Fork me on GitHub

hint_2-3.ipynb

Download this notebook in HTML

In [42]:
```
using Plots
using SymPy

toFunc(str) = eval(Meta.parse(str))
```

Out [42]: toFunc (generic function with 1 method)

In [9]:
```
@syms a, x
```

Out [9]: (a, x)

問 1 (2)

In [3]:
```
f = log(1-x)
```

Out [3]: log(1 - x)

In [4]:
```
diff(f, x)
```

Out [4]: $\frac{-1}{1 - x}$

問 1 (5)

In [5]:
```
g = 3^x * (x^2 + x)
```

Out [5]: $3^x \cdot (x^2 + x)$

In [6]:
```
diff(g, x)
```

Out [6]: $3^x \cdot (2 \cdot x + 1) + 3^x \cdot (x^2 + x) \cdot \log(3)$

注意

a^x を x で微分すると...

In [8]:
```
diff( a^x, x )
```

Out [8]: $a^x \cdot \log(a)$

なぜかと言うと、$a^x = (e^{\log a})^x = e^{x \log a}$ だから.

問 2 (1)

In [10]:
```
h = 2*x^2*sqrt(x) - 5*x^2
```

Out [10]:
```
   5/2      2
2⋅x    - 5⋅x
```

In [11]:
```
h1 = diff(h, x)
```

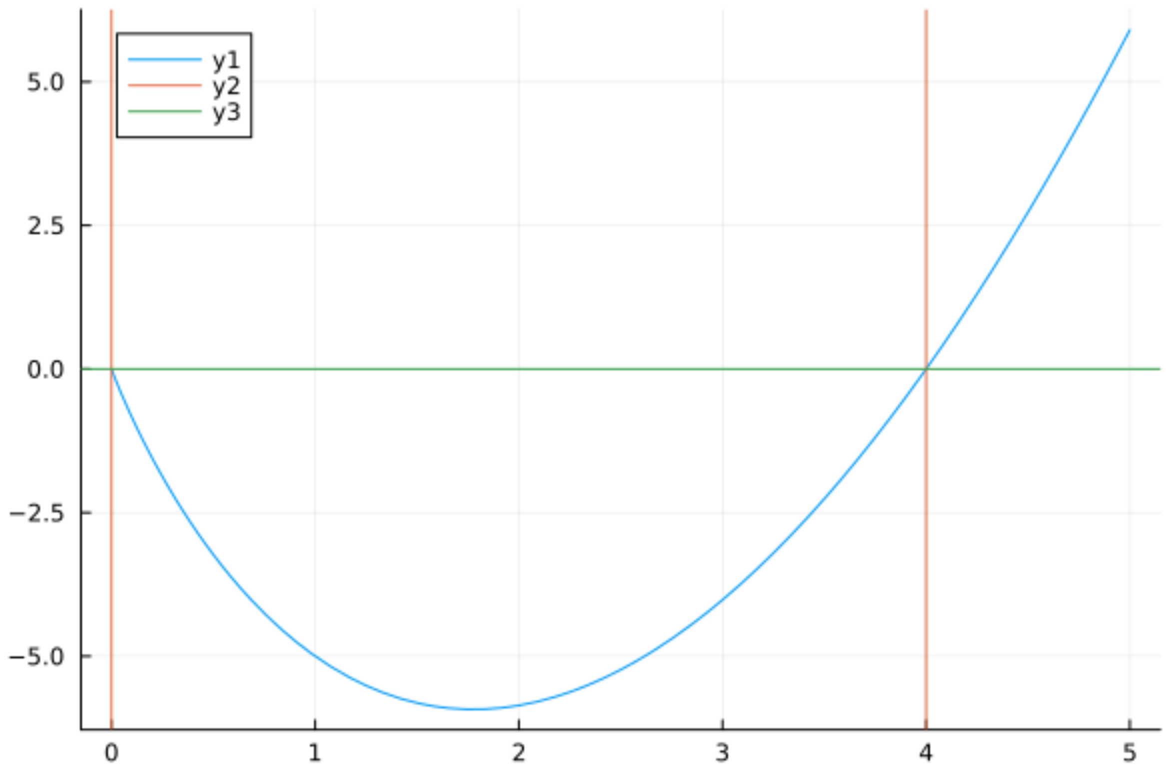
Out [11]:
```
   3/2
5⋅x    - 10⋅x
```

In [17]:
```
solve( h1, x )
```

Out [17]:
```
2-element Vector{Sym{PyCall.PyObject}}:
 0
 4
```

In [26]:
```
plot(h1)
vline!([0,4])
hline!([0.0])
```

Out [26]:



In [12]:
```
h2 = diff(h1, x)
```

Out [12]:
```
15⋅√x
───── - 10
  2
```

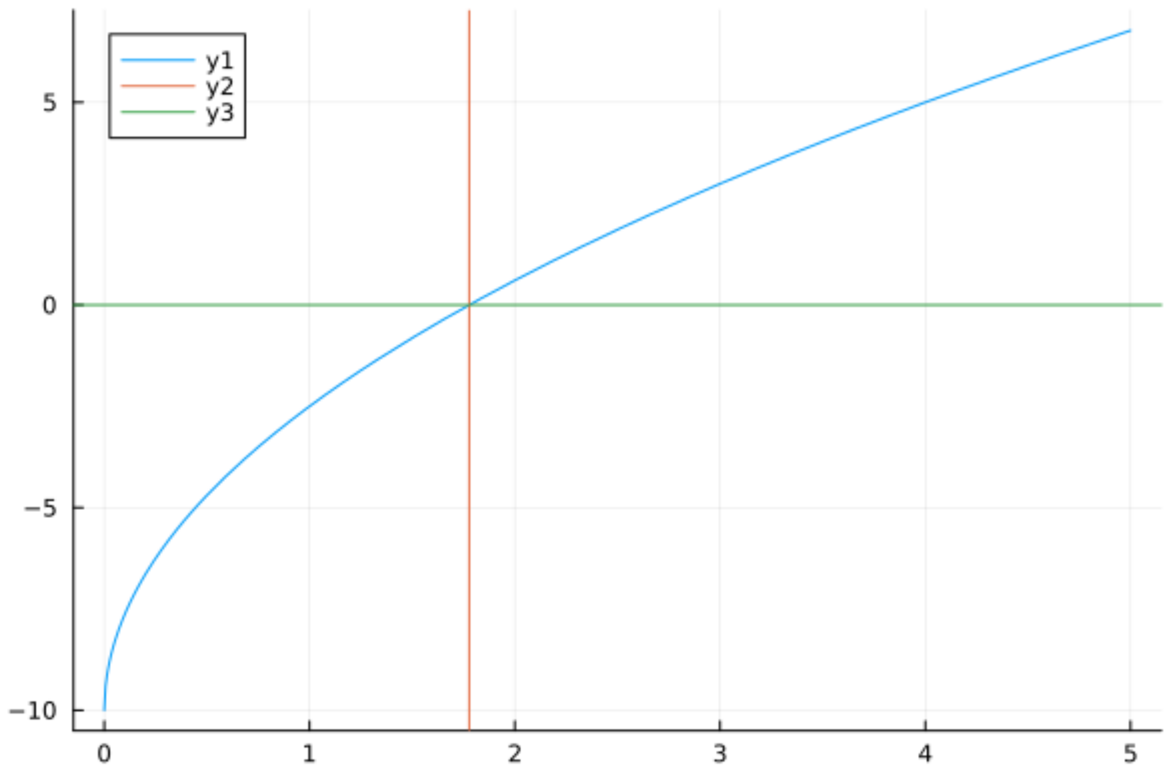
In [16]:
```
solve( h2, x )[1]
```

Out [16]: 16/9

In [25]:
```
plot(h2)
vline!([16/9])
```

```
hline!([0.0])
```

Out [25]:



情報を集めて整理すると…

x	$h(x)$	$h'(x)$	$h''(x)$	**備考**
$x = 0$	0	0	負	この点では平ら．下に凸なのですぐ下がっていく．
$0 < x < 16/9$	-	負	負	減少、下に凸
$x = 16/9$	$-1792/243 \cong -7.3745$	負	0	減少, **変曲点**
$16/9 < x < 4$	-	負	正	減少、上に凸
$x = 4$	-16	0	正	**極小点**、 上に凸
$4 < x$	-	正	正	増大、上に凸

実際にグラフを描いて答え合わせ．

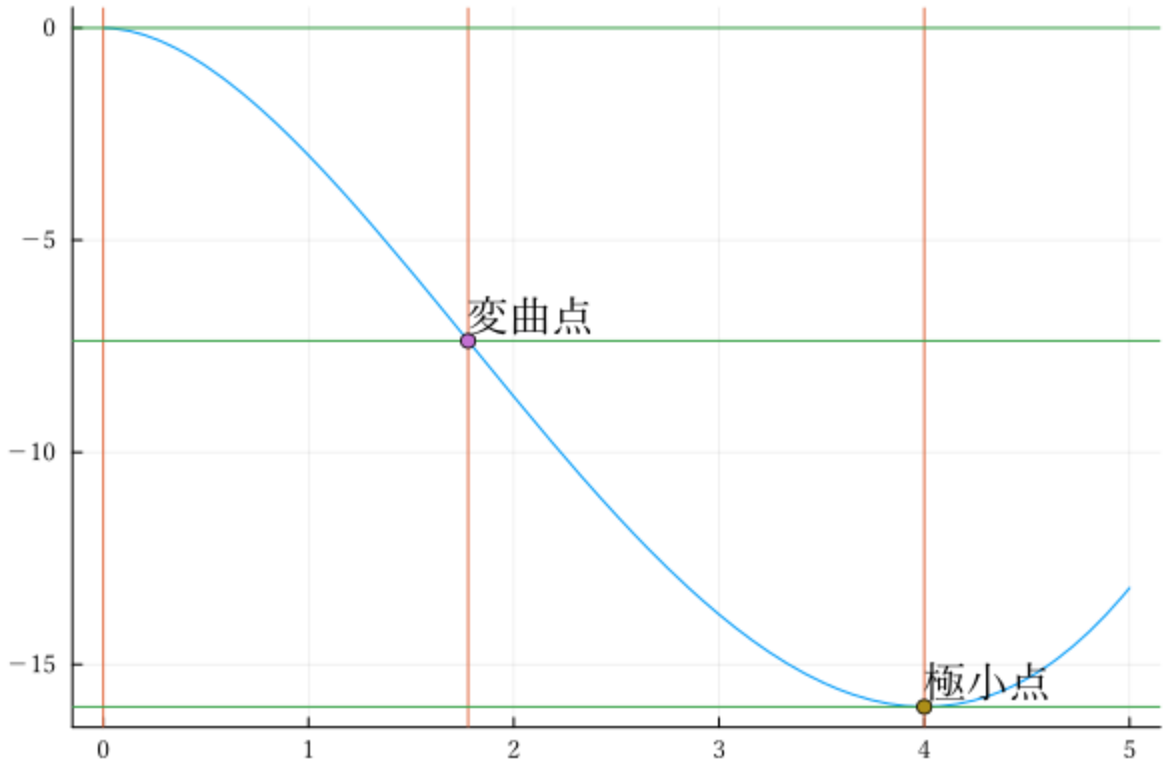
In [37]:

```
plot(h, legend = false, fontfamily = "yumin")
vline!([0,16/9,4])
hline!([0,-1792/243,-16])

plot!( (16/9, -1792/243), marker = :circle )
annotate!(16/9, -1792/243, ("変曲点", :bottom, :left))

plot!( (4, -16), marker = :circle )
annotate!( 4, -16, ("極小点", :bottom, :left))
```

Out [37]:



問 5 (3)

In [38]:

```
k = x^3 - 3
```

Out [38]: $x^3 - 3$

In [39]:

```
k1 = diff(k, x)
```

Out [39]: $3 \cdot x^2$

In [49]:

```
newton(y) = y - (y^3 - 3)/(3*y^2)
```

Out [49]: newton (generic function with 1 method)

In [51]:

```
x0 = 1.5
```

Out [51]: 1.5

In [53]:

```
x1 = newton(x0)
```

Out [53]: 1.4444444444444444

In [54]:

```
x2 = newton(x1)
```

Out [54]: 1.4422529037913654

In [55]:

```
x3 = newton(x2)
```

Out [55]: 1.442249570315113

In [56]:

```
x4 = newton(x3)
```

Out [56]: 1.4422495703074083

In [58]:
```
3^(1/3)
```

Out [58]: 1.4422495703074083

これもまとめると...

Newton法の適用回数 k	近似値 x_k	備考
0	1.5	初期値(適当に真ん中にした)
1	1.4444444444444444	
2	1.4422529037913654	
3	1.442249570315113	
4	1.4422495703074083	
	1.4422495703074083	参考値: 真値 $3^{(1/3)}$

となっており、初期値1.5 に対して、Newton法を 4回適用すると十分な近似値を得た、ということがわかる.

In [ ]:

Hey there! If you have any feedback for this tool - issues, ideas for improvement, or you want to just tell me about your use case for this, I'd love to know. E-mail me or tweet at me.