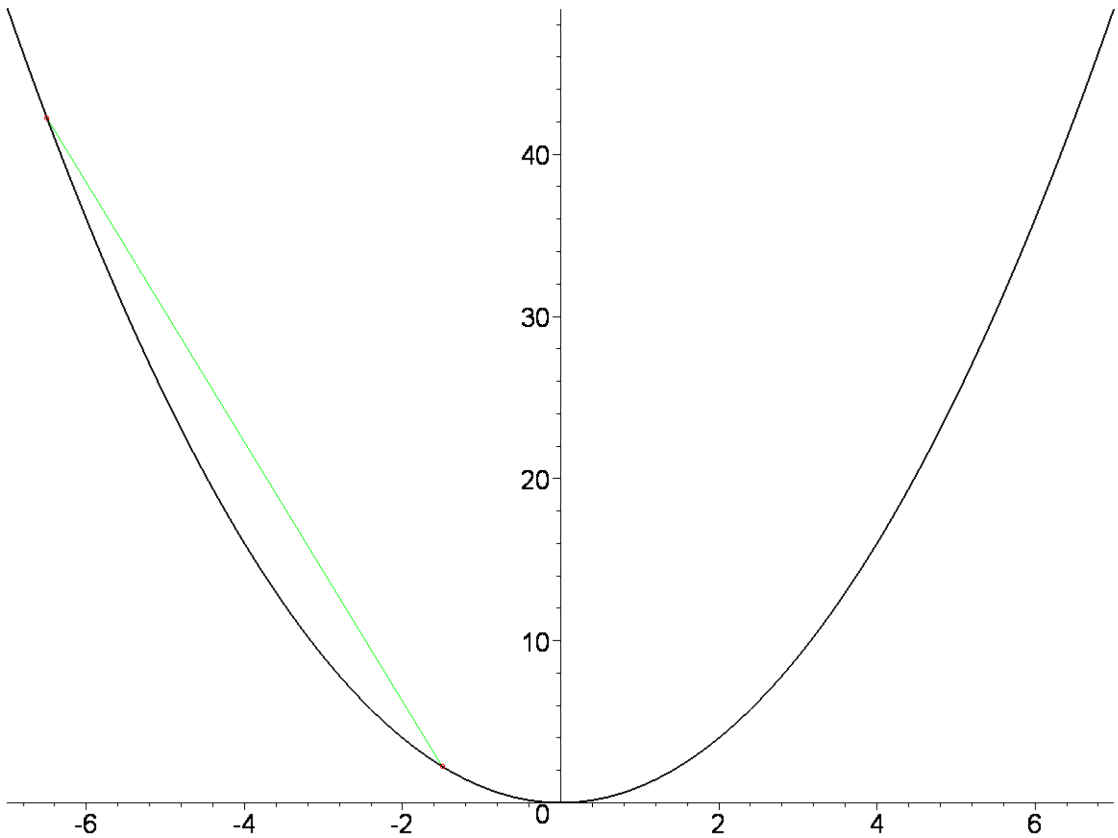
## Některé jednoduché příklady

Konvexita a konkávnost si lze nejlépe ukázat na grafu:

```
> f := x->x^2:
> pf := CURVES( [seq( [ i/10, evalf( f( i/10))], i=-70..70)],
  THICKNESS( 2)):
> dvabody := flnum -> POINTS( [evalf( flnum/10), evalf( f(
```

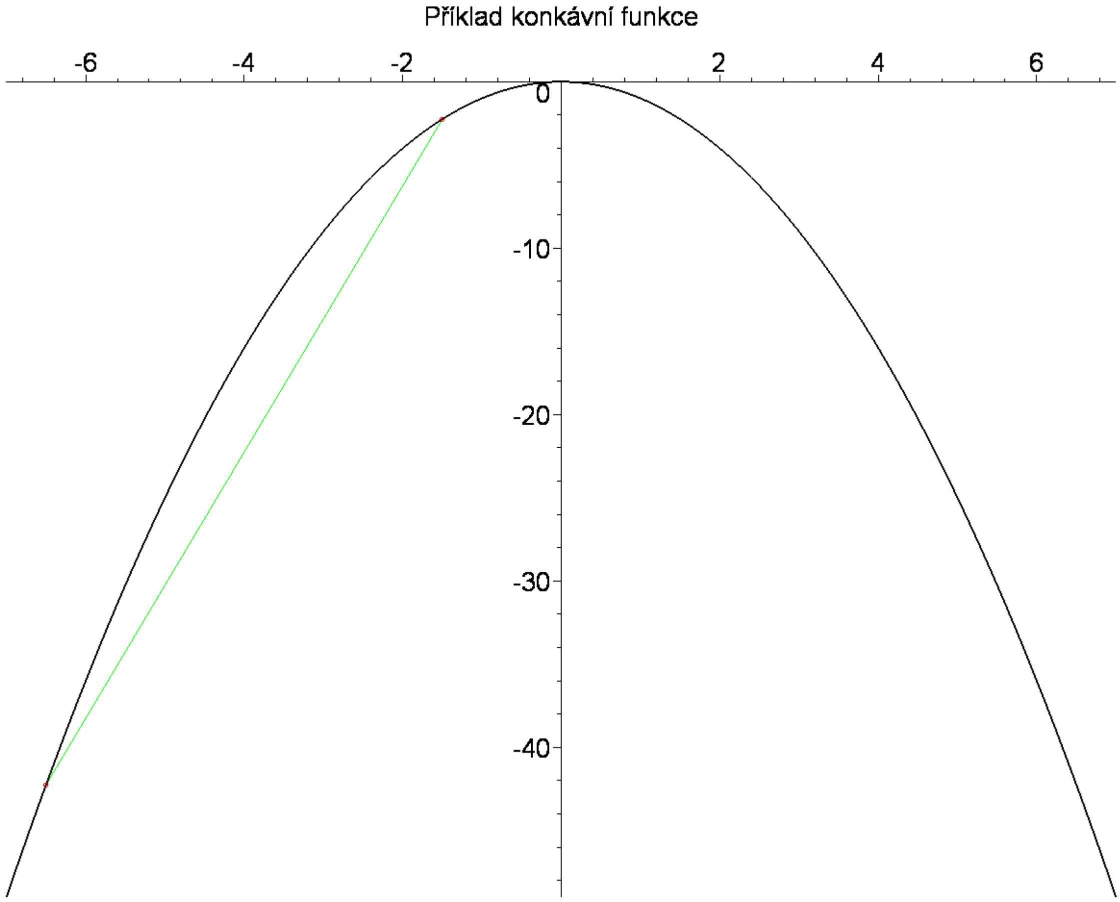
```
flnum/10))], [evalf( flnum/10) +5, evalf( f( flnum/10+5))],
COLOUR( RGB, 1, 0, 0), SYMBOL( CIRCLE)):
> cara := flnum -> CURVES( [[evalf( flnum/10), evalf( f(
flnum/10))], [evalf( flnum/10 +5), evalf( f( flnum/10+5))]],
COLOUR( RGB, 0, 1, 0), SYMBOL( CIRCLE)):
> gfce := flnum -> CURVES( [seq( [ i/10, evalf( f( i/10))],
i=flnum..flnum+50)], THICKNESS( 2), COLOR( RGB, 0, 0, 1)):
> p1 := flnum -> [pf, dvabody( flnum), cara( flnum), gfce(
flnum)]:
> nadpis := `Příklad konvexní funkce`:
> PLOT( ANIMATE( seq( p1( t), t=-65..15)), TITLE( nadpis));
```

Příklad konvexní funkce



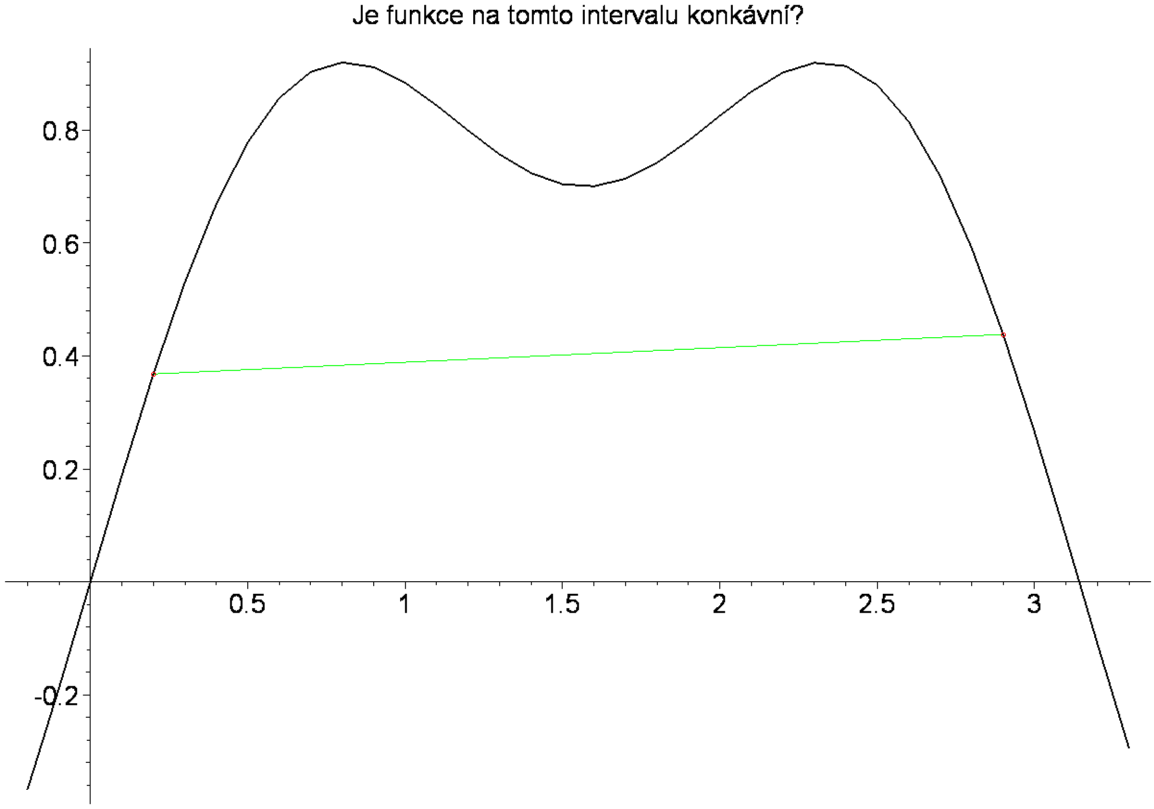

```
> f := x->-x^2:
> pf := CURVES( [seq( [ i/10, evalf( f( i/10))], i=-70..70)],
THICKNESS( 2)):
> dvabody := flnum -> POINTS( [evalf( flnum/10), evalf( f(
flnum/10))], [evalf( flnum/10) +5, evalf( f( flnum/10+5))],
COLOUR( RGB, 1, 0, 0), SYMBOL( CIRCLE)):
> cara := flnum -> CURVES( [[evalf( flnum/10), evalf( f(
flnum/10))], [evalf( flnum/10 +5), evalf( f( flnum/10+5))]],
COLOUR( RGB, 0, 1, 0), SYMBOL( CIRCLE)):
> gfce := flnum -> CURVES( [seq( [ i/10, evalf( f( i/10))],
i=flnum..flnum+50)], THICKNESS( 2), COLOR( RGB, 0, 0, 1)):
> p1 := flnum -> [pf, dvabody( flnum), cara( flnum), gfce(
flnum)]:
> nadpis := `Příklad konkávní funkce`:
```

```
> PLOT( ANIMATE( seq( p1( t), t=-65..15)), TITLE( nadpis));
```




Avšak pozor:

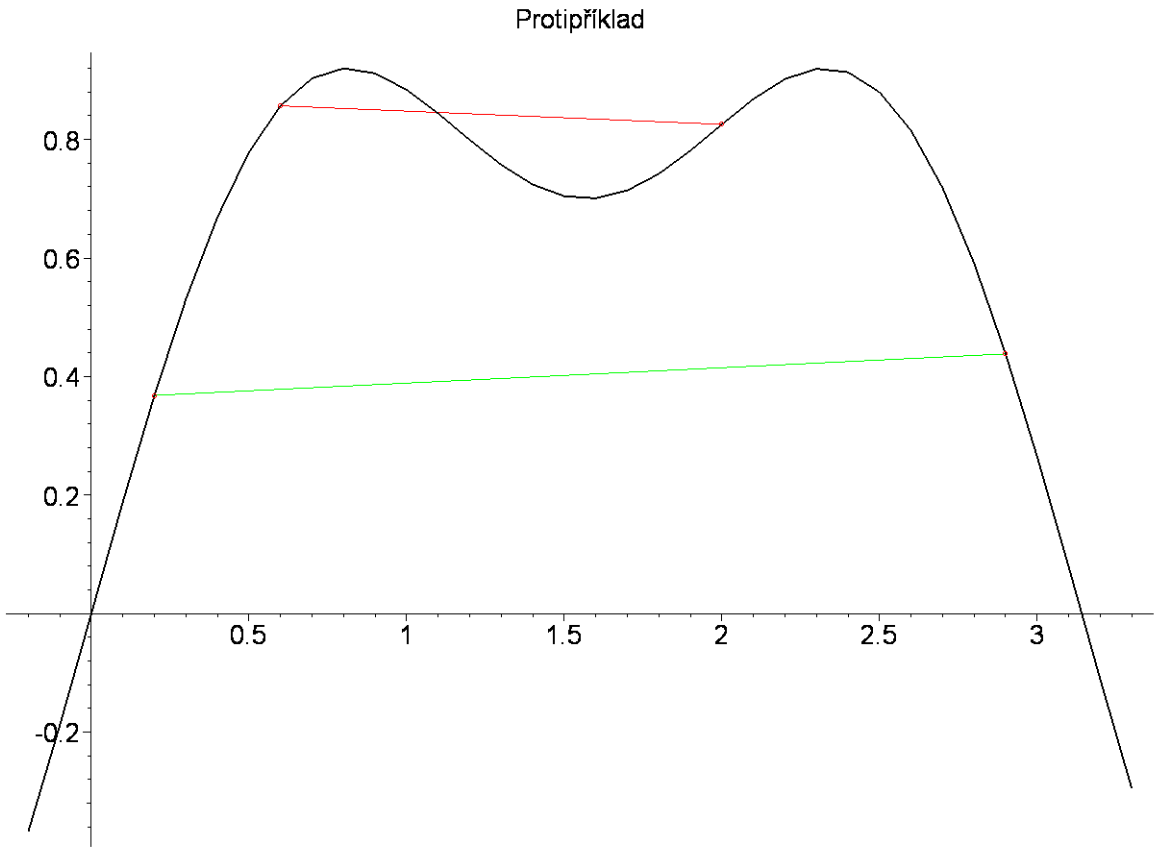
```
> f := x->sin( x)+0.3*sin(3*x):
> p1 := CURVES( [seq( [ i/10, evalf( f( i/10))], i=-2..33)],
  THICKNESS( 2)), POINTS( [0.2, evalf( f( 0.2))], [ 2.9, evalf(
  f( 2.9))], COLOUR( RGB, 1, 0, 0), SYMBOL( CIRCLE)), CURVES(
  [[evalf( 0.2), evalf( f( 0.2))], [evalf( 2.9), evalf( f(
  2.9))]], COLOUR( RGB, 0, 1, 0), SYMBOL( CIRCLE)), CURVES(
  [seq( [ i/10, evalf( f( i/10))], i=2..29)], THICKNESS( 2),
  COLOR( RGB, 0, 0, 1)):
> PLOT( p1, TITLE( `Je funkce na tomto intervalu konkávní?`));
```




Neni. Ačkoli jsou všechny hodnoty funkce v tomto intervalu nad úsečkou, nesplňuje definici, protože ta hovoří o všech dvojicích bodů v tomto intervalu.

Snadno se podíváme, že vhodným zvolením dvou bodů dotaneme funkci pod úsečku:

```
> f := x->sin( x)+0.3*sin(3*x):
> p1 := CURVES( [seq( [ i/10, evalf( f( i/10))], i=-2..33)],
  THICKNESS( 2)), POINTS( [0.2, evalf( f( 0.2))], [ 2.9, evalf(
  f( 2.9))], COLOUR( RGB, 1, 0, 0), SYMBOL( CIRCLE)), POINTS(
  [0.6, evalf( f( 0.6))], [ 2, evalf( f( 2))], COLOUR( RGB, 1,
  0, 0), SYMBOL( CIRCLE)), CURVES( [[evalf( 0.2), evalf( f(
  0.2))], [evalf( 2.9), evalf( f( 2.9))]], COLOUR( RGB, 0, 1,
  0)), CURVES( [[evalf( 0.6), evalf( f( 0.6))], [evalf( 2),
  evalf( f( 2))]], COLOUR( RGB, 1, 0, 0)), CURVES( [seq( [
  i/10, evalf( f( i/10))], i=2..29)], THICKNESS( 2), COLOR(
  RGB, 0, 0, 1)):
> PLOT( p1, TITLE( `Protipříklad`));
```




## Vztah konvexnosti, konkávnosti a druhé derivace

Podíváme se na větu z matematické analýzy:

Věta: Nechť $f$ je funkce spojitá na intervalu $I$ (libovolného druhu) a nechť f´´( x) > 0 (resp. f´´( x)  < 0) v každém vnitřním bodě intervalu . Pak $f$ je ryze konvexní (resp. ryze konkávní) na intervalu $I$.

Což nám hodně pomáhá vyšetřujeme-li funkce.

## Některé zajímavější příklady

Podíváme se na funkci sin( x), která je všem důverně známá. Sama o sobě nám poslouží jako pěkná ukázka spojité periodické funkce, která střídá konvexitu a konkavitu.

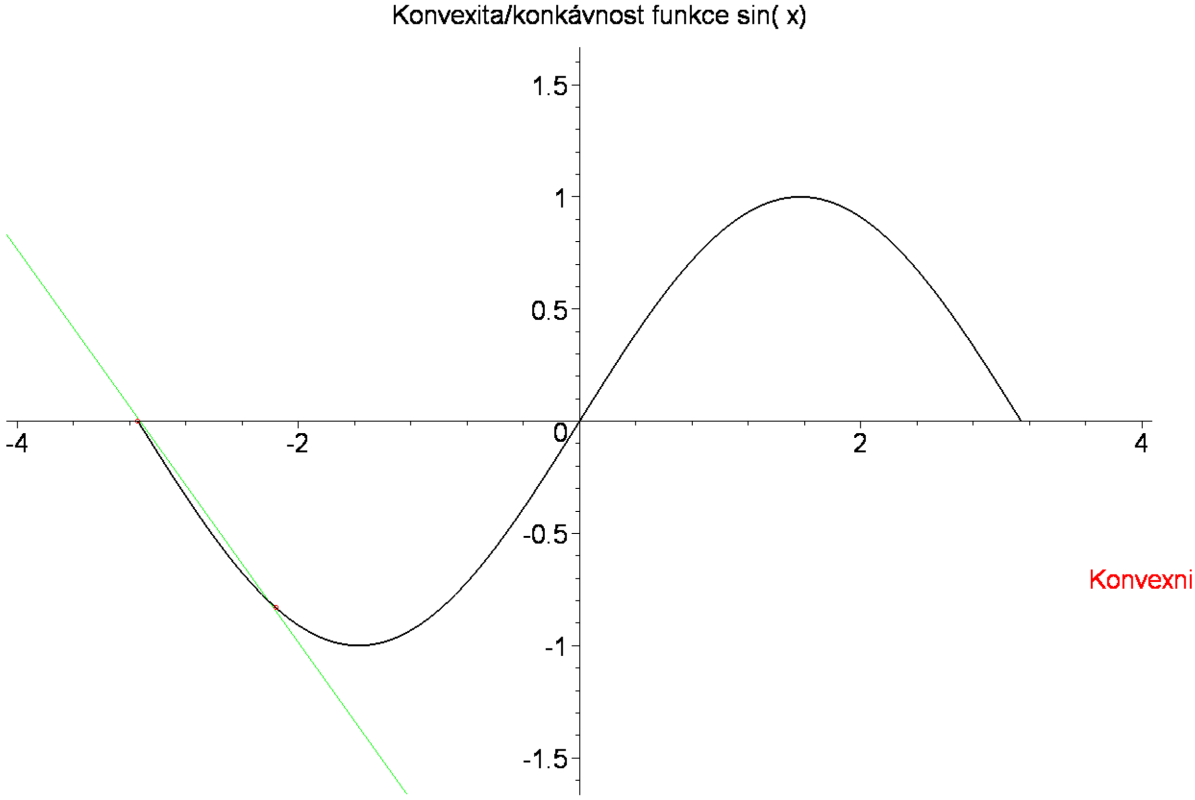
Podíváme se na graf:

```
> f := x -> sin( x*Pi/64);
```

$$f := x \rightarrow \sin\left(\frac{1}{64}\,x\,\pi\right)$$

```
> psin := CURVES( [seq( [ evalf( i*Pi/64), evalf( f( i))], 
  i=-64..64)], THICKNESS( 2)):
> dvabody := flnum -> POINTS( [evalf( flnum*Pi/64), evalf( f(
```

```
flnum))], [evalf( (flnum +20)*Pi/64), evalf( f( flnum+20))],
COLOUR( RGB, 1, 0, 0), SYMBOL( CIRCLE)):
> cara := flnum -> CURVES( [[evalf( flnum*Pi/64 -( (19*Pi/64)
)), evalf( f( flnum) -(f( flnum+20)-f( flnum)))], [evalf(
(flnum +20)*Pi/64 +( (19*Pi/64) )), evalf( f( flnum+20) +(f(
flnum+20) -f( flnum)))]], COLOUR( RGB, 0, 1, 0), SYMBOL(
CIRCLE)):
> gfce := flnum -> CURVES( [seq( [evalf( i*Pi/64), evalf( f(
i))], i=flnum..flnum+20)], THICKNESS( 2), COLOR( RGB, 0, 0,
1)):
> konkavni := TEXT( [4, -0.5], `Konkavni`):
> konvexni := TEXT( [4, -0.7], `Konvexni`):
> p1 := flnum -> [psin, dvabody( flnum), cara( flnum), gfce(
flnum), konvexni]:
> p2 := flnum -> [psin, dvabody( flnum), cara( flnum), gfce(
flnum), konkavni]:
> nadpis := `Konvexita/konkávnost funkce sin( x)`:
> PLOT( ANIMATE( (seq( p1( t), t=-64..-10), seq( p2( t),
t=-10..44))), TITLE( nadpis));
```




A jako poslední si ukážeme ještě jinou funkci:

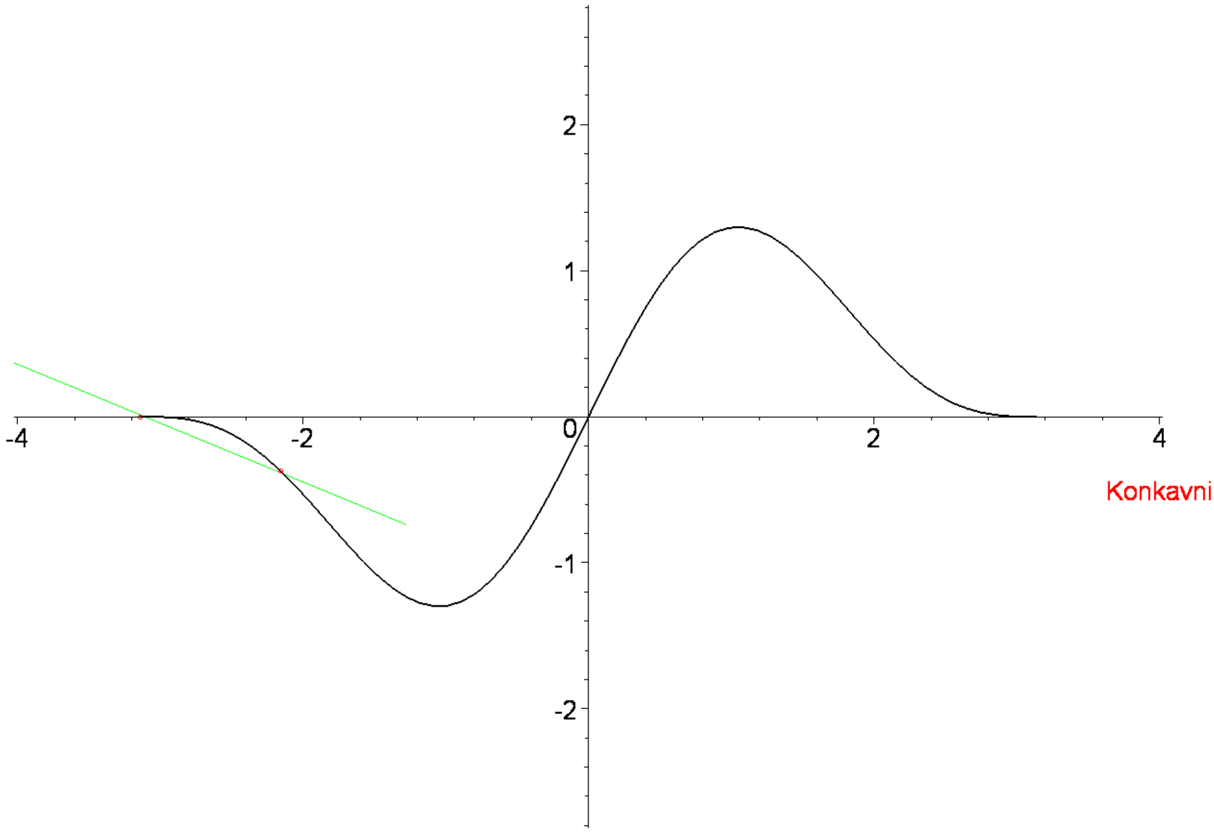
```
> f := x -> sin( x) +0.5*sin(x*2);
```

$$f := x \rightarrow \sin(x) + 0.5 \sin(2\,x)$$

```
> psin := CURVES( [seq( [ evalf( i*Pi/32), evalf( f(
i*Pi/32))], i=-32..32)], THICKNESS( 2)):
> dvabody := flnum -> POINTS( [evalf( flnum*Pi/32), evalf( f(
```

```
flnum*Pi/32))], [evalf( (flnum +10)*Pi/32), evalf( f(
(flnum+10)*Pi/32))], COLOUR( RGB, 1, 0, 0), SYMBOL( CIRCLE)):
> cara := flnum -> CURVES( [[evalf( flnum*Pi/32 -( (9*Pi/32)
)), evalf( f( flnum*Pi/32) -(f( (flnum+10)*Pi/32)-f(
flnum*Pi/32)))], [evalf( (flnum +10)*Pi/32 +( (9*Pi/32) )),
evalf( f( (flnum+10)*Pi/32) +(f( (flnum+10)*Pi/32) -f(
flnum*Pi/32)))]], COLOUR( RGB, 0, 1, 0), SYMBOL( CIRCLE)):
> gfce := flnum -> CURVES( [seq( [evalf( i*Pi/32), evalf( f(
i*Pi/32))], i=flnum..flnum+10)], THICKNESS( 2), COLOR( RGB,
0, 0, 1)):
> konvexni := TEXT( [4, -0.7], `Konvexni`):
> konkavni := TEXT( [4, -0.5], `Konkavni`):
> p1 := flnum -> [psin, dvabody( flnum), cara( flnum), gfce(
flnum), konkavni]:
> p2 := flnum -> [psin, dvabody( flnum), cara( flnum), gfce(
flnum), konvexni]:
> PLOT( ANIMATE( (seq( p1( t), t=-32..-24), seq( p2( t),
t=-24..-5), seq( p1( t), t=-5..14), seq( p2( t), t=14..22))),
TITLE(`Konvexita/konkávita funkce`));
```

Konvexita/konkávita funkce




```
>
```