

# Newton- Raphson Verfahren

Von Heinz - Volker Viehof

1: was ist das „Newton- Raphson Verfahren“:

Das Newton– Raphson- oder auch Newtonsches- Näherungsverfahren ist ein Algorithmus zur Errechnung von werten wie z.B. Wurzeln oder Kehrwerte, die man nicht unmittelbar in einem Term errechnen kann.

2: wie funktioniertes ?

wenn wir zum Beispiel die Quadratwurzel von x errechnen wollen, würden wir einen geschätzten Wert, bzw. einen Startwert  $y[\text{alt}]$  annehmen und berechnen:  $y[\text{neu}] = x / y[\text{alt}]$  danach würden wir prüfen ob  $y[\text{alt}]$  gleich  $y[\text{neu}]$ , wenn wahr ist y die Quadratwurzel von x. Ist  $y[\text{neu}]$  ungleich  $y[\text{alt}]$ , bilden wir ein „neues“  $y[\text{alt}]$  aus dem Mittelwert des bisherigen  $y[\text{alt}]$  und  $y[\text{neu}]$ :

$y[\text{alt}] = (y[\text{alt}] + y[\text{neu}]) / 2$  mit diesem „neuen“  $y[\text{alt}]$  wird die Funktion abermals durchlaufen und getestet ob sie sie erfüllt, und so weiter ...

3: von Hand ...

nehmen wir an wir wollen die Wurzel von 36 errechnen (ist 6, weiß ich auch ohne zu rechnen, aber es soll ja die Funktion bewiesen werden):

gegeben  $x=36$ , startwert  $n=1.1$ , gesucht Wurzel(y) von x

$y=x/n$  ;  $y=36/1.1 = 32.727$  Bedingung nicht erfüllt, also  $n=(y+n)/2=16.913$ , dann:

$y=x/n$  ;  $y=36/16.913 = 2.128$  Bedingung nicht erfüllt, also  $n=(y+n)/2=9.521$  , dann:

$y=x/n$  ;  $y=36/9.521 = 3.781$  Bedingung nicht erfüllt, also  $n=(y+n)/2=6.651$  , dann:

$y=x/n$  ;  $y=36/6.651 = 5.412$  Bedingung nicht erfüllt, also  $n=(y+n)/2=6.031$  , dann:

$y=x/n$  ;  $y=36/6.031 = 5.968$  Bedingung nicht erfüllt, also  $n=(y+n)/2=6.000$  , dann:

$y=x/n$  ;  $y=36/6.000 = 6.000$  y ist gleich n! die Bedingung ist erfüllt!

Die Wurzel von 36 ist 6!

4: Programmbeispiel in QBASIC:

```
REM <C> 2004 h.v.viehof
```

```
2 PRINT
```

```
PRINT "Wurzelziehen nach dem Newton-Raphson-Verfahren"
```

```
INPUT "wert:", wert
```

```
start = 1.1
```

```
zaehler = 0
```

```
1 neu = wert / start
```

```
IF start = neu GOTO 8
```

```
start = (start + neu) / 2
```

```

zaehler = zaehler + 1
IF zaehler <= 100 GOTO 1
8 PRINT "Die Wurzel von", wert, "ist:", start
INPUT "ende (j/n)", e$
IF e$ <> "j" GOTO 2
END

```

5: Als JavaScript – Funktion eingebettet in HTML:

```

<html>
<head>
<title>Test</title>
<script type="text/javascript">
<!--
function Wurzel() {
    var wert = 1.1;
    var z = 0;
    var wert_new = document.Formular.Eingabe.value / wert;
    while(wert_new != wert && z <=100){
        var wert_new = (wert + wert_new) /2;
        var wert = document.Formular.Eingabe.value / wert_new;
        var z = z + 1;
    }
    var Ergebnis = wert;
    alert("Die Wurzel von " + document.Formular.Eingabe.value + " = " + Ergebnis);
}
//-->
</script>
</head>
<body>
<h1> Die Quadratwurzel einer Zahl mithilfe des<br>
Newton-Raphson-Verfahrens errechnet :</h1>
<form name="Formular" action="">
<input type="text" name="Eingabe" size="3">
<input type="button" value="Wurzel errechnen" onClick="Wurzel()">
</form>
</body>
</html>

```

## 6: Programmbeispiel In Python:

Den Tkinter zu erklären würde hier zu weit führen, kurz gesagt, der Tkinter ist eine graphische Erweiterung von Python, die es ermöglicht Windows GUI Funktionen,

z.B. Fenster, Buttons, Eingabefelder und so weiter zu nutzen, wer hierüber mehr erfahren will schaue bitte im Internet oder entsprechender Literatur nach.\*

```
# Programmbeispiel in Python : Berechnung einer
# Quadratwurzel nach dem Newtonschen Näherungsverfahren bzw.
# Newton-Raphson-Verfahren

# Module importieren
from Tkinter import *
from tkSimpleDialog import *

# Funktion zum beenden des Programms und schließen des Fensters
def exit():
    nn.destroy()
    nn.quit()

# Hier beginnt die Programmhauptschleife :
nn=Tk()
nn.n=askinteger('Wert ?','Bitte geben Sie eine Ganzzahl ein !',initialvalue=36)

# Startwerte setzen (z = Zähler für Schleifendurchläufe)
x=1.1000
z=0

# das ist jetzt das eigentliche Newton- Raphson- Verfahren:
while not x == nn.n/x :
    x=(x+nn.n/x)/2

# Schleifendurchlauf zählen
z=z+1

# nach 100 Iterationen (Schleifendurchläufen)abbrechen – Genauer wollen
```

```

# wir's gar nicht wissen!
    if z>100:
        break
# jetzt werden die Ergebnisse in einem Ausgabefenster dargestellt:
mess="Wurzel von "+str(nn.n)+" : "+str(x)+"\nnach "+str(z)+" Iterationen"
nn.title("Quadratwurzel")
Label(nn,text=mess).pack()
# „Ende-Button schließt das Fenster und beendet das Programm
xx=Button(nn,text="Ende",command=exit).pack()
nn.mainloop()

```

Die Beispielprogramme ist auf meiner Downloadseite zu finden.

Zum Schluss noch die Realisation des Newton Raphson Verfahrens als Prozedur für 16 Bit Assembler:

```

_newton:
    mov bx, 3                ; das Newton Raphson Verfahren
    mov cx, 0ffff           ; Zaehler für max. Iteration
X1:
    xor dx, dx              ; DX loeschen da es bei der Division benutzt
                            ; wird
    mov ax,[Save]           ; den Wert in AX
    div bx                  ; und durch BX teilen
    cmp ax, bx              ; Wenn AX == BX ist die gesuchte Wurzel gefunden
    jz _exit               ; dann ist hier Schluss ;- )
    add bx, ax              ; sonst Mittelwert bilden: (AX+BX)
    shr bx, 1               ; /2
    inc word ptr[Count]    ; der [Count] Zaehler zeigt die Schleifen
                            ; durchlaeufer
    loop X1                 ; von vorn, sofern nicht max. Iteration
                            ; ueberschritten
_exit:
    ret                     ; hier fertig und zurueck!

```

In anderen Programmiersprachen wäre die Vorgehensweise ähnlich.

\*Literaturhinweis zu „Tkinter“, „Tk“ und „Python“:

Michael Lauer ; Python und GUI-Toolkits ; mitp Verlag ; ISBN 3-8266-0844-5

Martin Uzak ; Python 2.x Das Einsteigerseminar ; vmi Verlag ; ISBN 3-8266-7206-2

Newton-raphson/hvviehof/htmlversion/10.09.2004

Geändert und Ergänzt am 28.09.2004