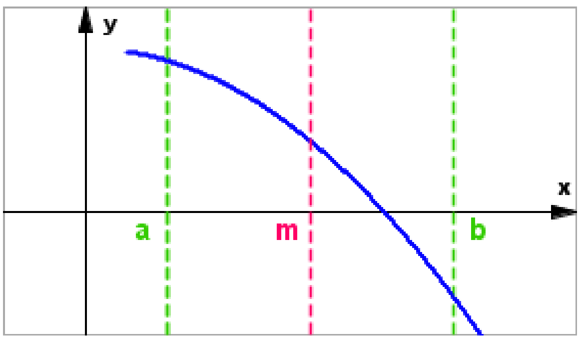
Algorithmen zur Nullstellenbestimmung

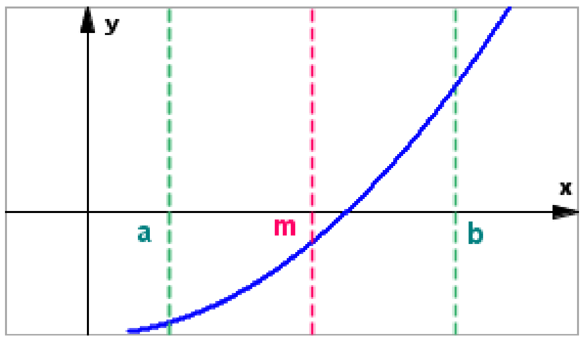
Bestimmt werden sollen Lösungen $\bar{x}$ der Gleichung **f(x) = 0** für eine stetige Funktion f .
Diese Lösungen $\bar{x}$ nennt man Nullstellen von f .

1. Methode: **Bisektionsverfahren**

Wir gehen aus von einem Intervall [a ; b] , das eine Nullstelle einschließt.
Voraussetzung: $f(a) \cdot f(b) < 0$, d.h. die Funktionswerte haben verschiedene Vorzeichen.

Aus den Werten a und b wird durch den Mittelwert $m = (a + b) / 2$ ein neuer Näherungswert für $\bar{x}$ gebildet.





Gilt dann $f(a) \cdot f(m) < 0$, so wird b = m gesetzt ; andernfalls wird a = m gesetzt.
Wir erhalten so ein neues Intervall [a ; b] , das die Nullstelle enthält.

Diese beiden Schritte (Mittelwert m bilden ; Intervallgrenzen neu setzen) werden fortgesetzt, bis b sich von a nur noch um eine Kleinigkeit (epsilon > 0) unterscheidet.

Algorithmus:

```
Eingabe a, b  mit  f(a) · f(b) < 0  sowie eps = 1E-10  (z.B.)
Wiederhole
    m = (a+b) / 2
    Falls  f(a) · f(m) < 0
        dann  b = m
        sonst  a = m
bis  | a - b | < eps
```

Beispiel: $f(x) = 4 - x^2$; a = 0 ; b = 3

a	b	m
0.0	3.0	1.5
1.5	3.0	2.25
1.5	2.25	1.875
1.875	2.25	2.0625
1.875	2.0625	1.96875
1.96875	2.0625	2.015625
...	...	...
1.9999999999999996	2.000000000000001	2.0

52 Iterationen

Ergebnis: $\bar{x}$ = 2,0

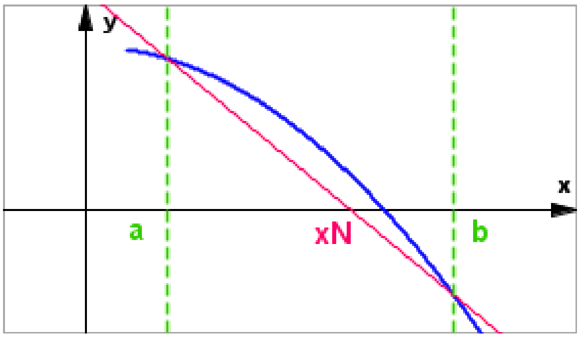
2. Methode: **Regula falsi**

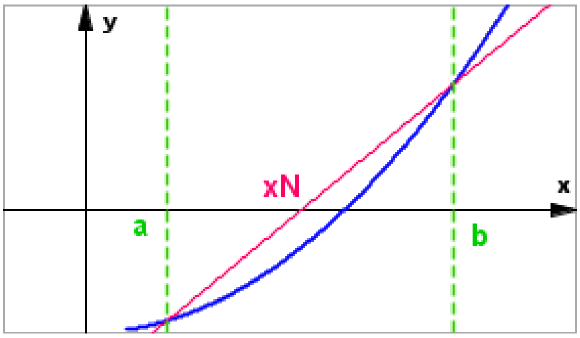
Wir gehen auch hier aus von einem Intervall [a ; b] , das eine Nullstelle einschließt.
Voraussetzung: $f(a) \cdot f(b) < 0$, d.h. die Funktionswerte haben verschiedene Vorzeichen.

Durch die beiden Punkte (a / f(a)) und (b / f(b)) wird die Sekante gelegt, und ihre Schnittstelle xN mit der x-Achse bestimmt .
Die (Geraden-)Gleichung dieser Sekante lautet nach der 2-Punkte-Form: $\frac{y-f(a)}{x-a} = \frac{f(b)-f(a)}{b-a}$

Für xN gilt y = 0, was in die Gleichung eingesetzt wird. Es folgt :

$$\frac{0-f(a)}{xN-a} = \frac{f(b)-f(a)}{b-a} \Leftrightarrow xN-a = -f(a)\frac{b-a}{f(b)-f(a)} \Leftrightarrow \boxed{xN = a - \frac{b-a}{f(b)-f(a)} \cdot f(a)}$$





Gilt dann $f(a) \cdot f(xN) < 0$, so wird b = xN gesetzt ; andernfalls wird a = xN gesetzt.
Wir erhalten so ein neues Intervall [a ; b] , das die Nullstelle enthält.

Anmerkungen:
In beiden Skizzen gilt $f(a) \cdot f(xN) > 0$, da dort f ' und f " gleiches Vorzeichen haben !
Läge z.B. in Skizze 1 eine Linkskrümmung (f " > 0) vor, dann wäre dort f(xN) negativ.
Hat f " überall im Intervall das gleiche Vorzeichen, so bleibt eine der Intervallgrenzen für alle weiteren Iterationen stehen, denn die Sekante liegt immer unterhalb bzw. oberhalb der Funktion.
Es gibt bessere Verfahren, die diesen Nachteil teilweise aufheben (z.B. Pegasus-Verfahren).

Die beiden Schritte (xN bilden ; Intervallgrenzen neu setzen) werden fortgesetzt, bis f(xN) betraglich genügend klein ist (Abbruchbedingung anders als beim Bisektionsverfahren, weil es vorkommen kann, dass a bzw. b sich nicht ändert (siehe Anmerkung oben bzw. Beispiel unten)

```
          Algorithmus:
Eingabe a, b  mit  f(a) · f(b) < 0  sowie eps = 1E-10  (z.B.)
Wiederhole
    xN = a - (b - a) / (f(b) - f(a)) · f(a)
    Falls  f(a) · f(xN) < 0
        dann  b = xN
        sonst  a = xN
bis  | f(xN) | < eps
```

Beispiel: $f(x) = 4 - x^2$; a = 0 ; b = 3

```
  a                  b          xN
0.0                  3.0    1.3333333333333333
1.3333333333333333   3.0    1.8461538461538463
1.8461538461538463   3.0    1.9682539682539681
1.9682539682539681   3.0    1.9936102236421724
...                  ...    ...
1.9999999999999996   3.0    2.0
24 Iterationen
```

Ergebnis: $\overline{x} = 2,0$

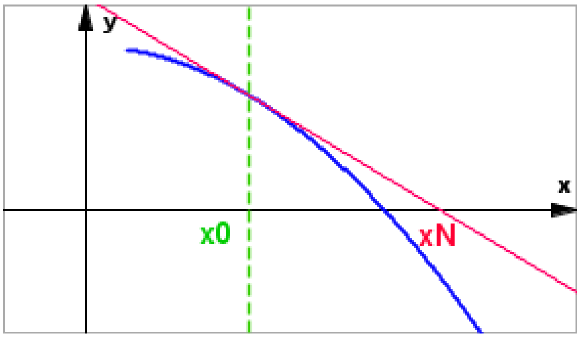
3. Methode: **Newton-Verfahren**

Das Newton-Verfahren ist ein Tangentenverfahren, benötigt also die erste Ableitung f ' (x) .
Es benötigt ferner zum Start einen Wert x_0 in der Nähe der Nullstelle .
Wir bestimmen zunächst $f(x_0)$ und legen die Tangente durch den Punkt (x_0 / $f(x_0)$) .

Tangentengleichung (Punkt-Steigungsform): $\frac{y-f(x_0)}{x-x_0}=f'(x_0)$

Ihre Schnittstelle xN mit der x-Achse wird bestimmt .

Für x = xN gilt y = 0: $\frac{0-f(x_0)}{xN-x_0}=f'(x_0) \Leftrightarrow xN-x_0=-\frac{f(x_0)}{f'(x_0)} \Leftrightarrow \boxed{xN=x_0-\frac{f(x_0)}{f'(x_0)}}$



x0 wird dann gleich xN gesetzt und die Formel wird fortlaufend angewandt, bis der zu subtrahierende Bruch genügend klein ist.

Achtung: Das Verfahren kann scheitern, da die Tangente die x-Achse "weit draußen" schneiden kann !!
Auch kann die Ableitung f ' in bestimmten Fällen 0 werden (Division durch Null !) .
Hierzu ein Beispiel weiter unten ("Newton im Käfig")

Algorithmus:

```
Eingabe   x0   sowie eps = 1E-10  (z.B.)
Wiederhole
    bruch = f(x0) / f '(x0)
    x0 = x0 - bruch
bis  | bruch | < eps
```

Beispiel: $f(x) = 2 - 0{,}15x^2$; x0 = 2

```
  bruch                        x0
-2.3333333333525874          2.0
0.6282051282331431           4.333333333352588
0.05325614407775464          3.7051282051194447
3.8832369208648853E-4        3.65187206104169
2.064849580811375E-8         3.6514837373496034
-0.0                         3.6514837167011076
6 Iterationen
```

Ergebnis: $\overline{x}$ = 3,6514837167011076

"Newton im Käfig"

Ein (hier: konstruiertes) Beispiel dafür, dass das Newtonverfahren scheitern kann.

Wir wählen $f(x) = (x-a)^3 - (x-a)$ mit dem Startwert $x_0 = a - \frac{1}{5}\sqrt{5}$ [≈ 0,5527864045 für a = 1] .

Es folgt: $f\,'(x) = 3(x-a)^2 - 1$

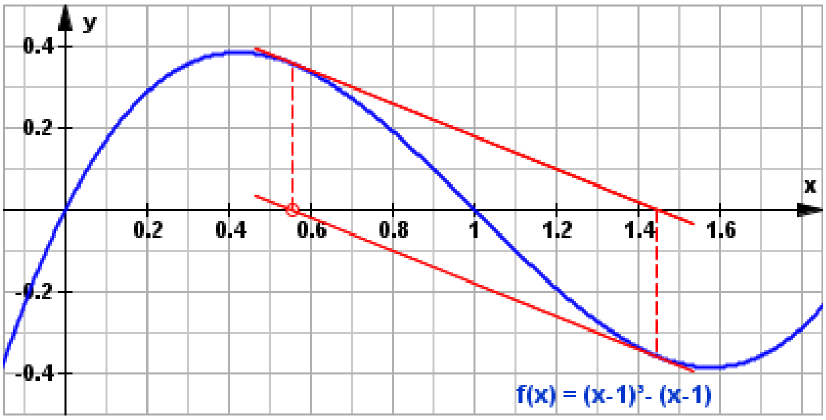
Iteration: $x = x - [\,(x-a)^3 – (x-a)\,] \,/\, [\,3(x-a)^2 - 1\,]$

Vereinfacht: $\mathbf{x = (2x^3 - 3ax^2 + a^3 - a) \,/\, (3x^2 - 6ax + 3a^2 - 1)}$

Für den Startwert $x_0 = a - \frac{1}{5}\sqrt{5}$ ergibt sich $x_1 = x_0 = a + \frac{1}{5}\sqrt{5}$ [≈ 1,4472135955 für a = 1] .

Setzt man dann x_1 in die Iterationsformel ein, so erhält man $x_2 = a - \frac{1}{5}\sqrt{5}$,

also wieder x_0 .



Man sieht also, dass die Iteration immer zwischen 2 Werten hin- und her pendelt !

Newton ist somit im "Käfig eingesperrt" !

"Newton löst auch numerisch kniffligeFälle"

Nochmals ein konstruiertes Beispiel, diesmal für ein numerisch kniffliges Problem.

Gelöst werden soll $f(x) = x^2 - 2\cdot 10^8\cdot x + 1 = 0$

Die bekannte p-q-Lösungsformel liefert als Lösungen $\boxed{x_{1;2} = 10^8 \pm \sqrt{10^{16} - 1}}$

Computer mit der üblichen 64bit - Gleitkommaarithmetik geben als Ergebnis für obige Wurzel 10^8 aus !

Somit wären die beiden (falschen) Lösungen $2\cdot 10^8$ und 0 .

Dieses falsche Wurzelergebnis entsteht durch "Auslöschung", weil die Zahl 1 gegenüber 10^{16} viel zu klein ist, um "ins Gewicht zu fallen".
Die Gleitkommaarithmetik liefert als Ergebnis der Subtraktion im Wurzelterm 10^{16}, richtig ist jedoch $9{,}999999999999999\cdot 10^{15}$. Selbst die manuelle Eingabe von $9{,}999999999999999\cdot 10^{15}$ wird intern zu 10^{16} umgewandelt, wie der folgende Java-Code zeigt:

```
x = 9.999999999999999E15;
System.out.println(x);
```

Ergebnis: 1.0E16

Mit einem Programm für reelle Langzahlen kann man übrigens genauere Lösungen finden, nämlich:

$10^{16} - 1 = 9999999999999999 = 9{,}999999999999999\cdot 10^{15}$
$\sqrt{(10^{16} - 1)} \approx 9{,}9999999999999999499999999999999875\cdot 10^{7}$

$x1 \approx 1{,}9999999999999999499999999999999875\cdot 10^{8}$
$x2 \approx 5{,}0000000000000001250000000000000627\cdot 10^{-9}$

Das **Newtonverfahren** liefert recht brauchbare Ergebnisse:

Startwert x0 = 0,1 :

```
0.1
4.949975956902719E-9
5.0E-9
5.0E-9
```

Startwert x0 = 2E8 :

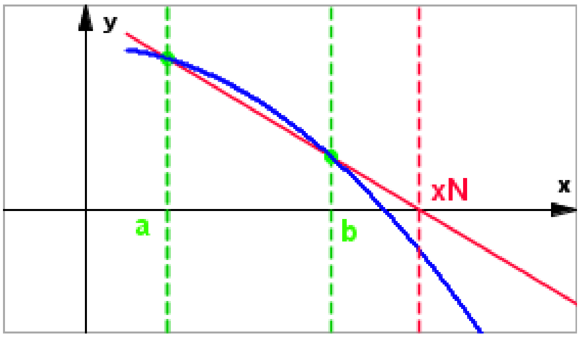
```
2.0E8
2.0E8
 ...
```

4. Methode: **Sekanten-Verfahren**

Wir gehen hier aus von einem Intervall [a ; b] , das aber nicht notwendig eine Nullstelle einschließt !
Dies ist einer der beiden Unterschiede zur "regula falsi" .
Man kann das Verfahren so interpretieren, dass man das Newton-Verfahren als Ausgangspunkt nimmt, dann aber statt der Tangente an (a / f(a)) eine Sekante durch diesen und einen zweiten Punkt (b / f(b)) legt.

Die Schnittstelle xN der Sekante mit der x-Achse wird bestimmt .
Hierfür ergibt sich natürlich die gleiche Formel wie bei der "regula falsi" :

$$xN = a - (b - a) / (f(b) - f(a)) \cdot f(a)$$



Bei der Bestimmung des neuen Intervalls [a ; b] ergibt sich der zweite Unterschied zur "regula falsi" :
Es wird nämlich hier ohne Fallunterscheidung gesetzt : a = b und b = xN .

Die beiden Schritte (xN bilden ; Intervallgrenzen neu setzen) werden fortgesetzt, bis f(xN) betraglich genügend klein ist. Die Abbruchbedingung ist hier so wie bei der "regula falsi" gewählt.

Achtung: Das Verfahren kann scheitern, da die Nullstelle nicht unbedingt in [a ; b] liegen muss !!

Algorithmus:

Eingabe a, b mit f(a) · f(b) < 0 sowie eps = 1E-10 (z.B.)
Wiederhole
    xN = a - (b - a) / (f(b) - f(a)) · f(a)
    a = b
    b = xN
bis | f(xN) | < eps

Beispiel: $f(x) = 4 - x^2$; a = 0 ; b = 3

a	b	xN
0.0	3.0	1.3333333333333333
3.0	1.3333333333333333	1.846153846153846
1.3333333333333333	1.846153846153846	2.032258064516129
1.846153846153846	2.032258064516129	1.99872040946897
2.032258064516129	1.99872040946897	1.9999897600262144
1.99872040946897	1.9999897600262144	2.0000000032768
1.9999897600262144	2.0000000032768	1.9999999999999916
2.0000000032768	1.9999999999999916	2.0

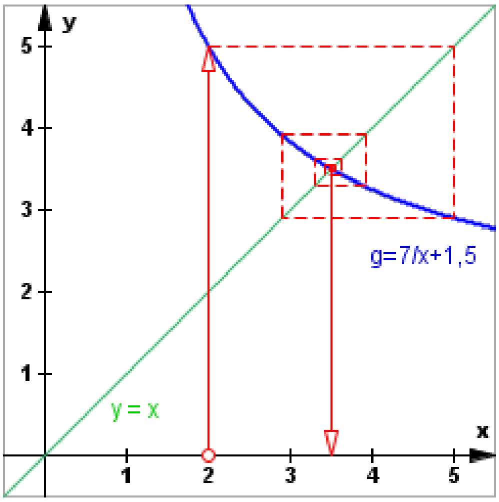
8 Iterationen

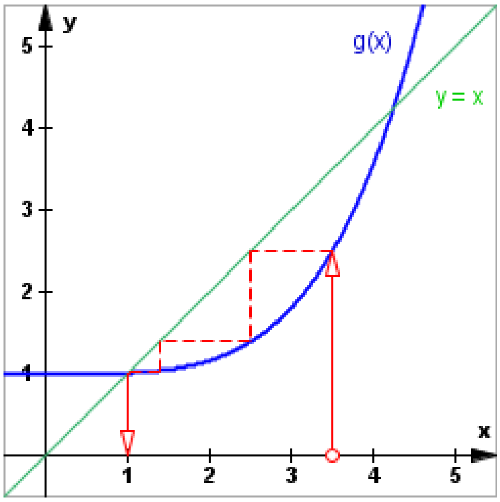
Ergebnis: $\bar{x}$ = 2,0

5. Methode: **Fixpunktverfahren (Allgemeines Iterationsverfahren)** $x_{n+1} = g(x_n)$

Dieses Verfahren unterscheidet sich von den 4 vorherigen grundlegend, da das Ziel die Lösung der Gleichung **x = g(x)** für eine stetige Funktion g ist .

Das Verfahren konvergiert nur unter der Voraussetzung, dass in einer Umgebung der Lösung der obigen Gleichung gilt : $|g'(x)| < 1$.





Ein Startwert x0 muss vorgegeben werden; hiermit dann fortwährend x0 = g(x0) bilden, bis sich zwei aufeinanderfolgende Ergebnisse nicht mehr wesentlich ändern .

Algorithmus:

```
Eingabe   gx=x0   sowie eps = 1E-10  (z.B.)
Wiederhole
    x0 = gx
    gx = g(x0)
bis  | gx - x0 | < eps
```

Beispiel: g(x) = cos(x) ; gx = 0,5

```
     x0
0.5
0.8775825618903728
0.6390124941652592
0.8026851006823349
0.6947780267880062
0.7681958312820161
0.719165445942419
0.752355759421527
    ...
0.7390851332151602
0.7390851332151609
88 Iterationen
```

Ergebnis: $\bar{x}$ = 0,7390851332151609

Nullstellen mit dem Fixpunktverfahren finden

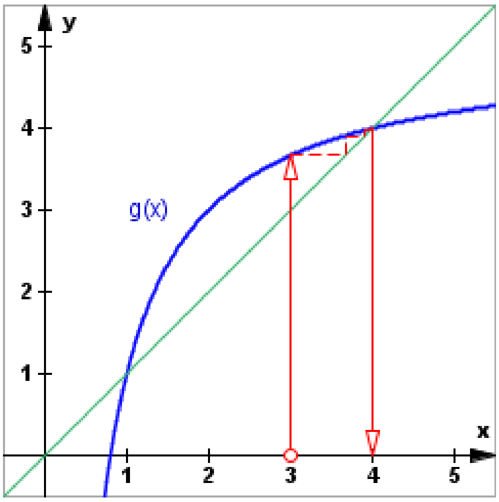
Die Lösung von $f(x) = 0$ lässt sich in vielen Fällen auch dann finden, wenn wir diese Gleichung auf die Form $x = g(x)$ bringen.
So ist das Allgemeine Iterationsverfahren auch als Nullstellensucher verwendbar.

Beispiel 1: $f(x) = x^2 - 5x + 4 = 0$

Äquivalenzumformung (für $x <> 0$): $x - 5 + 4 / x = 0$ bzw. $x = 5 - 4 / x$.
Die Funktion **$g(x) = 5 - 4/x$** besitzt die Ableitung $g'(x) = 4 / x^2$
Für alle x mit $|x| > 2$ ist $|4 / x^2| < 1$.
Wir können also im Bereich $|x| > 2$ iterieren.

Wählen wir den Startwert $x0 = 3$, so ergibt sich die gegen 4 konvergierende Folge

3,0
3,666667
3,909091
3,976744
3,994152
3,998536
3,999634
3,999908
...

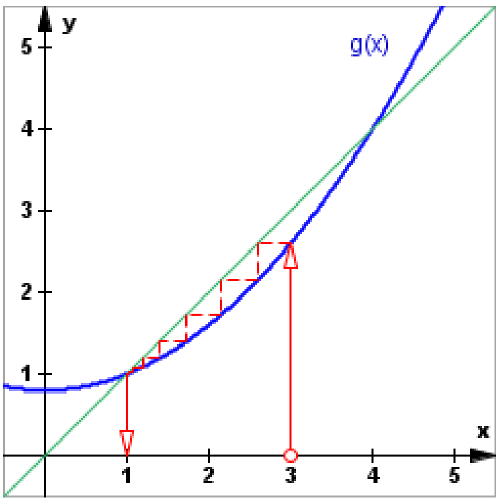


Die andere Nullstelle $x = 1$ von f(x) erhalten wir mit $g(x) = 5 - 4/x$ nicht, da die betreffende Folge in der Umgebung von $x = 1$ divergiert (Es sei denn, wir wählen den Startwert = 1) !

In vielen Fällen hilft es jedoch, die Gleichung $f(x) = 0$ auf andere Weise umzuformen, so dass man auf eine konvergent Folge hoffen kann. Z.B. $x^2 - 5x + 4 = 0 \rightarrow x^2 + 4 = 5x \rightarrow x = (x^2 + 4) / 5$
Die Funktion **$g(x) = (x^2 + 4) / 5$** besitzt die Ableitung $g'(x) = 0{,}4x$
Die Steigung ist betraglich < 1 im Intervall] -2,5 ; 2,5 [.
Der Startwert kann aber z.B. auch 3 sein (s.u.).

Wählen wir den Startwert $x0 = 3$, so ergibt sich die gegen 1 konvergierende Folge

3
2,6
2,152
1,726221
1,395968
1,189745
1,083099
1,034621
1,014088
1,005675
1,002276
...



Hinweis:

Die naheliegende Umformung $x^2 - 5x + 4 = 0 \rightarrow x^2 = 5x - 4 \rightarrow x = \sqrt{(5x - 4)} \rightarrow$
$g(x) = \sqrt{(5x - 4)}$ und $g'(x) = 2{,}5 / \sqrt{(5x - 4)}$ führt übrigens nicht auf die Lösung 1 ,
weil $|g'(x)| > 1$ für $x < 2{,}05$ gilt.
Jedoch konvergiert die betreffende Folge gegen 4, falls der Startwert > 1 gewählt wird .

Beispiel 2: $f(x) = x^4 - 3x + 1 = 0$

Mögliche Umformungen sind

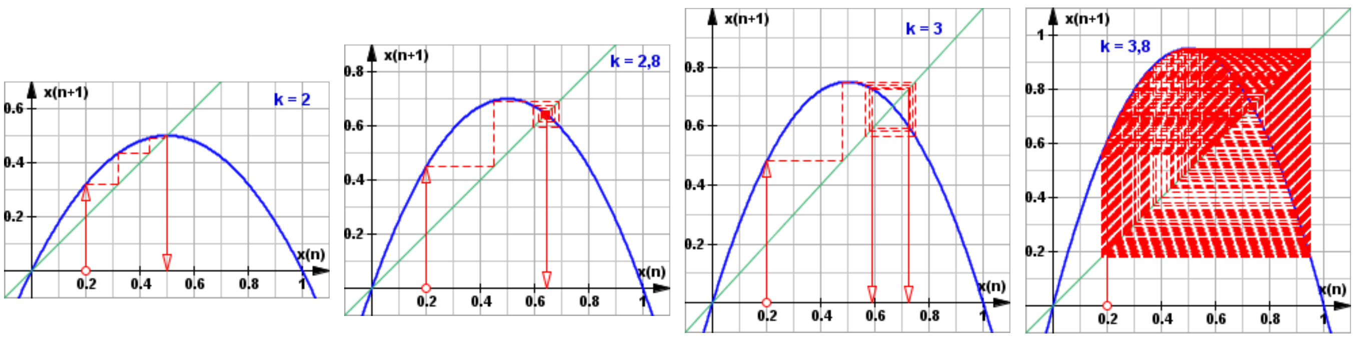
$x = (x^4 + 1) / 3$ $\quad x = (3x - 1)^{0,25}$ $\quad x = \sqrt{(3 / x - 1 / x^2)}$ $\quad x = (3x - 1) / x^3$ $\quad x = \sqrt[3]{(3 - 1 / x)}$

Beispiel 3: $f(x) = k \cdot x \cdot (1-x)$; $k > 0$

Die zugehörige Folge $\mathbf{x_{n+1} = k \cdot x_n \cdot (1 - x_n)}$ hat es "in sich" .
Sie ist unter der Bezeichnung "Chaosfolge" bekannt !
Die Gleichung heißt "Logistische Gleichung" .

Je nach Wahl des Parameters k erhalten wir ganz verschiedene Iterationsergebnisse:

- für $0 < k \leq 1$ besitzt die Folge den Grenzwert 0
- für $1 < k \leq 3$ besitzt die Folge den Grenzwert $1 - \frac{1}{k}$
- für $3 < k \leq 3{,}4494897$ gibt es 2 Häufungswerte der Folge
- für $3{,}4494897 < k \leq 3{,}544090$ gibt es 4 Häufungswerte der Folge
- ...
- für $3{,}57 < k \leq 4$ gibt es unendlich viele Häufungswerte; die Folge verhält sich "**chaotisch**"
- $k > 4$ divergiert die Folge



Zusammenhänge zwischen den Verfahren

Das Allgemeine Iterationsverfahren ist, wie der Name schon sagt, das allgemeinste der 5 Verfahren :

Z.B. lässt sich das Newton-Verfahren als Spezialfall von $x_{n+1} = g(x_n)$ auffassen, wenn $g(x_n)$ definiert wird als $g(x_n) = x_n - f(x_n) / f'(x_n)$.

Newton-Verfahren: $\mathbf{x_{n+1} = x_n - f(x_n) / f'(x_n)}$.

Auch für das Newton-Verfahren lässt sich ein spezielles Verfahren definieren, und zwar das sog. HERON-Verfahren, welches bei der Wurzelberechnung eine Rolle spielt.

Dazu definiert man: $f(x) = x^k - r$.

Mit $f'(x) = k \cdot x^{k-1}$ ergibt sich :

$\mathbf{x_{n+1} = x_n - (x_n^k - r) / (k \cdot x_n^{k-1})}$ (HERON-Verfahren zur Berechnung der k-ten Wurzel aus r)

Beispiel: $k = 3$; $r = 15{,}625$; Startwert $x0 = 2$

Die betreffende HERON-Folge ist dann $x_{n+1} = x_n - (x_n^3 - 15{,}625) / (3 \cdot x_n^2)$

Es folgen:

x0 = 2.0
x1 = 2.6354166666554493
x2 = 2.5068389905233284
x3 = 2.5000186407098828
x4 = 2.5000000001389893
x5 = 2.5
Lösung: 2.5