## *Das Newton-Verfahren (und das Paket "Student")*

```
> with(plots):
```

```
> with(Student[Calculus1]):
```

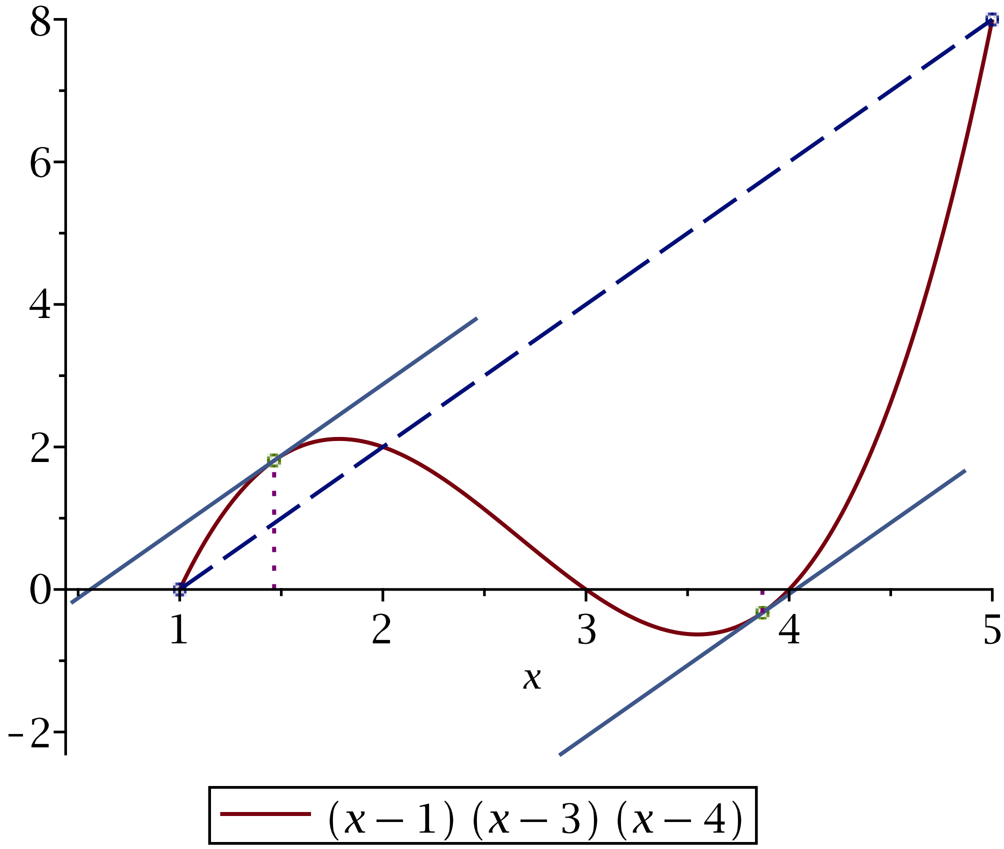
Das Paket "Student[Calculus1]" wird geladen. Es enthält eine Reihe von Tutorials und Routinen zum Nacharbeiten
und selbständigen Erforschen von Standardthemen der Differential- und Integralrechnung.
Zunächst wird eine Funktion $f$ definiert:

```
> f:=x->(x-1)*(x-3)*(x-4);
```

$$f := x \mapsto (x-1)\,(x-3)\,(x-4) \tag{1}$$

Der Mittelwertsatz (Englisch: Mean value theorem) kann mit Hilfe des Kommandos "MeanValueTheorem" aus dem "Student" Paket illustriert werden:

```
> MeanValueTheorem(f(x),x=1..5);
```

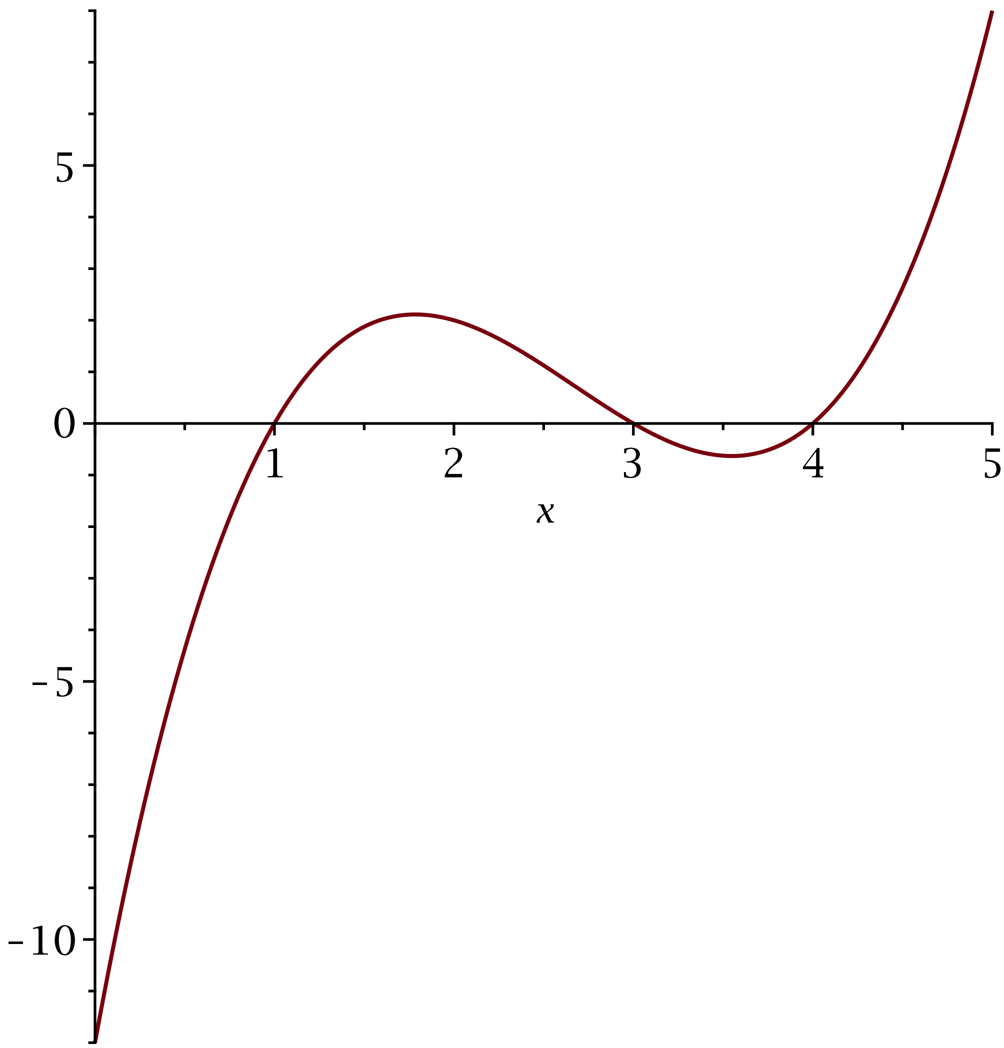



The Mean Value Theorem is illustrated for the function
$(x-1)\,(x-3)\,(x-4) = (x-1)\,(x-3)\,(x-4)$.

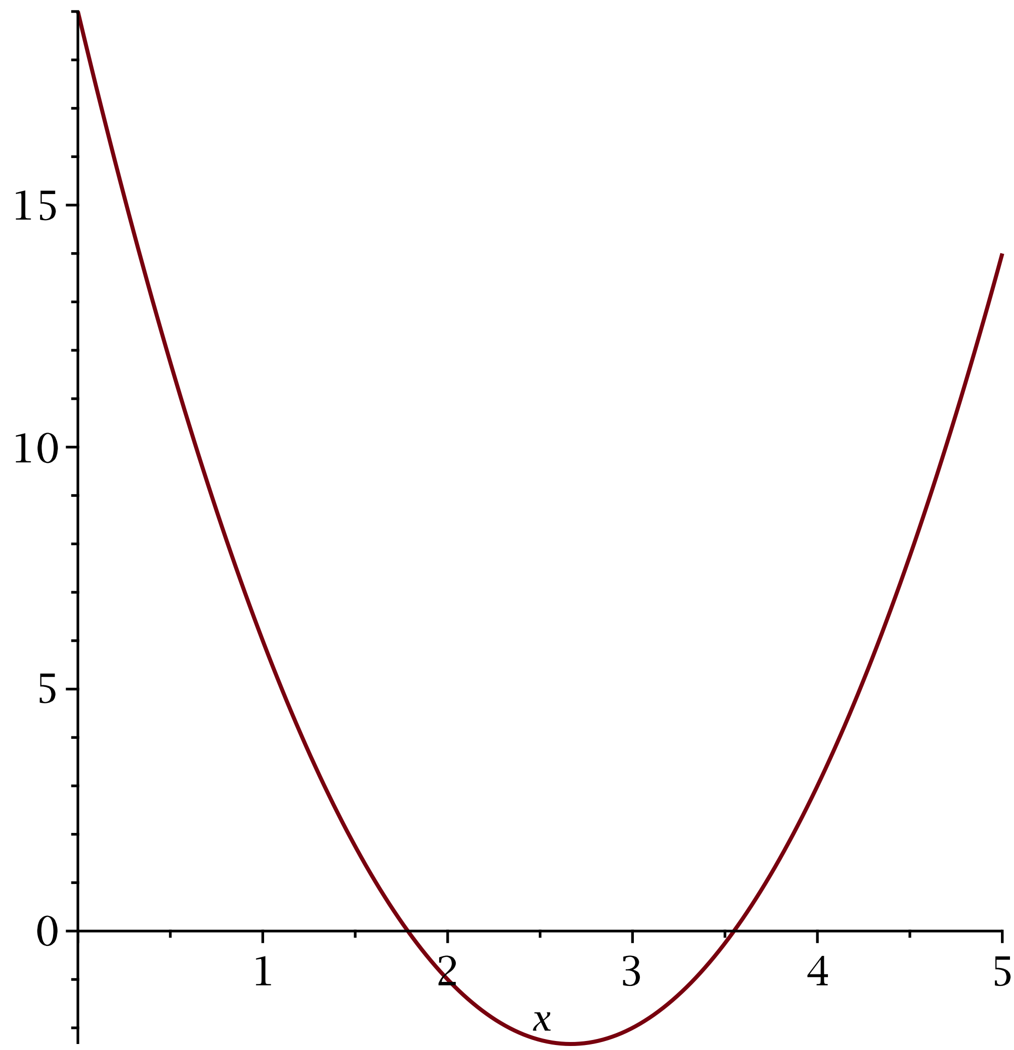
Man erhält eine Zeichnung des Graphen der Funktion $f$ im angegebenen Intervall [$a$, $b$], die Sehne, die den Punkt ($a$, $f(a)$) mit dem Punkt ($b$, $f(b)$) verbindet, sowie Tangenten an Punkte des Graphen, die parallel zu der Sehne sind (der Mittelwertsatz garantiert, dass wenigstens ein Punkt im Intervall existiert, in dem die Tangente parallel zu der Sehne ist).

Wir betrachten jetzt den Graphen der Funktion $f$ sowie die Graphen der ersten und der zweiten Ableitung:
Da $f$ eine ganzrationale Funktion (Polynom) vom Grad 3 ist, ist die Ableitung vom Grad 2,
hat also als Graphen eine Parabel.
Die zweite Ableitung hat Grad 1 (ist eine lineare Funktion), ihr Graph ist also eine Gerade.
Man sieht durch Betrachten der Graphen: In dem Teilintervall, in dem die zweite Ableitung negativ ist, fällt die erste Ableitung monoton, die Funktion ist in diesem Teilintervall konkav.
In dem Teilintervall, in dem die zweite Ableitung positiv ist, wächst die erste Ableitung monoton, die Funktion ist in diesem Teilintervall konvex.
Randpunkt dieser beiden Teilintervalle ist der Wendepunkt (Nullstelle der zweiten Ableitung mit Extremwert der ersten Ableitung),
in dem die erste Ableitung ihr Minimum hat. Die beiden Nullstellen der ersten Ableitung liefern bei dieser Funktion lokale Extrema (ein Minimum und ein Maximum).

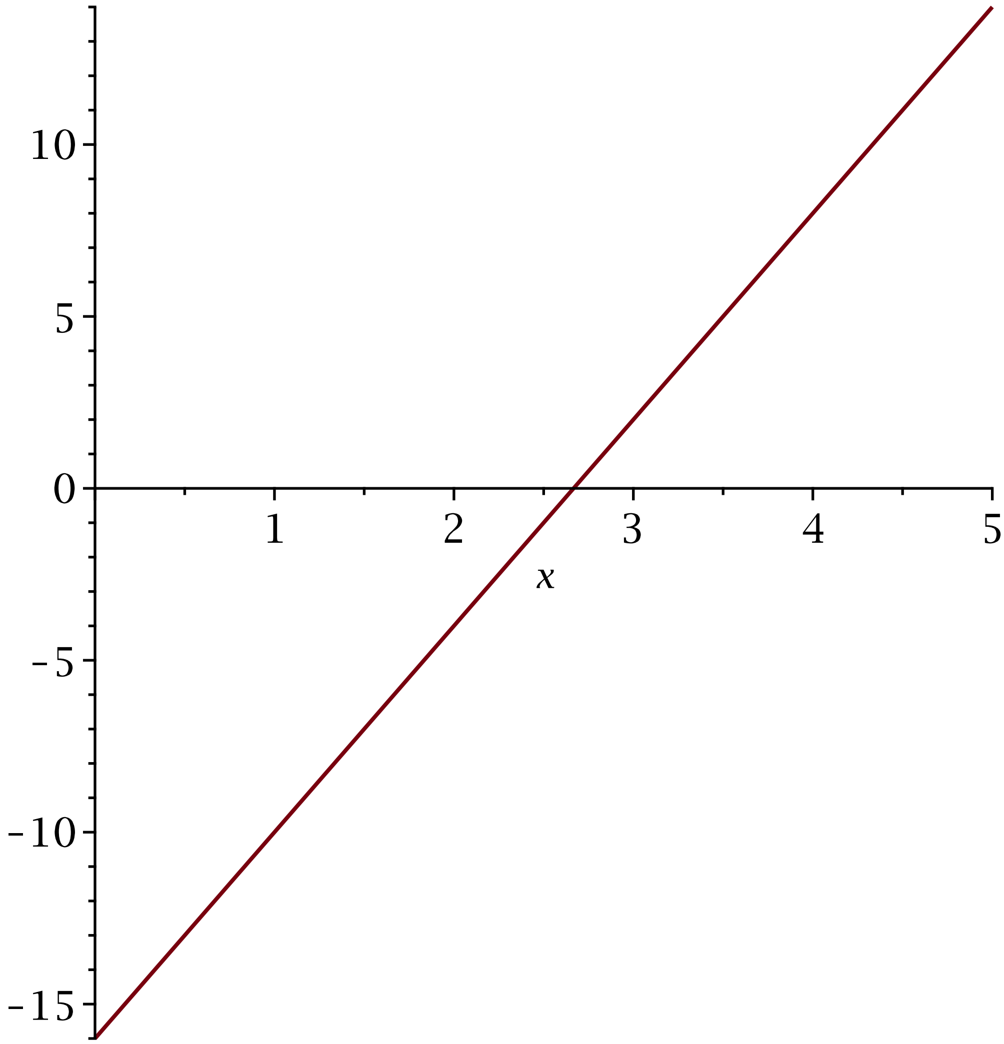
```
> plot(f(x),x=0..5);
```




```
> plot(D(f)(x),x=0..5);
```

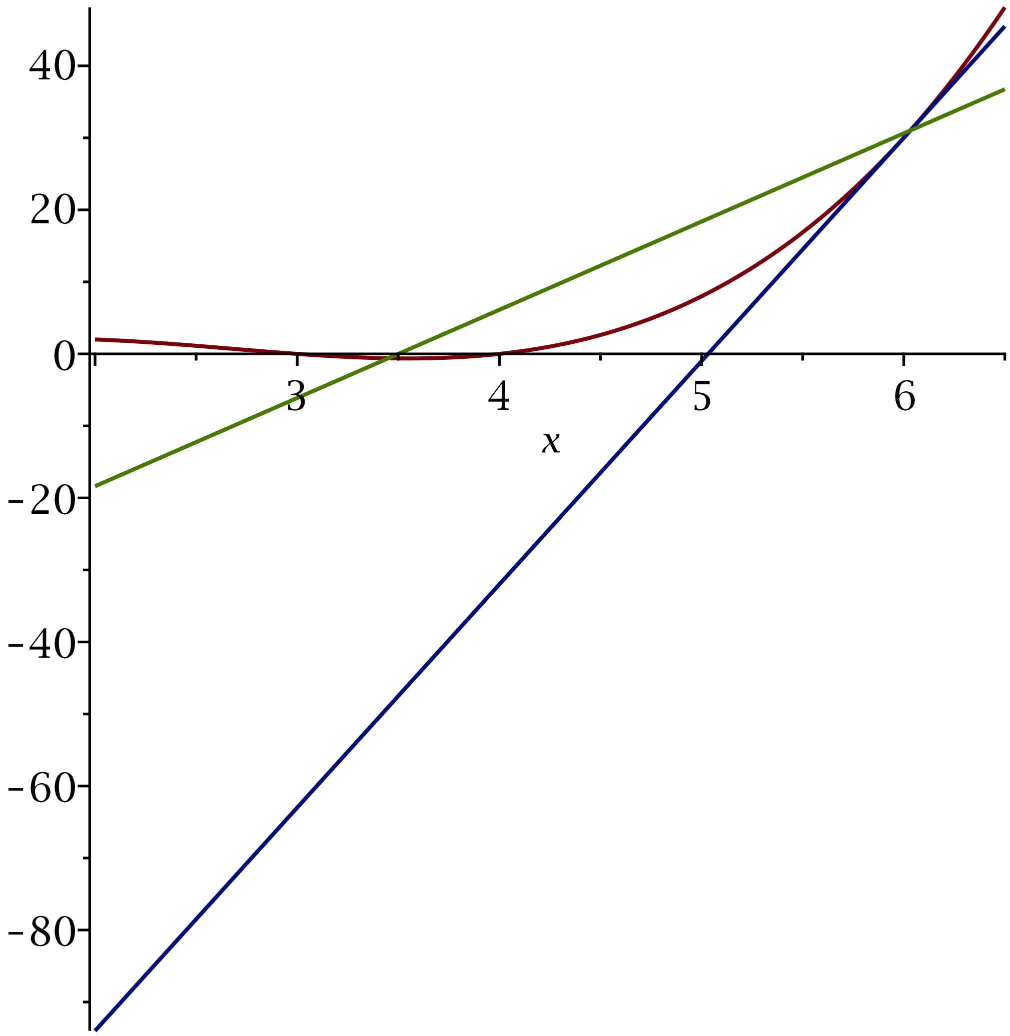



```
> plot(D(D(f))(x),x=0..5);
```

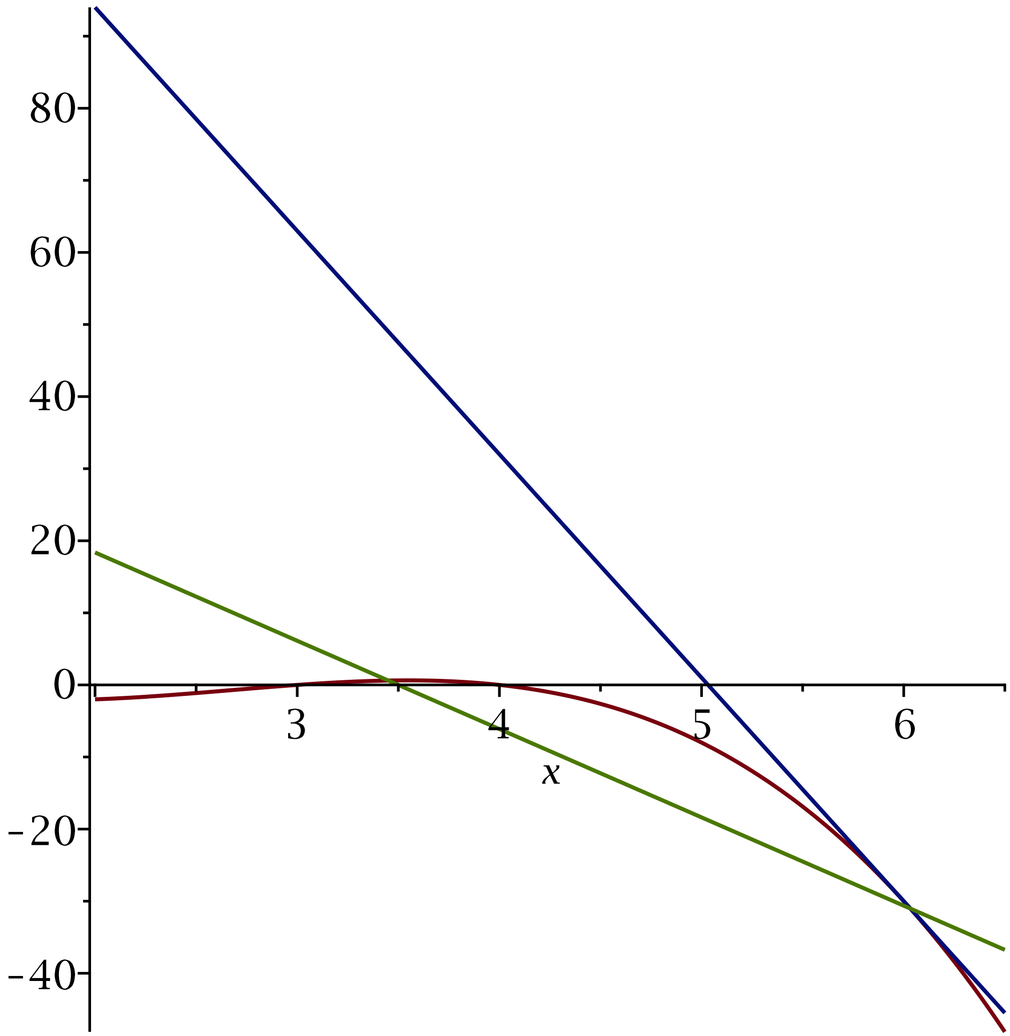



Um das Newton-Verfahren zu illustrieren zeichnen wir jetzt den Graphen der Funktion im Intervall $[3.5, 6]$, in dem die Funktion konvex ist, gemeinsam mit der Sekante durch die Punkte $(3.5, f(3.5))$ und $(6, f(6))$ sowie der Tangente an den Graphen im Punkt $(6, f(6))$. Man sieht, dass die Sehne oberhalb des Graphen liegt, die Tangente unterhalb, wie dies bei konvexen Funktionen stets der Fall ist.
Danach wird das Gleiche für den Graphen der Funktion $-f$ gemacht, die konkav ist, bei der also umgekehrt
die Sehne unterhalb, die Tangente oberhalb des Graphen verläuft.

```
> plot({f(x),((f(6)-f(3.5))/2.5)*(x-3.5),D(f)(6)*(x-6)+f(6)},x=2.
  .6.5);
```

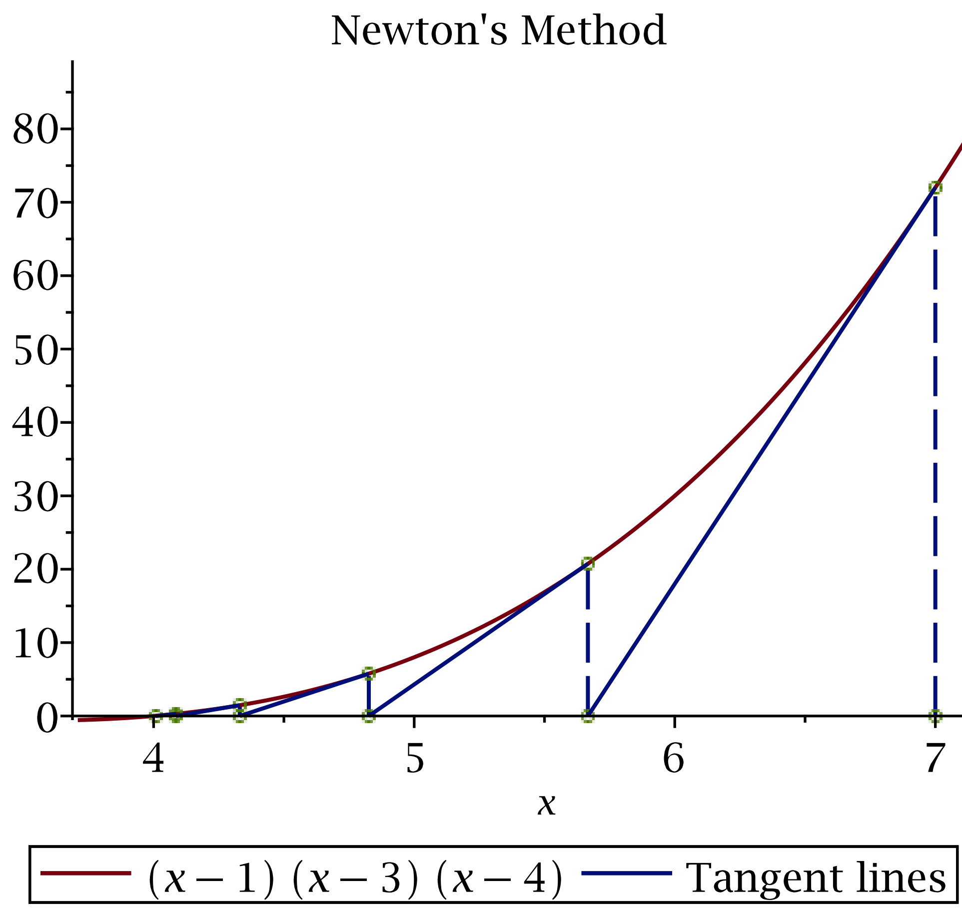



```
> plot({-f(x),-((f(6)-f(3.5))/2.5)*(x-3.5),-D(f)(6)*(x-6)-f(6)},x=
  2..6.5);
```




Nun sehen wir uns an, wie das "Student"-Paket das Newton-Verfahren illustriert: Man sieht wieder den Graphen, als Startpunkt wird hier $x_0 = 7$ gewählt. Man sieht jeweils die Tangente in den Punkten mit Abszisse $x_n$ und den Schnittpunkt $x_{n+1}$ dieser Tangente mit der $x$-Achse für die ersten 5 Schritte des Verfahrens. Die Nullstelle 4 wird damit schon recht gut approximiert.

```
> NewtonsMethod(f(x),7,output=plot);
```




From the initial point 7, at most 5 iteration(s) of Newton's method for

$$(x-1)\,(x-3)\,(x-4) = (x-1)\,(x-3)\,(x-4)$$

Anschließend ein Beispiel dafür, wie man das gleich Verfahren in MAPLE von Hand programmieren und die entsprechenden Graphen zeichnen kann:

```
> x[0]:=7.0;
```

$$x_0 := 7.0 \quad \textbf{(2)}$$

```
> for n from 1 to 7 do x[n]:=x[n-1]-f(x[n-1])/D(f)(x[n-1]) end do;
```

$$x_1 := 5.666666667$$

$$x_2 := 4.825825826$$

$$x_3 := 4.330773364$$

$$x_4 := 4.085368049$$

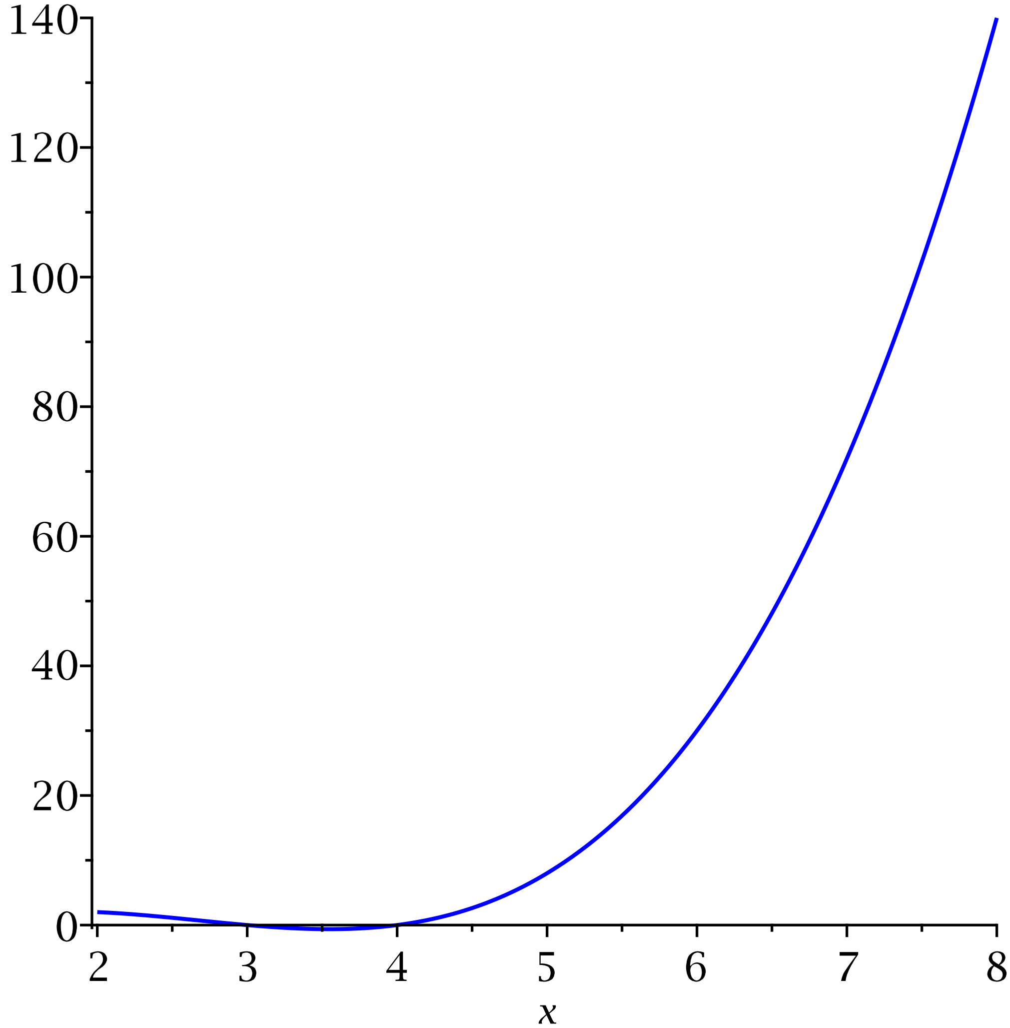
$$x_5 := 4.008204229$$

$$x_6 := 4.000088179$$

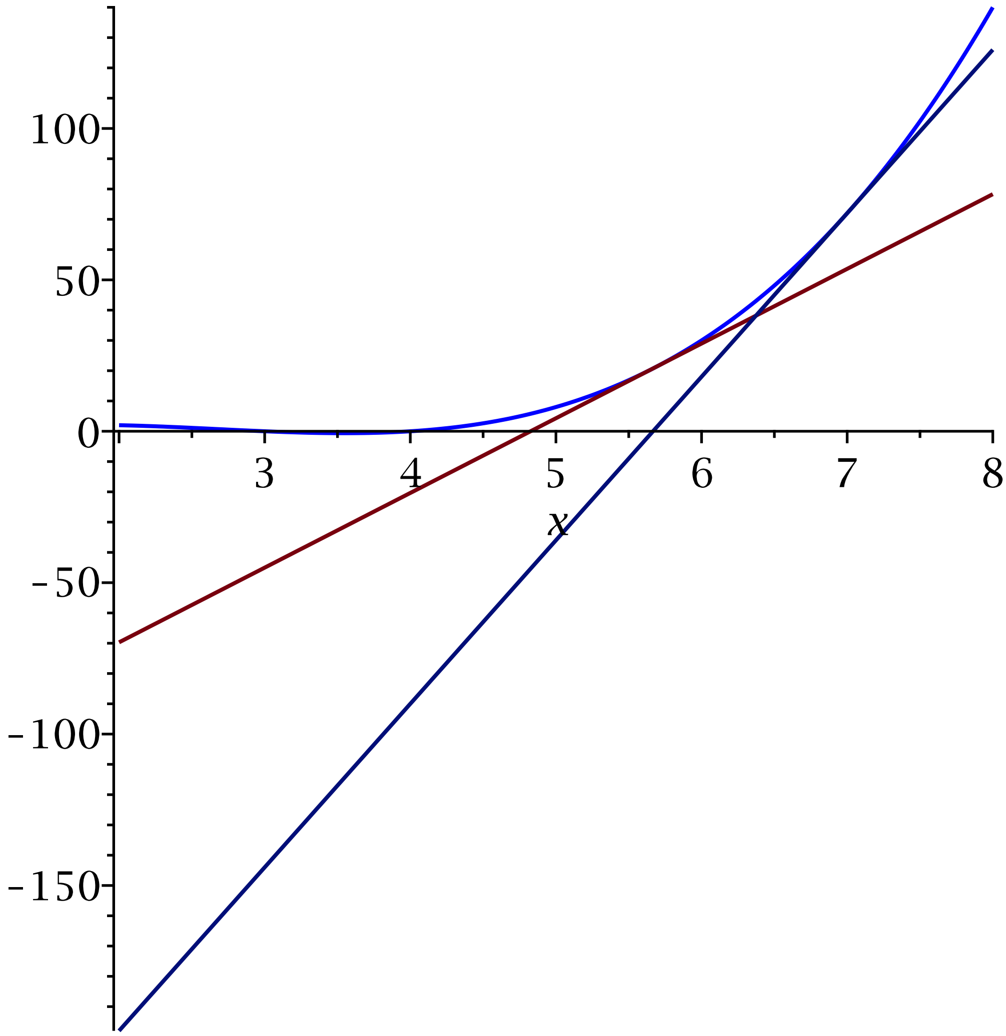
$$x_7 := 4.000000010 \quad \textbf{(3)}$$

```
> funktion:=plot(f(x),x=2..8,color=blue):
```

```
> tangenten:=plot({D(f)(x[0])*x+f(x[0])-D(f)(x[0])*x[0],D(f)(x[1])*
  x+f(x[1])-D(f)(x[1])*x[1]},x=2..8):
> display(funktion);
```




```
> display(funktion,tangenten);
```



100
50
0
-50
-100
-150
3
4
5
6
7
8
x