Riemannův integrál

Riemannův integrál

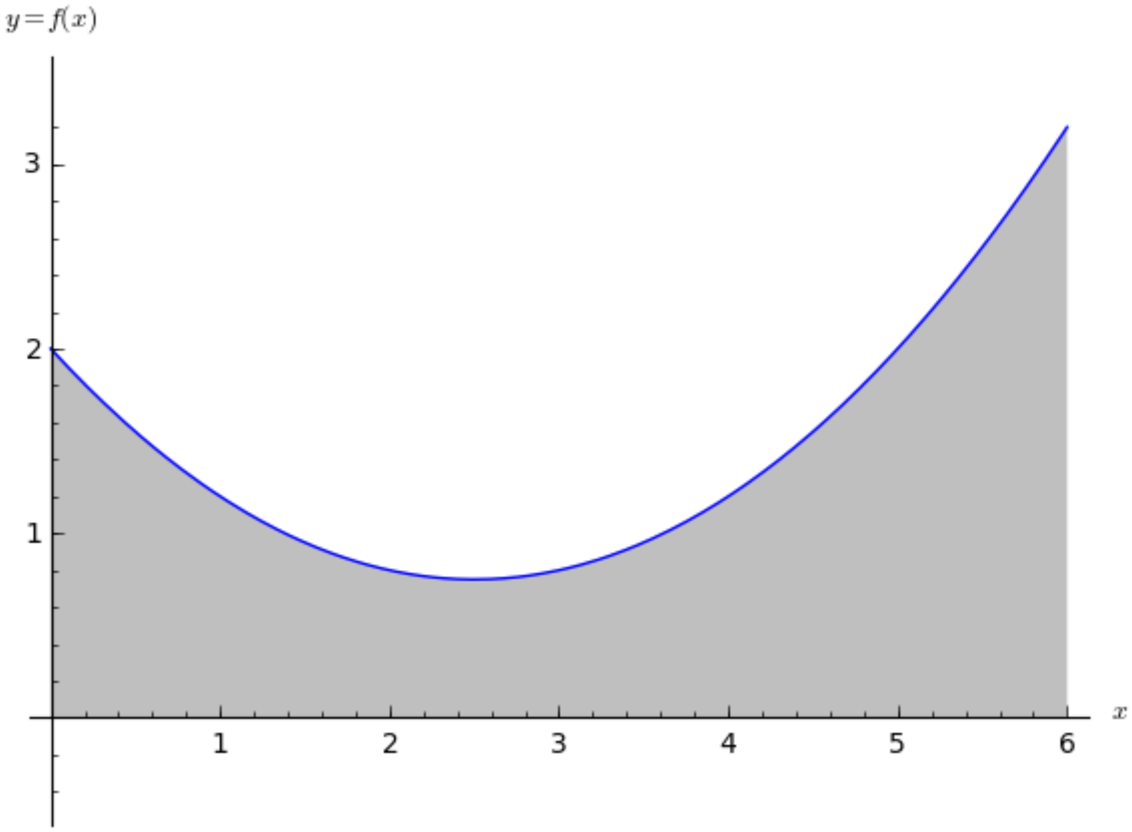
Tomáš Kasalický, 2014

V tomto notebooku si ukážeme numerickou integraci na příkladu funkce $f(x) = 2 + 1/5x^2 - x$ na intervalu $\langle 0, 6\rangle$. Zvídavý čtenář může snadno experimentovat s jinými volbami.

```
f(x)=2+1/5*x^2-x
a=0
b=6
```

Graf této funkce je uveden na následujícím obrázku. Riemannův (určitý) integrál funkce f nad intervalem $\langle a, b\rangle$ pak představuje obsah plochy pod grafem funkce f.

```
graph = Graphics()
graph_params = dict(ymin = -.5, ymax = 3.5)
graph += plot( f(x) , (x, 0, 6), color="blue", axes_labels=
['$x$','$y=f(x)$'], fill="axis")
show (graph, figsize=6,**graph_params)
```



Neboť se na uvažovaném intervalu jedná o kladnou funkci, je plocha pod grafem funkce dána přímo integrálem

$$\int_a^b f(x)\,\mathrm{d}x.$$

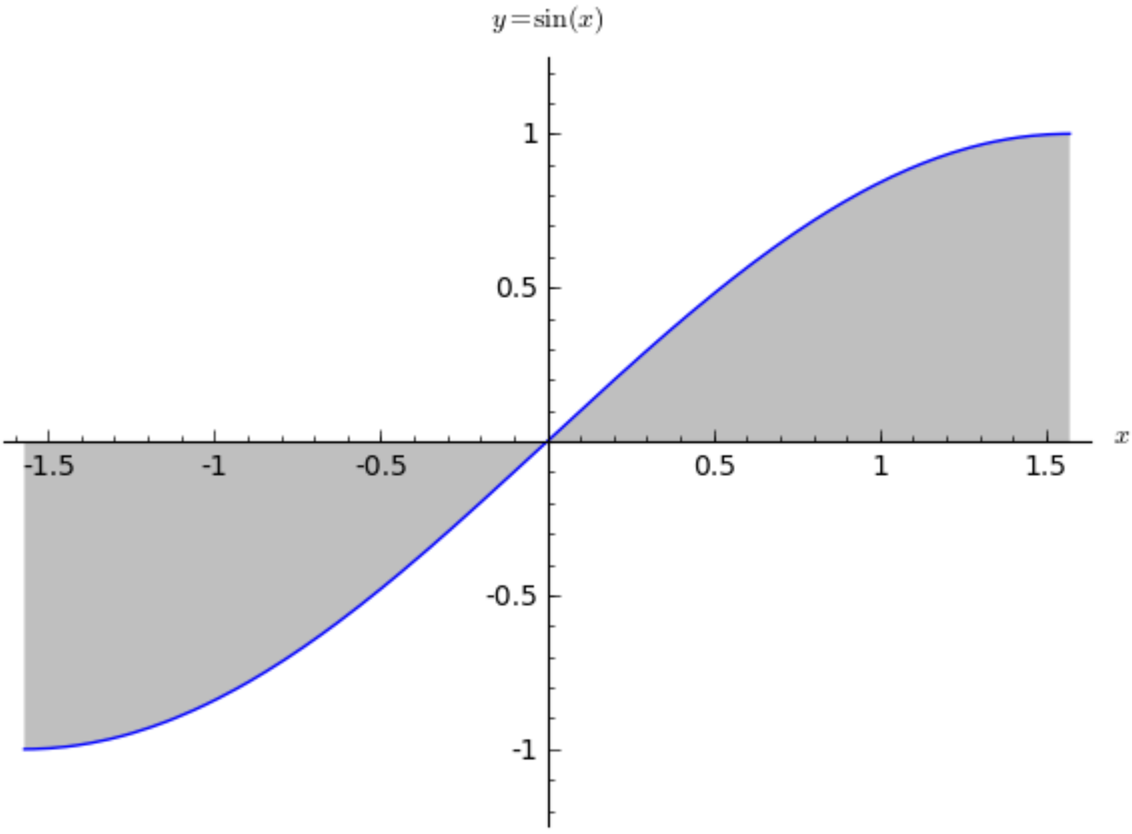
Sage tento integrál snadno spočte.

```
integral(f(x),(x,a,b))
```

    42/5

Pozor, plocha je chápána jako orientovaná. Tedy nad osou x s kladným znaménkem a pod osou x se záporným znaménkem. Například integrál z funkce sin na intervalu $\langle -\pi/2, \pi/2\rangle$ je nulový. Podívejte se na obrázek a výsledek integrace:

```
graph = Graphics()
graph_params = dict(ymin = -1.2, ymax = 1.2)
graph += plot( sin(x) , (x, -pi/2, pi/2), color="blue",
axes_labels=['$x$','$y=\sin(x)$'], fill="axis")
show (graph, figsize=6,**graph_params)
```




```
integral(sin(x),(x,-pi/2,pi/2))
```

    0

Dolní a horní součty

Uvažme rozdělení intervalu $\langle a, b\rangle$ na $n \in \mathbb{N}$ ekvidistantních (stejně dlouhých) dílků. Každý dílek má tedy délku $\Delta = (b-a)/n$. Rozdělení značínem řeckým písmenkem σ.

```
n=10
delta=(b-a)/n;
sigma=[(a+(i-1)*delta) for i in [1..n+1]]
```

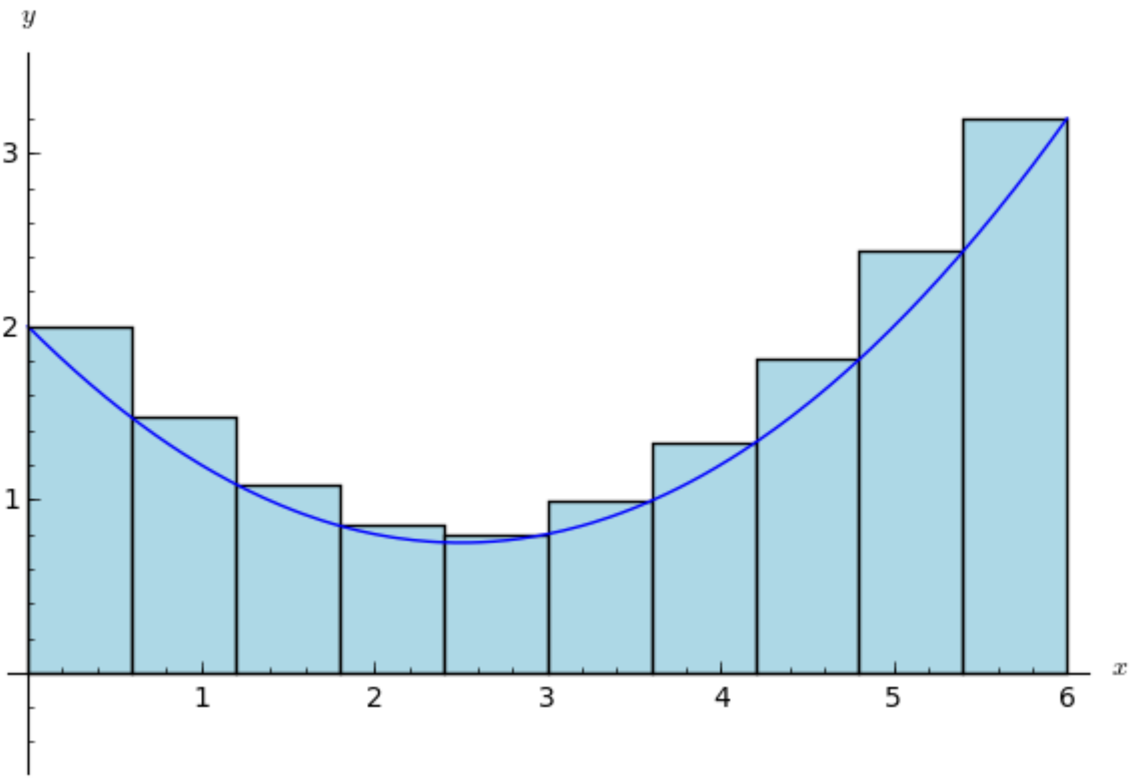
Příslušným horním a dolním součtem jsou čísla:

```
minima = [(f.find_local_minimum(sigma[i-1],sigma[i])) for i
in [1..n]]
dolni = sum([x[0]*delta for x in minima])
maxima = [(f.find_local_maximum(sigma[i-1],sigma[i])) for i
in [1..n]]
horni = sum([x[0]*delta for x in maxima])
```

A jejich grafické reprezentace.

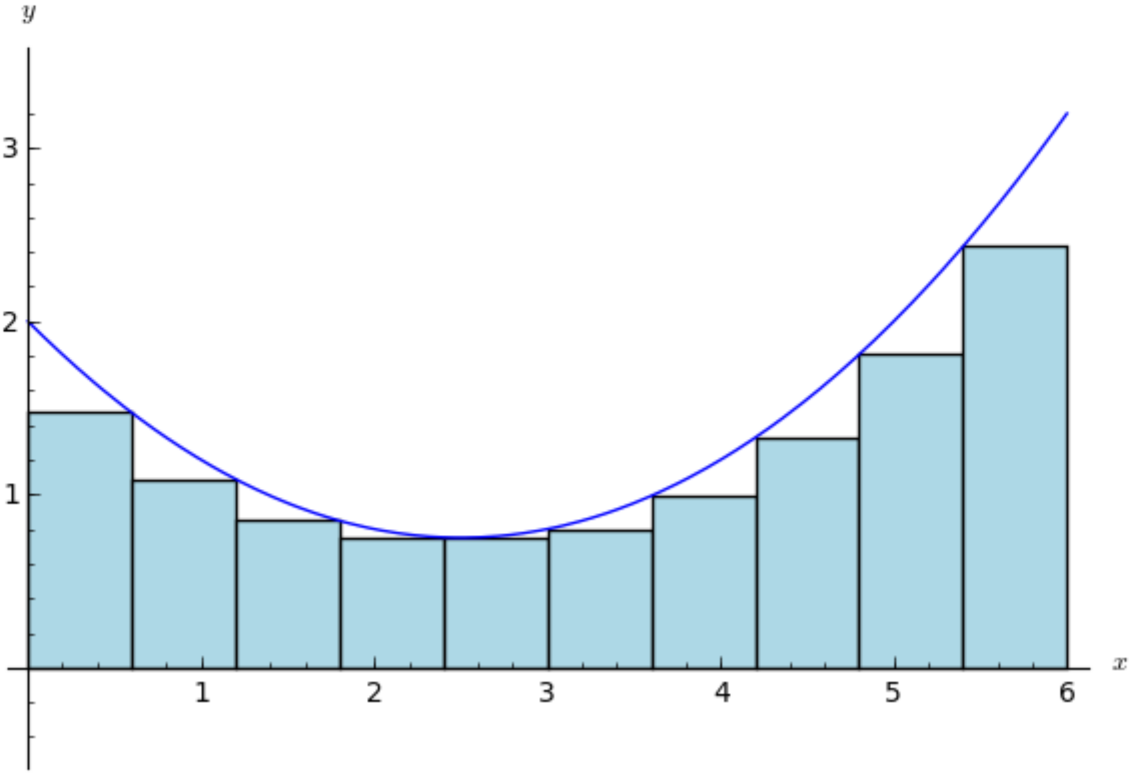
```
print('Horní součet: ' + str(horni))
graph = Graphics()
graph_params = dict(ymin = -.5, ymax = 3.5)
graph += plot( f , (x, 0, 6), color="blue", axes_labels=
['$x$','$y$'])
for i in range(n):
    graph += polygon2d([[sigma[i],0], [sigma[i+1],0],
[sigma[i+1],maxima[i][0]], [sigma[i],maxima[i][0]]],
rgbcolor='lightblue')
    graph += polygon2d([[sigma[i],0], [sigma[i+1],0],
[sigma[i+1],maxima[i][0]], [sigma[i],maxima[i][0]]],
rgbcolor='black',fill=false)
show (graph, figsize=6,**graph_params)
```

```
Horní součet: 9.58079966199
```




```
print('Dolní součet: ' + str(dolni))
graph = Graphics()
graph_params = dict(ymin = -.5, ymax = 3.5)
graph += plot( f , (x, 0, 6), color="blue", axes_labels=
['$x$','$y$'])
for i in range(n):
    graph += polygon2d([[sigma[i],0], [sigma[i+1],0],
[sigma[i+1],minima[i][0]], [sigma[i],minima[i][0]]],
rgbcolor='lightblue')
    graph += polygon2d([[sigma[i],0], [sigma[i+1],0],
[sigma[i+1],minima[i][0]], [sigma[i],minima[i][0]]],
rgbcolor='black',fill=false)
show (graph, figsize=6,**graph_params)
```

Dolní součet: 7.36200024761



Na závěr ještě interaktivní manipulace.

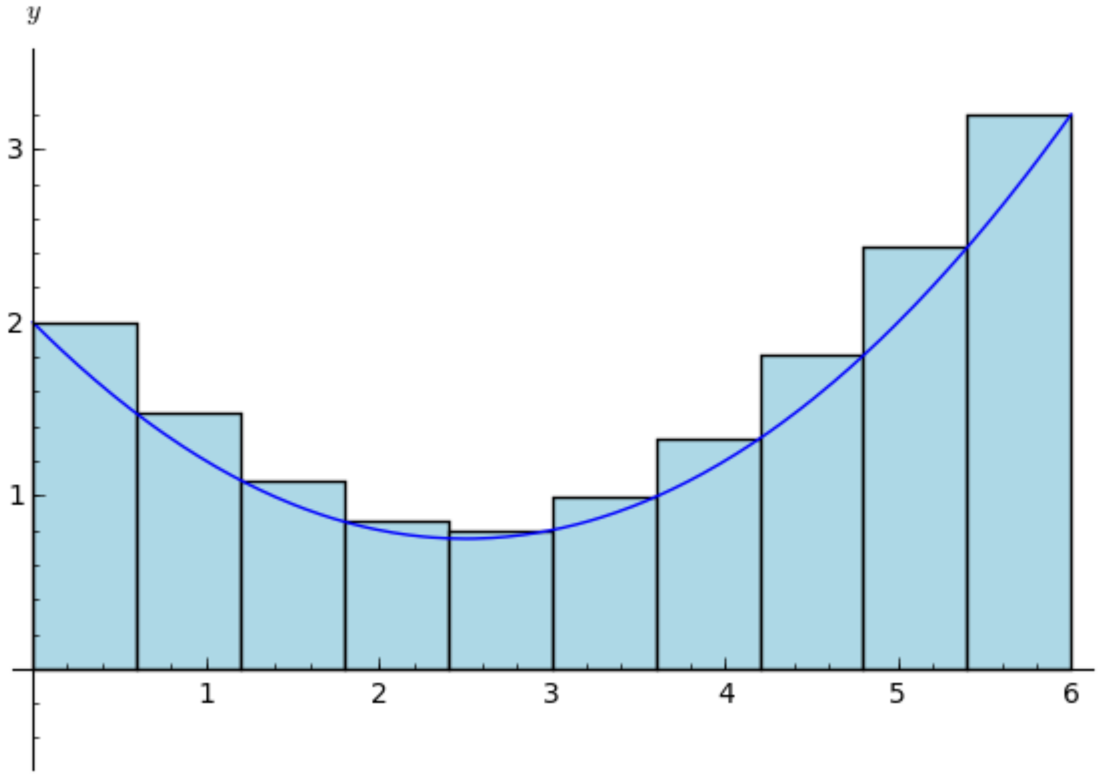
```
def soucty(n,func):
    delta=(b-a)/n;
    sigma=[(a+(i-1)*delta) for i in [1..n+1]]
    if(func == 'Dolní součet'):
        extremy= [(f.find_local_minimum(sigma[i-1],sigma[i]))
for i in [1..n]]
    else:
        extremy= [(f.find_local_maximum(sigma[i-1],sigma[i]))
for i in [1..n]]
    soucet= sum([x[0]*delta for x in extremy])
    print('Součet: ' + str(soucet))
    graph = Graphics()
    graph_params = dict(ymin = -.5, ymax = 3.5)
    graph += plot( f , (x, 0, 6), color="blue", axes_labels=
['$x$','$y$'])
    for i in range(n):
        graph += polygon2d([[sigma[i],0], [sigma[i+1],0],
[sigma[i+1],extremy[i][0]], [sigma[i],extremy[i][0]]],
rgbcolor='lightblue')
        graph += polygon2d([[sigma[i],0], [sigma[i+1],0],
[sigma[i+1],extremy[i][0]], [sigma[i],extremy[i][0]]],
rgbcolor='black',fill=false)
    show (graph, figsize=6,**graph_params)
#####################
# make Interaction
######################
```

```
@interact
def _( n= slider(srange(3,31,1),default = 10,label="$n$"),func=selector(['Horní součet', 'Dolní součet'],buttons=True,label="")):
    soucty(n,func)
```

n 10

Horní součet | Dolní součet

```
Součet: 9.58079966199
```



Primitivní funkcí k funkci f je funkce F, kterou snadno zjitíme (i ručně).

```
x = var('x')
F(x) = f.integrate(x)
show(F(x))
```

$$\frac{1}{15}\,x^3 - \frac{1}{2}\,x^2 + 2\,x$$

Tudíž plocha vypočítávaná na obrázku výše je přesně:

```
F(b) - F(a)
```

Integrální součet

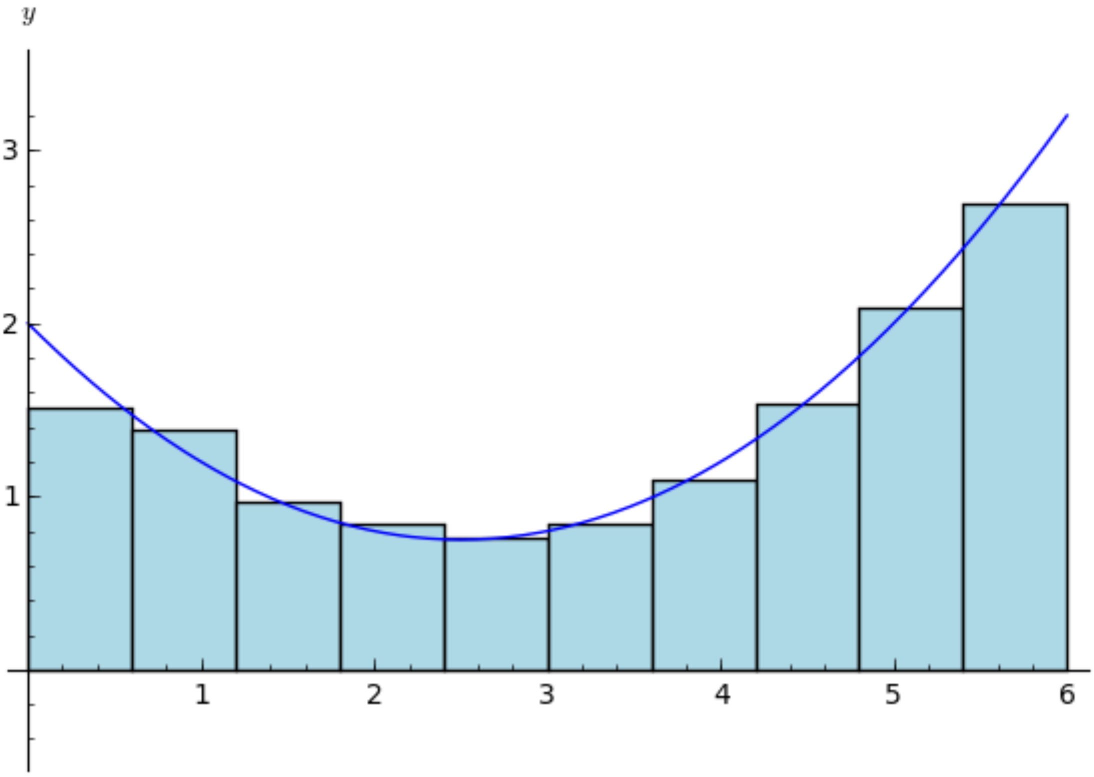
Jak víme z přednášek, přibližnou hodnotu určitého integrálu nemusíme počítat pomocí maxim a minim na dělících intervalech, která je složitá hledat, ale můžeme náhodně volit hodnotu funkce v bodě dělícího intervalu. Pro jednoduchost volme třeba jeho střed.

```
def soucty(n,func):
    delta=(b-a)/n;
    sigma=[(a+(i-1)*delta) for i in [1..n+1]]
    if(func == 'Náhodně'):
        extremy= [(f(uniform(sigma[i-1],sigma[i]))) for i in
[1..n]]
    else:
        extremy= [(f((sigma[i-1]+sigma[i])/2)) for i in
[1..n]]
    soucet= sum([x*delta for x in extremy])
    print('Součet: ' + str(N(soucet)))
    graph = Graphics()
    graph_params = dict(ymin = -.5, ymax = 3.5)
    graph += plot( f , (x, 0, 6), color="blue", axes_labels=
['$x$','$y$'])
    for i in range(n):
        graph += polygon2d([[sigma[i],0], [sigma[i+1],0],
[sigma[i+1],extremy[i]], [sigma[i],extremy[i]]],
rgbcolor='lightblue')
        graph += polygon2d([[sigma[i],0], [sigma[i+1],0],
[sigma[i+1],extremy[i]], [sigma[i],extremy[i]]],
rgbcolor='black',fill=false)
    show (graph, figsize=6 ,**graph_params)
#####################
# make Interaction
######################
@interact
def _( n= slider(srange(3,31,1),default =
10,label="$n$"),func=selector(['Náhodně', 'Průměry
(středy)'],buttons=True,label="")):
    soucty(n,func)
```

n 10

Náhodně | Průměry (středy)

```
Součet: 8.22646682092677
```

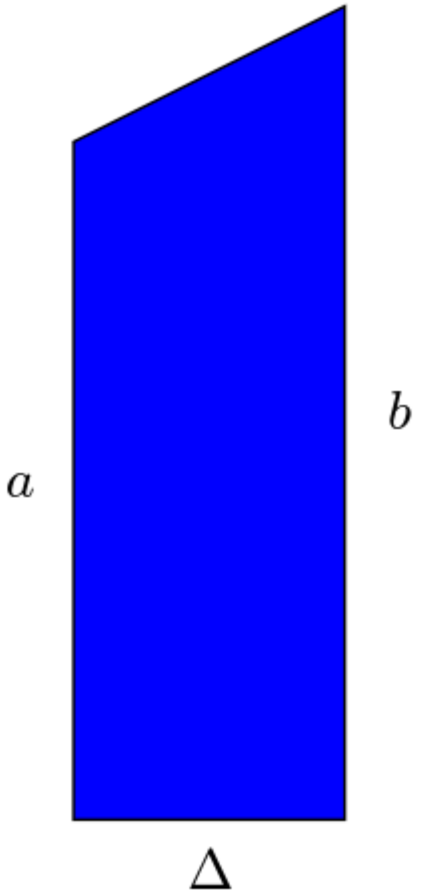


Lichoběžníkové pravidlo

V případě braní průměrné hodnoty ale nejde o nic jiného než o sčítání obsahů lichoběžníků. Pokud si představíte lichoběžník s podstavou Δ a svislými stranami dlouhými a a b, $a < b$, pak jeho obsah je

$$S = a \cdot \Delta + \frac{1}{2}\,(b - a) \cdot \Delta = \Delta \cdot \frac{1}{2}\,(a + b).$$

```
graph = Graphics()
graph += polygon2d([[0,0], [1,0], [1,3], [0,2.5]],
rgbcolor='blue', axes=false)
graph += polygon2d([[0,0], [1,0], [1,3], [0,2.5]],
rgbcolor='black',fill=false, axes=false)
graph += text('$a$',(-0.2,1.25),color='black',fontsize=20)
graph += text('$b$',(1.2,1.5),color='black',fontsize=20)
graph += text('$\Delta$',(0.5,-.2),color='black',fontsize=20)
show (graph ,**graph_params)
```

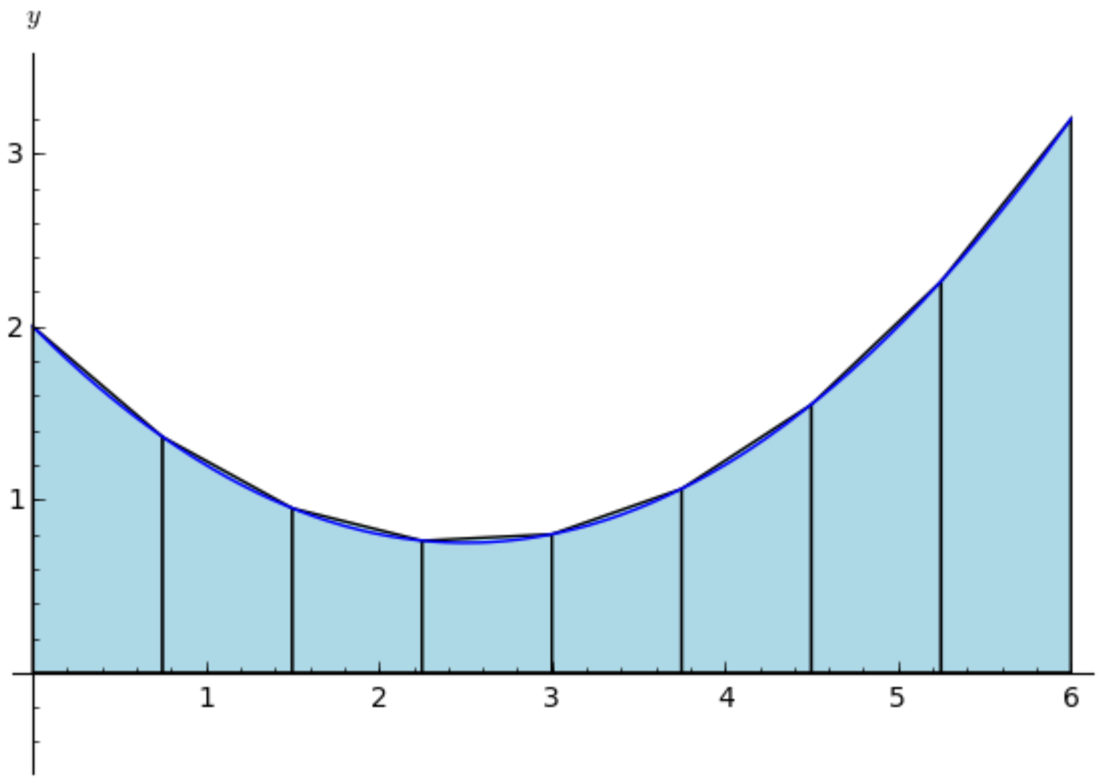



```
def soucty(n):
    delta=(b-a)/n;
    sigma=[(a+(i-1)*delta) for i in [1..n+1]]
    soucet=0
    for i in range(n):
        soucet+=delta/2*(f(sigma[i])+f(sigma[i+1]))
    show('Součet: ' + str(N(soucet)))
    graph = Graphics()
    graph_params = dict(ymin = -.5, ymax = 3.5)
    graph += plot( f , (x, 0, 6), color="blue", axes_labels=
['$x$','$y$'])
    for i in range(n):
        graph += polygon2d([[sigma[i],0], [sigma[i+1],0],
[sigma[i+1],f(sigma[i+1])], [sigma[i],f(sigma[i])]],
rgbcolor='lightblue')
        graph += polygon2d([[sigma[i],0], [sigma[i+1],0],
[sigma[i+1],f(sigma[i+1])], [sigma[i],f(sigma[i])]],
rgbcolor='black',fill=false)
```

```
    show (graph, figsize=6 ,**graph_params)
##    ###################
# make Interaction
######################
@interact
def _( n= slider(srange(3,31,1),default = 10,label="$n$")):
    soucty(n)
```

n 10

Součet: 8.51250000000000



Simpsonovo pravidlo

Na dělícím intervalu $\langle c, d\rangle$ nahraďme integrand kvadratickou funkcí (ne přímkou jako výše) procházející body $(c, f(c))$, $(d, f(d))$ a průměrem $(m, f(m))$, kde $m = (c + d)/2$. Následuje odvození vzorce.

```
aa,bb,cc,c,d,fc,fcd2,fd,ff = var('aa,bb,cc,c,d,fc,fcd2,fd,ff')
x=var('x')
inter(x) = aa*x^2 + bb*x + cc;
solution=solve([inter(c) == fc,inter((c + d)/2) ==
```

```
fcd2,inter(d) == fd],(aa,bb,cc))
show(solution)
```

$$\left[\left[aa = \frac{2\,(fc - 2\,fcd_2 + fd)}{c^2 - 2\,cd + d^2}, bb = -\frac{(c + 3\,d)fc - c(4\,fcd_2 - 3\,fd)}{c^2 - 2\,cd + d^2}\right.\right.$$

Integrujme tuto funkce na $\langle c, d\rangle$, dostáváme:

```
subs=inter().subs_expr(*solution[0])
show(subs.integrate(x,c,d).full_simplify())
```

$$-\frac{1}{6}\,(c-d)fc - \frac{2}{3}\,(c-d)fcd_2 - \frac{1}{6}\,(c-d)fd$$

Dostáváme tak Simpsonovo pravidlo, které spočívá v nahrazení integrálu nad intervalem $\langle c, d\rangle$ integrálem z tohoto kvadratického interpolantu. Konkrétně

$$\int_c^d f(x)dx \approx \frac{1}{6}\,(d-c)\left(f(c) + f(d) + 4f\left(\frac{c+d}{0}\right)\right).$$

```
f(x) = 2 + sin(4*x)
goal = integral(f,x,a,b)
def simpson(n):
    delta=(b-a)/n;
    sigma=[(a+(i-1)*delta) for i in [1..n+1]]
    simpson = [1/6*delta*
(f(sigma[i])+f(sigma[i+1])+4*f((sigma[i]+sigma[i+1])/2)) for
i in [0..n-1]]
    soucet=0
    for i in range(n):
        soucet+=simpson[i]
    print('Součet podle Simpsona: ' + str(N(soucet)))
    print('Skutečná hodnota: ' + str(N(goal)))
    graph = Graphics()
    graph_params = dict(ymin = -.5, ymax = 3.5)
    graph += plot( f , (x, a, b), color="blue", axes_labels=
['$x$','$y$'])
    for i in range(n):
        graph += line([(sigma[i],0),
(sigma[i],f(sigma[i]))],linestyle='dashed',color='grey')
    show (graph, figsize=6,**graph_params)
#####################
# make Interaction
######################
```

```
@interact
def _( n= slider(srange(3,31,1),default = 10,label="$n$")):
    simpson(n)
```

n 10

Není jednoduché napočítávat kvadratické interpolace, proto zobrazujeme jenom šířky sloupců pro ilustraci.