Iterative Verfahren zur Nullstellenbestimmung von Funktionen **zeroin**-Algorithmus

Christiane Helzel

Heinrich-Heine-Universität, Düsseldorf

July 11, 2018

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Verfahren zur Berechnung von $\sqrt{2}$

Beispiel: Bestimme $\sqrt{2}$ (wir wissen $1 < \sqrt{2} < 2$)

Verfahren zur Berechnung von $\sqrt{2}$

Beispiel: Bestimme $\sqrt{2}$ (wir wissen $1 < \sqrt{2} < 2$)

Startwert: $x_0 = 1\frac{1}{2}$

Verfahren zur Berechnung von $\sqrt{2}$

Beispiel: Bestimme $\sqrt{2}$ (wir wissen $1 < \sqrt{2} < 2$)

Startwert: $x_0 = 1\frac{1}{2}$
Es gilt: $x_0^2 = \frac{9}{4} > 2 \quad \Rightarrow x_0 > \sqrt{2}$

Verfahren zur Berechnung von $\sqrt{2}$

Beispiel: Bestimme $\sqrt{2}$ (wir wissen $1 < \sqrt{2} < 2$)

Startwert: $x_0 = 1\frac{1}{2}$
Es gilt: $x_0^2 = \frac{9}{4} > 2 \quad \Rightarrow x_0 > \sqrt{2}$

wähle $x_1 = 1\frac{1}{4}$

Verfahren zur Berechnung von $\sqrt{2}$

Beispiel: Bestimme $\sqrt{2}$ (wir wissen $1 < \sqrt{2} < 2$)

Startwert: $x_0 = 1\frac{1}{2}$
Es gilt: $x_0^2 = \frac{9}{4} > 2 \quad \Rightarrow x_0 > \sqrt{2}$

wähle $x_1 = 1\frac{1}{4}$
Es gilt: $x_1^2 = \frac{25}{16} < 2$

Verfahren zur Berechnung von $\sqrt{2}$

Beispiel: Bestimme $\sqrt{2}$ (wir wissen $1 < \sqrt{2} < 2$)

Startwert: $x_0 = 1\frac{1}{2}$
Es gilt: $x_0^2 = \frac{9}{4} > 2 \quad \Rightarrow x_0 > \sqrt{2}$

wähle $x_1 = 1\frac{1}{4}$
Es gilt: $x_1^2 = \frac{25}{16} < 2 \quad \Rightarrow 1\frac{1}{4} < \sqrt{2} < 1\frac{1}{2}$

Verfahren zur Berechnung von $\sqrt{2}$

Beispiel: Bestimme $\sqrt{2}$ (wir wissen $1 < \sqrt{2} < 2$)

Startwert: $x_0 = 1\frac{1}{2}$
Es gilt: $x_0^2 = \frac{9}{4} > 2 \quad \Rightarrow x_0 > \sqrt{2}$

wähle $x_1 = 1\frac{1}{4}$
Es gilt: $x_1^2 = \frac{25}{16} < 2 \quad \Rightarrow 1\frac{1}{4} < \sqrt{2} < 1\frac{1}{2}$

Fortsetzung dieses Prozesses:

$$1\frac{1}{2}, 1\frac{1}{4}, 1\frac{3}{8}, 1\frac{5}{16}, 1\frac{13}{32}, 1\frac{27}{64}, \ldots$$

Matlab: $k = 52$, `a=1.414213562373095`, `1.414213562373095`

Bisektion: Matlab-Programm 1

```
M=2;
a=1;
b=2;
k=0;
while b-a > eps
    x = (a+b)/2;
    if x^2 > M
        b = x
    else
        a = x
    end
    k= k+1;
end
```

$k = 52,\ a = 1.414213562373095,\ b = 1.414213562373095$

Verallgemeinerung

Bestimme x mit $f(x) = 0$ für eine gegebene Funktion f.

Verallgemeinerung

Bestimme x mit $f(x) = 0$ für eine gegebene Funktion f.

Annahmen:

- Wir kennen ein Intervall $[a, b]$ mit $f(a)f(b) < 0$
- Wir können die Funktion f für jedes x berechnen

Verallgemeinerung

Bestimme x mit $f(x) = 0$ für eine gegebene Funktion f.

Annahmen:

- Wir kennen ein Intervall $[a, b]$ mit $f(a)f(b) < 0$
- Wir können die Funktion f für jedes x berechnen

Ziel: Finde ein sehr kleines Intervall $[\tilde{a}, \tilde{b}]$, auf dem die Funktion f ihr Vorzeichen wechselt.

Verallgemeinerung

Bestimme x mit $f(x) = 0$ für eine gegebene Funktion f.

Annahmen:

- Wir kennen ein Intervall $[a, b]$ mit $f(a)f(b) < 0$
- Wir können die Funktion f für jedes x berechnen

Ziel: Finde ein sehr kleines Intervall $[\tilde{a}, \tilde{b}]$, auf dem die Funktion f ihr Vorzeichen wechselt.

Bsp: Finde ein (sehr kleines) Intervall, in dem eine Nullstelle der Funktion $f(x) = x^2 - 2$ liegt.

Bisektion: Matlab-Programm 2

```
k=0;
while abs(b-a) > eps*abs(b)
    x = (a+b)/2;
    if sign(f(x)) == sign(f(b))
        b = x;
    else
        a = x;
    end
    k = k+1;
end
```

Bisektion: Eigenschaften

Bisektion / **Intervallhalbierung** erzeugt konvergente Folge von Intervallschachtelungen

Bisektion: Eigenschaften

Bisektion / **Intervallhalbierung** erzeugt konvergente Folge von Intervallschachtelungen

- Bisektion ist **langsam**
  $k = 52$ bei Verwendung des Abbruchkriteriums

```
while abs(b-a) > eps*abs(b)
```

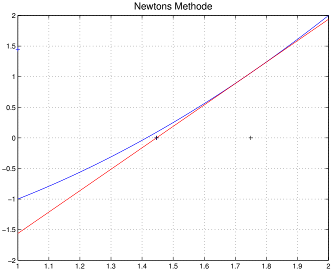
Bisektion: Eigenschaften

Bisektion / **Intervallhalbierung** erzeugt konvergente Folge von Intervallschachtelungen

- Bisektion ist **langsam**
  $k = 52$ bei Verwendung des Abbruchkriteriums
  ```
  while abs(b-a) > eps*abs(b)
  ```
- Bisektion ist **zuverlässig**

Newton's Methode

Zeichne an einen Punkt $\tilde{x}$ die Tangente an den Graph der Funktion f und bestimme den Schnittpunkt der Tangente mit der x-Achse.



wähle Startwert x_0

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Newtons Methode: Matlab $\sqrt{2}$

```
k=0;
while abs(x - xprev) > eps*abs(x)
    xprev = x;
    x = x - f(x) / fprime(x)
    k = k+1;
end
```

Newtons Methode

Bsp.: Berechnung von $\sqrt{2}$

$$\begin{aligned} f(x) &= x^2 - 2 \\ f'(x) &= 2x \end{aligned}$$

Wir erhalten das Iterationsverfahren

$$\begin{aligned} x_{n+1} &= x_n - \frac{f(x_n)}{f'(x_n)} \\ &= x_n - \frac{x_n^2 - 2}{2x_n} \\ &= \frac{1}{2}\left(x_n + \frac{2}{x_n}\right) \end{aligned}$$

Newtons Methode: Matlab

```
while abs(x - xprev) > eps*abs(x)
    xprev = x
    x = 0.5*(x+2/x);
end
```

Newtons Methode: Matlab

```
while abs(x - xprev) > eps*abs(x)
    xprev = x
    x = 0.5*(x+2/x);
end
```

```
1.500000000000000
1.416666666666667
1.414215686274510
1.414213562374690
1.414213562373095
1.414213562373095
```

Newtons Methode: Eigenschaften

Vorteile:

- **Schnelle** Berechnung der Nullstelle
- Newtons Methode ist ein wesentlicher Bestandteil effizienter Algorithmen zur Nullstellenbestimmung

Newtons Methode: Eigenschaften

Vorteile:

- **Schnelle** Berechnung der Nullstelle
- Newtons Methode ist ein wesentlicher Bestandteil effizienter Algorithmen zur Nullstellenbestimmung

Nachteile:

- Nur anwendbar für stetig differenzierbare Funktionen $f(x)$
- Man muss $f'(x)$ berechnen
- Der Startwert muss in der *Nähe* der Nullstelle liegen

Newtons Methode: Konvergenz

Sei $\tilde{x}$ Nullstelle von $f(x)$ und

$$e_n = x_n - \tilde{x}$$

der Fehler der n-ten Iteration.

Newtons Methode: Konvergenz

Sei $\tilde{x}$ Nullstelle von $f(x)$ und

$$e_n = x_n - \tilde{x}$$

der Fehler der n-ten Iteration.

- $f'(x)$ und $f''(x)$ existieren und sind stetig
- Startwert x_0 ist in der Nähe von $\tilde{x}$

Newtons Methode: Konvergenz

Sei $\tilde{x}$ Nullstelle von $f(x)$ und

$$e_n = x_n - \tilde{x}$$

der Fehler der n-ten Iteration.

- $f'(x)$ und $f''(x)$ existieren und sind stetig
- Startwert x_0 ist in der Nähe von $\tilde{x}$

Dann gilt

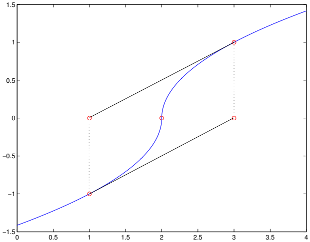
$$e_{n+1} = \frac{1}{2} \frac{f''(\xi)}{f'(x_n)} e_n^2,$$

mit einem ξ zwischen x_n und $\tilde{x}$.

Anwendungsbeispiel

Bestimme die Nullstelle von

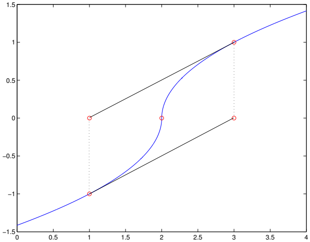
$$f(x) = \mathsf{sign}(x-2)\sqrt{|x-2|}$$



Anwendungsbeispiel

Bestimme die Nullstelle von

$$f(x) = \mathsf{sign}(x-2)\sqrt{|x-2|}$$

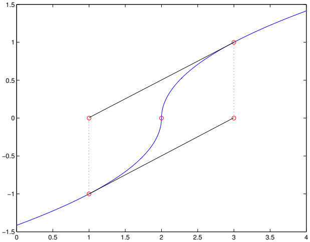


Newtons Method liefert $1, 3, 1, 3, 1, 3, 1, 3, \ldots$

Anwendungsbeispiel

Bestimme die Nullstelle von

$$f(x) = \mathsf{sign}(x-2)\sqrt{|x-2|}$$



Newtons Method liefert $1, 3, 1, 3, 1, 3, 1, 3, \ldots$

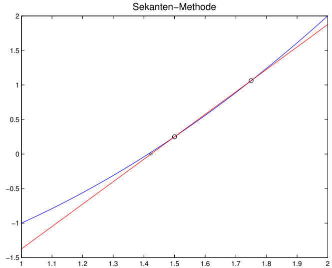
Bisektion mit $a = 1$, $b = 3$ liefert Lösung nach 53 Iterationen

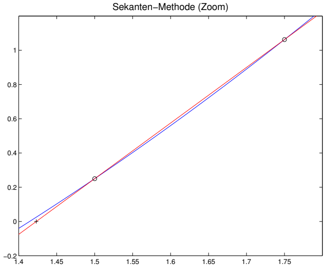
Sekanten-Verfahren

Ersetze im Newton-Verfahren die Tangente durch eine **Sekante**.

$$s_n = \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}$$

$$x_{n+1} = x_n - \frac{f(x_n)}{s_n}$$





Sekanten-Methode: Matlab $\sqrt{2}$

```
while abs(b-a) > eps*abs(b)
    c = a;
    a = b;
    b = b + (b-c) / (f(c) / f(b) -1);
    k = k+1;
end
```

Sekanten-Methode: Matlab $\sqrt{2}$

```
while abs(b-a) > eps*abs(b)
    c = a;
    a = b;
    b = b + (b-c) / (f(c) / f(b) -1);
    k = k+1;
end
```

```
1.333333333333333
1.400000000000000
1.414634146341463
1.414211438474870
1.414213562057320
1.414213562373095
1.414213562373095
```

Sekanten-Methode: Eigenschaften

- Wir benötigen 2 Startwerte
- Wir benötigen keine Kenntnis über $f'(x)$
- Ähnlich Konvergenzeigenschaft wie beim Newton-Verfahren

Inverse Quadratische Interpolation (IQI)

Wie wäre es, wenn wir 3 Startwerte verwenden würden?

Inverse Quadratische Interpolation (IQI)

Wie wäre es, wenn wir 3 Startwerte verwenden würden?

Berechne eine Parabel $P(y)$, die durch die Punkte $(f(a), a)$, $(f(b), b)$ und $(f(c), c)$ verläuft.
Benutze $x = P(0)$ als neue Nährung an die Nullstelle von $f(x)$.

Inverse Quadratische Interpolation (IQI)

Wie wäre es, wenn wir 3 Startwerte verwenden würden?

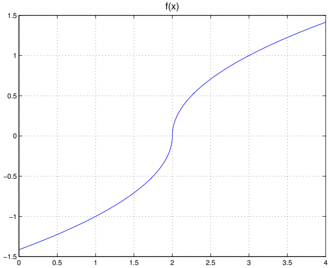
Berechne eine Parabel $P(y)$, die durch die Punkte $(f(a), a)$, $(f(b), b)$ und $(f(c), c)$ verläuft.
Benutze $x = P(0)$ als neue Nährung an die Nullstelle von $f(x)$.

```
k=0;
while abs(c-b) > eps*abs(c)
    x = polyinterp([f(a), f(b), f(c)], [a, b, c],0)
    a = b;
    b = c;
    c=x;
    k = k+1;
end
```

Inverse Quadratische Interpolation: Illustration

Betrachte

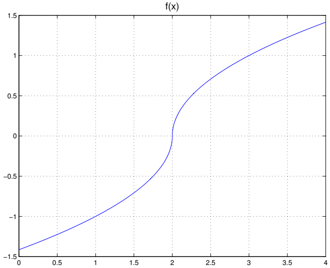
$$f(x) = \mathsf{sign}(x-2)\sqrt{|x-2|}$$

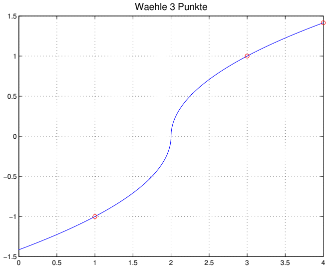


Inverse Quadratische Interpolation: Illustration

Betrachte

$$f(x) = \mathsf{sign}(x-2)\sqrt{|x-2|}$$

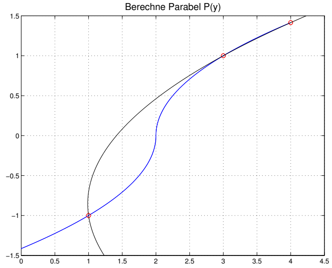




Inverse Quadratische Interpolation: Illustration

Betrachte

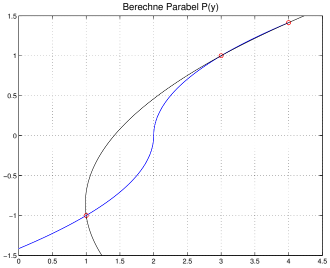
$$f(x) = \mathsf{sign}(x-2)\sqrt{|x-2|}$$

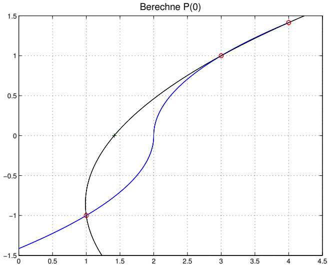


Inverse Quadratische Interpolation: Illustration

Betrachte

$$f(x) = \mathsf{sign}(x-2)\sqrt{|x-2|}$$





Inverse Quadratische Interpolation: Eigenschaften

- $f(a)$, $f(b)$ und $f(c)$ müssen voneinander verschieden sein
- Bei ungünstiger Wahl der 3 Punkte kann IQI sehr ungenau sein
- Schnelle Konvergenz in der Nähe einer Nullstelle

Die Methode der Wahl: Dekkers Zeroin-Algorithmus (1969)

Kombiniere die Zuverlässigkeit der Bisektions-Methode mit der schnellen Konvergenz der Sekanten-Methode und der IQI-Methode

- Wähle a, b mit $f(a)f(b) < 0$
- Berechne c unter Verwendung der Sekanten-Methode
- Führe die folgenden Schritte durch, bis $|b - a| < \varepsilon|b|$ oder $f(b) = 0$ gilt
- Vertausche die Reihenfolge von a, b, c, so dass
  - $f(a)f(b) < 0$
  - $|f(b)| \leq |f(a)|$
  - c das b aus dem vorherigen Schritt ist
- Falls $c \neq a$, dann führe eine IQI-Iteration durch
- Falls $c = a$, dann führe eine Sekanten-Iteration durch
- Falls Nährung aus IQI oder Sekanten-Iteration im Intervall $[a, b]$ liegt, dann akzeptiere diesen Schritt. Führe ansonsten einen Bisektionsschritt durch.

Referenz

Cleve B. Moler, Numerical Computing with MATLAB, SIAM 2004.